



**OTIMIZAÇÃO DE GASTOS ENERGÉTICOS  
EM COMUNICAÇÕES ENTRE SENSORES  
*LOW POWER*  
Algoritmos bio inspirados e recolha energética**

**Débora da Costa Pinho**

Dissertação para obtenção do Grau de Mestre em  
**Engenharia Eletrotécnica e de Computadores**  
2º ciclo de estudos

Orientador: Prof. Doutor José António da Costa Salvado  
Coorientador: Prof. Doutor António Eduardo Vitória do Espírito Santo

**janeiro de 2024**

**Folha em branco**

## **Declaração de Integridade**

Eu, Débora da Costa Pinho, que abaixo assino, estudante com o número de inscrição M11791 de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior.**

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 31 / 01 / 2024

Débora da Costa Pinho

**Folha em branco**

# Dedicatória

*Pelos que já cá não estão.  
Que a nossa coragem seja,  
no mínimo,  
do tamanho dos nossos sonhos.*

**Folha em branco**

# Agradecimentos

Aos meus pais Fátima Santos e João Pinho e à minha avó Maria Eugênia, um grande agradecimento pelo esforço de me permitirem e proporcionarem que chegasse e concluísse mais esta etapa, pela paciência, apoio, dedicação, alegria, alento e incentivo que me foram dando ao longo de todos estes anos, não só de mestrado, como do meu percurso académico. Agradeço ainda todo o resto que me deram e darão sempre, sem o vosso apoio jamais conseguiria conquistar o que conquistei.

A ti, Fábio Rodrigues, um especial agradecimento pela ajuda, sugestões e tempo dedicado comigo, no auxílio da realização da presente dissertação. Obrigada pela paciência, apoio, compreensão e encorajamento, mesmo nos momentos menos bons. Obrigada por seres uma inspiração e antes de qualquer outra coisa, o meu melhor amigo. Por estares sempre ao meu lado, neste momento e em todos os restantes que a vida nos trouxe e trará. ” *Pouca coisa é necessária para transformar inteiramente uma vida: amor no coração e sorriso nos lábios.*”

Ao meu orientador professor doutor José António da Costa Salvado, por me guiar, aconselhar e fornecer todos os recursos e conhecimentos necessários para a realização desta trabalho, e acima de tudo, pelo tempo que despendeu no auxílio e realização desta dissertação. Ao coorientador professor doutor António Espírito Santo pelo auxílio prestado na elaboração deste trabalho.

A ti, Daniela Pereira, pela força e incentivo, mas também por todos os momentos de descontração, coragem e confiança que me foste e vais oferecendo. E ainda pela nossa amizade pois, “*Deus fez-nos amigas, porque nenhuma mãe nos aguentaria como irmãs...*”, obrigada.

Aos meus restantes amigos que me apoiam, acreditam em mim e me dão força para continuar.

À universidade da Beira Interior pela boa forma que me recebeu, por me permitir chegar onde cheguei e pela experiência, num todo, que me proporcionou a entrada cá.

A todos os mencionados e aos restantes que de alguma forma, direta ou indiretamente, me apoiaram e motivaram ao longo desta etapa, o meu muito obrigado a todos, sem vocês não seria possível, e como se costuma dizer na Covilhã, “ *um bem haja*”.

**Folha em branco**

## Resumo

Atualmente, o interesse pela utilização de fontes de energia biológica e inspiradas também em processos biológicos, tem vindo a intensificar-se. Ao longo dos anos a procura por redes autossustentáveis e autónomas tem crescido, desde pequenas aplicações como sensores de monitorização até grande escala como a indústria 4.0.

Os avanços principais das publicações, encontram aplicação em sistemas de grande escala, bastante direcionadas aos problemas do dia-a-dia e às redes de energia, surgiu então a ideia de aplicar estes conceitos e métodos a uma escala menor (*low power*, sendo esta entendida como transmissores de rádio, aparelhos eletrónicos de baixa potência usados para telecomunicações e transmissão de informações), de forma a alimentar sensores, recorrendo a fontes de energia de origem biológica, e também de otimizar consumos de energia, a esta mesma escala, com base em algoritmos bio inspirados.

Esta dissertação tem como objetivo tentar uma abordagem diferente de otimização de gastos energéticos em redes de sensores sem fios, formadas por sensores *low power*, e também que a aplicação dessa otimização seja feita em RSSF que sejam alimentadas por sistemas de recolha energética. Desta forma os dispositivos além de possuírem autonomia de alimentação, essa autonomia pode ser aumentada e a necessidade de manutenção para substituição de baterias também é diminuída.

Assim este trabalho incide em duas grandes áreas, a recolha energética do meio ambiente para alimentação de sistemas de baixo consumo e a otimização dos gastos energéticos desses por meio da utilização de algoritmos, neste caso bio inspirados. Ao longo da presente dissertação são abordados esses dois temas bem como muitos outros envolvidos e necessários à sua boa compreensão.

## Palavras-chave

Algoritmos; bio inspirados; recolha energética; otimização; nó sensor; deriva temporal;

**Folha em branco**

## **Abstract**

Currently, the interest in using biological energy sources and those inspired by biological processes has been increasing. Over the years, the demand for self-sustaining and autonomous networks has grown, ranging from small applications such as monitoring sensors to large-scale applications like Industry 4.0. The main advances in publications find application in large-scale systems, highly focused on everyday problems and energy networks. This led to the idea of applying these concepts and methods on a smaller scale (low power, understood as radio transmitters, low-power electronic devices used for telecommunications and information transmission) to power sensors using biologically sourced energy. Additionally, the goal is to optimize energy consumption at this scale based on bio-inspired algorithms.

This dissertation aims to explore a different approach to optimizing energy expenditure in wireless sensor networks consisting of low-power sensors. The focus is on applying this optimization to Wireless Sensor Networks (WSNs) powered by energy harvesting systems. This approach not only provides devices with energy autonomy but also extends autonomy, reducing the need for maintenance to replace batteries.

Therefore, this work focuses on two major areas: harvesting energy from the environment to power low-consumption systems and optimizing the energy expenditure of these systems using bio-inspired algorithms. Throughout this dissertation, these two themes, along with many others essential for a comprehensive understanding, are addressed.

## **Keywords**

Algorithms; bio-inspired; energy harvesting; optimization; sensor node; temporal drift;

**Folha em branco**

# Índice

Declaração de Integridade.....	<b>Erro! Marcador não definido.</b>
Dedicatória .....	v
Agradecimentos .....	vii
Resumo .....	ix
Abstract .....	xi
Keywords .....	xi
Índice .....	xiii
Lista de Figuras .....	xvii
Lista de Tabelas.....	xxiii
Lista de acrónimos.....	xxv
Capítulo 1 - Introdução.....	1
1.1.    Enquadramento .....	1
1.2.    Motivação.....	3
1.3.    Organização do documento.....	4
Capítulo 2 - Conceitos fundamentais e estado da arte.....	5
2.1.    Redes de sensores sem fios .....	5
2.1.1.    Nós sensores .....	6
2.1.2.    Restrições de Energia em Redes de Sensores de Baixa Potência.....	6
2.2.    Recolha Energética .....	9
2.2.1.    Transformações de energia .....	10

2.2.2.	Tipos, fontes de energia e mecanismos de conversão .....	11
2.2.2.1.	Energia da radiação: .....	11
2.2.2.2.	Energia cinética .....	12
2.2.2.3.	Energia térmica.....	15
2.2.2.4.	Energia Rádio Frequência.....	17
2.2.2.5.	Energia química:.....	17
2.3.	Compromissos entre o Custo de Energia e o Desempenho.....	19
2.4.	Sincronismo Temporal e Deriva da Frequência de Relógio .....	19
2.4.1.	RTC.....	20
2.4.2.	Fontes de erros no sincronismo.....	20
2.4.3.	Sincronismo entre relógios .....	21
2.5.	Comunicações segundo a Norma IEEE 1451.....	22
2.5.1.	O que é a norma IEEE 1451?.....	23
2.5.2.	Comunicações segundo a norma IEEE 1451 .....	24
2.5.3.	Organização das comunicações entre aplicação (NCAP) e Sensores (TIM).....	25
2.5.3.1.	O que são TIM .....	25
2.5.3.2.	O que são NCAP .....	31
2.6.	Algoritmos bio inspirados .....	38
2.7.	Resumo.....	40
Capítulo 3 - Formulação do Problema e Descrição do Modelo/Algoritmo Proposto .....		43
3.1.	Formulação do problema .....	43
3.2.	Descrição do modelo .....	48

## Índice

3.2.1.	Formulação teórica:.....	48
3.2.2.	Formulação matemática: .....	49
3.2.2.1.	Modelo de sincronização: .....	49
3.2.2.2.	Modelo de custo energético: .....	51
3.2.2.3.	Modelo de <i>duty-cycling</i> síncrono:.....	52
3.3.	Algoritmo selecionado .....	53
3.3.1.	<i>Particle swarm optimization</i> .....	53
3.3.2.	Formulação do PSO.....	54
3.4.	Modelo.....	57
3.4.1.	Conceitos base a ter em conta:.....	57
3.4.2.	Recolha energética.....	57
3.4.3.	Adaptação da formulação:.....	59
3.4.4.	Adaptação do algoritmo: .....	60
3.4.5.	Definição dos parâmetros: .....	61
3.5.	Caso em estudo: .....	63
3.5.1.	CC2500 – Modos de operação selecionados:.....	65
3.5.2.	Cenários considerados.....	66
Capítulo 4 - Resultados e Validação do Modelo.....		69
4.1.	Resultados e Análise .....	69
4.1.1.	Análise prévia .....	69
4.1.2.	Resultados.....	69
4.1.2.1.	Valores comuns a todas as simulações:.....	70

4.1.2.2. Simulação 1.....	76
4.1.2.3. Simulação 2 .....	81
4.1.2.4. Simulação 3 .....	86
4.1.2.5. Simulação 4 .....	91
4.1.3. Análise dos resultados.....	96
Capítulo 5 – Conclusão .....	105
5.1. Perspetivas futuras .....	105
Bibliografia.....	107
Anexos .....	109

## Lista de Figuras

Figura 1 - Ilustração de uma rede de sensores sem fios, fonte: autor.....	5
Figura 2 - Constituição básica de um nó alimentado por um sistema de recolha de energia, fonte: autor.....	6
Figura 3 – Componentes de um nó sensor, fonte: autor, inspirado em [2] .....	8
Figura 4 – Sistema de recolha e acumulação de energia [5] .....	10
Figura 5 - Fluxograma de desenvolvimento de um sistema de RE, fonte: autor, inspirado em [6] .....	11
Figura 6 - Efeito fotoelétrico, fonte: autor, inspirado em [7] .....	12
Figura 7 - Turbina eólica e principais constituintes, fonte: autor, inspirado em [7] .....	13
Figura 8 - Ligação de vários sensores piezoelétricos, fonte: [6] .....	13
Figura 9 - Sistemas híbridos: a – hidráulico e eólico; b- eólico e solar; c- hidráulico e solar. fonte: autor inspirado em [7] e [8] .....	14
Figura 10 - Energia térmica, fonte: autor .....	15
Figura 11 - Efeito termoelétrico, fonte: autor .....	15
Figura 12 - Ciclo de Carnot Inverso, fonte: autor inspirado em [9], [10] .....	16
Figura 13 - Turbinas a vapor, fonte: autor inspirador em [9] .....	16
Figura 14 - Diagrama de blocos de conversão de energia de ondas eletromagnéticas, fonte: autor inspirado em [7].....	17
Figura 15 - Célula de combustível, fonte: autor .....	19
Figura 16 - Circuito integrado de um rtc, fonte: [11].....	20

## *Lista de Figuras*

Figura 17 - Comportamento do relógio rápido, perfeito e lento [1] .....	22
Figura 18 – Modelo Cliente- Servidor, fonte: autor .....	25
Figura 19 -Diagrama funcional da plataforma IEEE 1451, fonte: autor, inspirado em [12]..	26
Figura 20 - Contexto funcional do TIM, fonte: autor inspirado em [12].....	26
Figura 21 - Blocos funcionais entre a rede e o transdutor, fonte: autor inspirado em [13] ...	26
Figura 22 - Transdutores suportados pelo TIM e modos de amostragem, fonte: autor inspirado em [12].....	27
Figura 23 - Estrutura de dados de um canal do TIM, fonte: autor inspirado em [12].....	29
Figura 24 - Mensagens entre o microprocessador e o TIM, fonte: autor inspirada em [12] ..	31
Figura 25 - Contexto funcional da NCAP, fonte: autor, inspirado em [12] .....	32
Figura 26 - Contexto funcional da microNCAP, fonte: autor, inspirado em [12] .....	32
Figura 27 - Modelo de comunicação cliente-servidor, fonte: autor, inspirado em [12].....	34
Figura 28 - Modelo de comunicação publicador-subscritor, fonte: autor inspirado em [12]	34
Figura 29 – Hierarquia de classes IEEE, fonte: autor inspirado em [13].....	37
Figura 30 - Caracterização de algoritmos de otimização pelos mecanismos que usam, fonte: autor inspirado em [14] .....	39
Figura 31 – Ilustração do sistema, fonte: autor .....	43
Figura 32 - Calendarização das comunicações em ordem a $t_1+x$ , fonte: autor .....	44
Figura 33 - Janelas e derivas temporais, fonte: autor .....	44
Figura 34 - Diminuição da deriva temporal, fonte: autor .....	45
Figura 35 - Comunicações base entre a NCAP e TIM, fonte: autor .....	45
Figura 36 – Troca de comunicações entre NCAP e TIM, fonte: autor .....	46
Figura 37 – Fluxograma de decisão de tarefas consoante energia disponível, fonte: autor ...	47

Figura 38 – Diagrama de blocos da formulação de base, fonte: autor .....	48
Figura 39 - Funcionamento das comunicações base entre a NCAP e as TIM, fonte: autor ..	49
Figura 40 - Fluxograma do PSO, fonte: autor, inspirado em [16] .....	56
Figura 41 - Ilustração dos constituintes de uma função, fonte: autor .....	57
Figura 42 - Exemplo do painel fotovoltaico com medidas, fonte: autor .....	59
Figura 43 - Fluxograma do algoritmo proposto, fonte: autor .....	61
Figura 44 - Fluxograma da primeira parte do código, fonte: autor .....	62
Figura 45 - Troca de mensagens numa comunicação, adaptado de [18].....	63
Figura 46 - Cenários de estudo, fonte: autor.....	66
Figura 47 - Fluxograma do algoritmo, fonte: autor .....	67
Figura 48 – Resultados do Cálculo da Potência total para cada modo de operação .....	70
Figura 49 – Valores das componentes que compõem a potência, para cada modo de operação, para Tim 1.....	71
Figura 50 - Valores das componentes que compõem a potência, para cada modo de operação, para Tim 2 .....	72
Figura 51 - Valores das componentes do modo sleep, para cada modo de operação, para Tim 1.....	73
Figura 52 – Zoom dos valores das componentes do modo sleep, para cada modo de operação, para Tim 2 .....	74
Figura 53 - Zoom dos valores das componentes do modo sleep, para cada modo de operação .....	75
Figura 54 – Valores otimizados de $k \cdot \Theta$ , para cada modo de operação e cada cenário A e B 76	
Figura 55 – Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação .....	77

## *Lista de Figuras*

Figura 56 – Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B.....	78
Figura 57 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM.....	79
Figura 58 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM.....	80
Figura 59 - Valores otimizados de 'k.Θ', para cada modo de operação e cada cenário A e B .	81
Figura 60 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação .....	82
Figura 61 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B.....	83
Figura 62 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM.....	84
Figura 63 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM.....	85
Figura 64 - Valores otimizados de 'k.Θ', para cada modo de operação e cada cenário A e B	86
Figura 65 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação .....	87
Figura 66 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B.....	88
Figura 67 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM.....	89
Figura 68 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM.....	90
Figura 69 - Valores otimizados de 'k.Θ', para cada modo de operação e cada cenário A e B .	91
Figura 70 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação .....	92

Figura 71 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B..... 93

Figura 72 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM..... 94

Figura 73 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM..... 95

**Folha em branco**

## Lista de Tabelas

Tabela 1 – Técnicas para alcançar BP nas RSSF, fonte: autor .....	7
Tabela 2 - Classificação de células de combustível, fonte: autor inspirado em [10] .....	18
Tabela 3 – Fontes de erros no sincronismo e definições, inspirado em [1] .....	20
Tabela 4 - Divisões da norma IEEE 1451, fonte: autor, inspirado em [12] .....	23
Tabela 5 - Bits de estado do TIM, fonte: autor inspirado em [12] .....	30
Tabela 6 - Propriedades básicas de uma entidade, fonte: autor inspirado em [13] .....	36
Tabela 7 - Traços de inspiração de alguns tipos de algoritmos [4] .....	38
Tabela 8 - Pontos fortes e fracos da plataforma IEEE .....	41
Tabela 9 - Tabela de irradiação, retirada de [7] .....	58
Tabela 10 – Cálculos para vários cenários/quantidades de bytes .....	64
Tabela 11 - Cenários de gasto energético selecionados, fonte[17] .....	65
Tabela 12 - Relação entre as energias original e otimizada, simulação 1 - Tim 1 .....	97
Tabela 13 - Relação entre as energias original e otimizada, simulação 2 - Tim 1 .....	97
Tabela 14 – Comparação das relações de energia das simulações 1 e 2, entre o cenário A e B, para TIM 1 e 2 nos dois modos de operação .....	98
Tabela 15 – Relação das diferenças energéticas entre os modos de operação para ambos os TIM, simulação 1 e 2 .....	99
Tabela 16 - Comparação percentual para um mesmo cenário em modos de operação diferentes .....	99
Tabela 17 – Relação das diferenças energéticas entre os modos de operação para ambos os TIM .....	102

*Lista de Figuras*

Tabela 18 - Comparação percentual para um mesmo cenário em modos de operação diferentes para TIM.....102

## Lista de acrónimos

6LoWPAN	Ipv6 Low Power Wireless Personal Area Network
AFC	Alkaline Fuel Cell
BP	Baixa Potência
CA	Corrente Alternada
CC	Corrente Contínua
CAN	Controller Area Network
DMFC	Direct Methanol Fuel Cell
EH	Energy Harvesting
HT-PEMFC	High Temperature PEMFC
HomePNA	Home Phoneline Networking Alliance
IEEE	Instituto de Engenheiros Eletrotécnicos e Eletrónicos
IoT	Internet of Things
MCFC	Molten Carbonate Fuel Cell
MMI	Mixed Mode Interference
NCAP	Network Capable Application Pocessor
NS	Nós Sensor
PAFC	Phosphoric Acid Fuel Cell
PEMFC	Polymer Electrolyte Membrane Fuel Cell
PPM	Partes Por Milhão
RE	Recolha Energética
RF	Rádio Frequência
RFID	Radio Frequency IDentification
RPC	Remote Procedure Call
RSSF	Redes de Sensores Sem Fios
SOFC	Solid Oxide Fuel Cell
TED	Transdutor Eletrónico de Dados
TEDS	Transducer Electronic Data Sheet
TEGs	Thermoelectric Generators
TII	Transducer Independent Internet
TIM	Transducer Interface Module
USB	Universal Serial Bus

**Folha em branco**

# Capítulo 1 - Introdução

## 1.1. Enquadramento

As origens da vida terrestre estão intimamente ligadas à energia disponível, proveniente do meio ambiente. É muito provável que nos tempos primórdios, as primeiras formas de vida na terra tenham aproveitado a energia do ambiente de forma a sobreviverem e a se reproduzirem. Os organismos capazes de aproveitar a energia de forma mais eficiente tinham maior capacidade de sobrevivência e reprodução, o que nos trouxe, com o avanço do tempo, uma evolução e diversificação de diferentes espécies.

Desde a energia solar usada pelas bactérias fotossintéticas à utilização, já mais recente, da energia do vento para mover, por exemplo, as caravelas dos descobrimentos. Atualmente os humanos continuam a dar uso à energia proveniente do meio ambiente, como forma de atender às suas necessidades. No entanto, com a evolução tecnológica e o aumento das necessidades, o uso que é feito desta energia é cada vez mais complexo e variado que primordialmente.

A cada dia são aprimoradas quer as formas de recolha, quer o tipo de energia que utilizamos, desde os combustíveis fósseis às energias renováveis, que vêm a tornar-se cada vez mais, uma parte maior do bolo energético. A recolha energética tem uma grande vantagem pois, em geral, é melhor para todos nós e para o planeta a nível de poluição e exploração, e ainda para que os dispositivos consigam sustentar-se energeticamente a partir de uma recolha de energia proveniente do ambiente. Estes dispositivos, além de usarem uma energia renovável, se estiverem aplicados em locais inóspitos e/ou perigosos à presença humana, diminuem a necessidade de manutenção e aumentam a sua longevidade.

Dispositivos de baixa potência (BP, ou do inglês, *low-power*), no contexto computacional, podem ser assim classificados relativamente à quantidade de energia que utilizam em operação ou em suspensão. Por norma estes dispositivos, designados por nós sensor, são aplicados em redes de sensores sem fios (RSSF). Quando possuem sistemas de obtenção de energia, alcançam uma maior durabilidade pois funcionam de forma autónoma, por longos períodos, sem a necessidade de intervenção técnica.

As RSSF possuem diversas aplicações, entre elas, a monitorização ambiental, em ambiente hospitalar, fabril, na indústria 4.0, entre outros. Resumindo em aplicações de sistemas ciberfísicos e de forma geral em IoT (do Inglês *Internet of Things*, ou Internet das Coisas).

Com o objetivo de alcançar a melhor eficiência energética para as RSSF, a vida útil dos elementos que as constituem deve ser aumentada e deve ser mantido o mote de energia “verde”.

Além disto há a necessidade de manter todos os processos e operações otimizadas, com o auxílio de algoritmos de otimização.

Nesta dissertação são abordados todos estes temas, de modo a tentar produzir um algoritmo que otimize o gasto energético, por redução dos tempos em ativo e de transmissão, pela otimização das janelas temporais em escuta ou em descoberta, e pela diminuição das derivas temporais decorrentes de desvios nos relógios. Visa também a otimização dos processos de comunicação em função da quantidade de energia necessária, dos recursos energéticos disponíveis e dos tempos requeridos para recolha de energia.

## **1.2. Motivação**

A procura crescente por tecnologias BP tem impulsionado grandes avanços na área da IoT. No entanto, estes sistemas continuam a padecer do desafio relativo à eficiência energética, particularmente quando falamos de comunicações entre sensores deste tipo. Logo, a otimização energética dos gastos das comunicações é muito importante para prolongar a vida útil dos dispositivos e com isto reduzir os custos de manutenção e o impacto ambiental por estes causado. Tendo em conta estes fatores, esta dissertação visa explorar e aprimorar algoritmos de otimização, que possam ser aplicados nestes casos, para reduzir os seus consumos e aproveitar a energia disponível de uma forma mais racional e eficiente.

Como se refere no capítulo 2, a eficiência energética é o problema central dos sistemas de IoT baseados em BP, sendo que, atualmente, a maioria destes é alimentada por baterias limitadas que impõem restrições ao desempenho e duração de operação. Desta forma, é indispensável desenvolver soluções de otimização que permitam comunicações eficientes entre sensores, reduzindo o consumo, mas mantendo a qualidade e confiabilidade dos dados transmitidos.

Além da redução dos consumos energéticos podem-se ainda implementar sistemas de recolha de energia, de forma a carregar as baterias e assim fornecer energia de forma momentânea/instantânea e podendo ainda ser armazenada e usada posteriormente. Evitam-se, desta forma, intervenções de manutenção para troca de baterias e providenciam-se fontes alternativas de energia para alimentação dos sistemas.

O objetivo principal desta dissertação passa por investigar e desenvolver um algoritmo de otimização para minimizar os gastos energéticos em comunicações entre sensores de baixa potência (BP), alimentados por um sistema de recolha de energia (RE) de forma a minimizar os gastos energéticos e maximizar o uso da energia disponível, sem nunca descuidar a qualidade das transmissões. Para alcançar este objetivo incidir-se-á sobre a diminuição dos gastos causados pelo alargamento das janelas temporais de comunicação, e encontrando um período melhor para as sincronizações, de forma a combater os desvios de relógio causados pelo espaçamento temporal entre comunicações. Ao diminuir o tempo de escuta consegue-se também aumentar o tempo de recolha energética.

Em suma o objetivo consiste em encontrar uma harmonia entre as variantes referidas de forma a maximizar o uso da energia disponível e minimizar os gastos desnecessários.

### **1.3. Organização do documento**

A dissertação apresentada é composta por 5 capítulos. O primeiro capítulo introduz os temas que vão ser desenvolvidos ao longo deste trabalho. É composto pelo enquadramento, motivação e organização do documento. O seu objetivo é dar uma perspetiva geral do trabalho e da relevância do seu estudo.

No capítulo 2 os conceitos fundamentais são abordados em 6 subcapítulos. No primeiro ocorre o aprofundamento do conhecimento das redes de sensores sem fios, são descritas as suas restrições no que às redes de baixa potencia diz respeito, é introduzido e explicado o conceito de nó sensor. O subcapítulo seguinte tem como objetivo identificar as diferentes formas de recolha de energia que podem ser utilizadas para alimentar os sistemas e enumera os passos necessários para transformar energias. Nos subcapítulos 3 e 4 a questão da energia é comparada com o desempenho obtido e é clarificada a importância do sincronismo temporal dos dispositivos assim como as possíveis fontes de falhas. Já nos últimos dois as normas relevantes são explicadas e o mesmo acontece com o conceito de algoritmos bio inspirados. Este capítulo tem como secção final um pequeno resumo.

O capítulo 3 é utilizado para a formulação do problema e descrever o algoritmo proposto, desta forma são explicados de forma mais detalhada os algoritmos, formulas e modelos utilizados assim como todas as adaptações feitas para obter o resultado final pretendido.

No capítulo 4 são apresentados e explicados os resultados, sendo que esses também têm forma gráfica. Neste local são também retiradas e explicadas as respetivas conclusões da experiência realizada.

O último capítulo é o 5 e nele é feita uma recapitulação do trabalho desenvolvido e são apresentadas as respetivas conclusões.

## Capítulo 2 - Conceitos fundamentais e estado da arte

### 2.1. Redes de sensores sem fios

Uma RSSF é uma rede sensores, interligados que comunicam entre si através de técnicas de propagação, isto é, sem fios, que consiste num conjunto de pequenos dispositivos de baixo custo e de baixa potência, denominados nós sensor. Estes são capazes de recolher informação e comunicá-la, por meio de protocolos de comunicação sem fios, a uma entidade gestora local ou central, para aí serem processados, analisados e até armazenados. Os nós sensor podem ser equipados com vários tipos de atuadores de recolha capazes de medir, por exemplo, temperatura, humidade, pressão, luminosidade, som, entre outros.

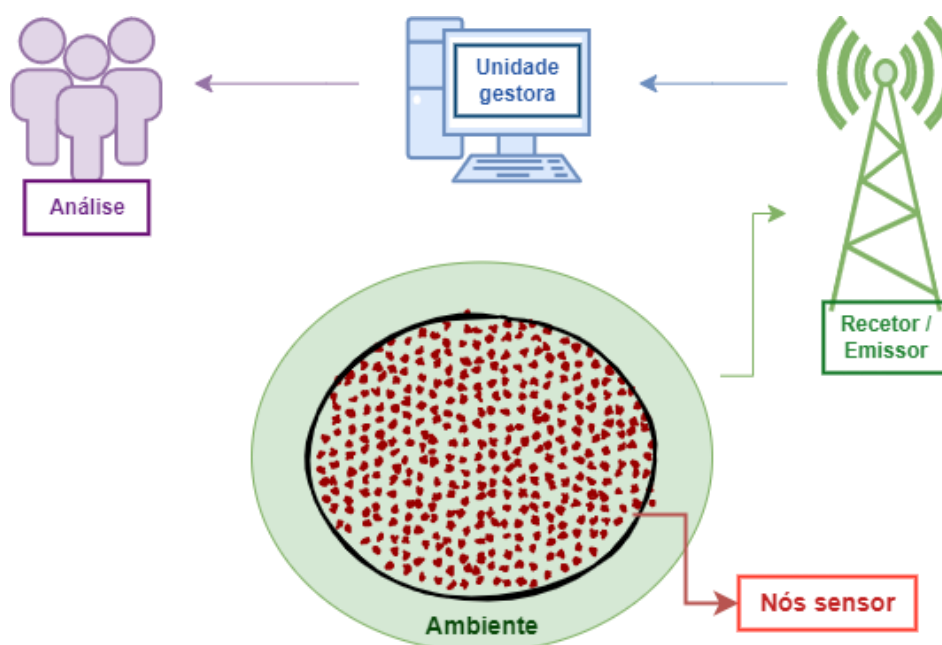


Figura 1 - Ilustração de uma rede de sensores sem fios, fonte: autor

Sendo uma grande parte integrante na IoT, e cada vez mais usadas em aplicações de controlo ambiental, de controlo industrial (indústria 4.0), de automação residencial, monitorização de saúde e sistemas urbanos inteligentes. O seu uso crescente advém da capacidade de fornecer informações, em tempo real, sobre as condições do ambiente/equipamento que está a ser monitorizado. Este fornecimento de informação, quase instantânea, permite otimizar operações, reduzir o consumo energético e até melhorar a eficiência de alguns processos/comunicações.

### 2.1.1. Nós sensores

Um nó sensor é um dispositivo de pequenas dimensões e baixo custo, equipado com um ou vários sensores, de forma a recolher dados do ambiente ou de um sistema físico. Geralmente é constituído por um microcontrolador, um módulo emissor/recetor (transceiver) composto por rádio e antena, uma fonte de energia e um ou mais sensores (Figura 3). O transceiver sem fios permite ao nó comunicar com outros nós da rede e com a central/cluster head. As fontes de energia, dependendo da aplicação, por norma são baterias, ou seja, são sensores de low power com energia limitada e finita. No entanto essa lacuna pode ser colmatada usando técnicas de recolha de energia (RE), vulgarmente conhecidas pela sua designação em Inglês - Energy Harvesting - (EH) (Figura 2), como sejam a energia proveniente da radiação solar, cinética, eólica, entre outras, e que são referidas com mais detalhe no ponto 2.2.2 do documento.

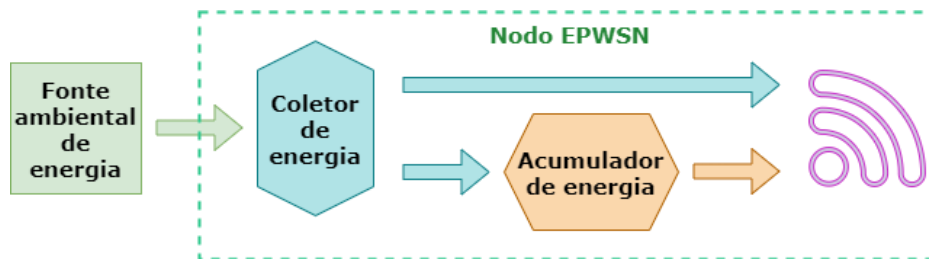


Figura 2 - Constituição básica de um nó alimentado por um sistema de recolha de energia, fonte: autor

No entanto, as RSSF, enfrentam alguns desafios como, por exemplo, recursos limitados de energia e de processamento, pequena faixa de comunicação, além de questões de segurança. Isto leva a que o projeto e a otimização deste tipo de redes necessite de técnicas especializadas, de forma a atender aos requisitos específicos de cada aplicação, podendo assim garantir uma operação confiável e segura. Tem-se ainda o fator do ambiente de operação, que pode impedir ou dificultar a substituição das baterias dos NS, e que pode ser reduzido se houver um sistema de reposição/recolha de energia incorporado em cada um.

### 2.1.2. Restrições de Energia em Redes de Sensores de Baixa Potência

Quando se fala em baixa potência (BP), ou *low power*, significa que os dispositivos com esta característica necessitam de menos quantidade de energia elétrica do que é normalmente necessário para executar uma tarefa ou função. Esta característica é muito útil em situações onde a energia é limitada, ou onde a eficiência energética é uma prioridade, como é o caso de dispositivos móveis ou sistemas alimentados por energia renovável.

O uso de BP pode ser alcançado de várias formas, onde se incluem a otimização de circuitos, usando componentes e tecnologias de BP, reduzindo a velocidade do relógio e implementando estratégias de gestão de energia, que ajustem os consumos energéticos com base nas necessidades das tarefas a realizar e nos fatores ambientais que condicionam a disponibilidade energética, nos casos em que os dispositivos utilizam RE. Num contexto computacional, BP, pode ser a quantidade de energia gasta por um dispositivo enquanto está em operação, no modo de espera ou até em suspensão.

Sendo que os dispositivos que constituem as RSSF acarretam um baixo custo e são de dimensões reduzidas, possuem também baterias com baixa capacidade. Tendo como vantagem a mobilidade dos dispositivos, bem como a operação à distância e de forma autónoma, isto leva a que sejam alimentados por baterias, logo a sua eficiência energética é um dos principais desafios, devido à disponibilidade restrita de energia [1].

No caso das RSSF, o uso de BP é crucial pois, muitas das suas aplicações são em zonas remotas, de difícil acesso ou permanência ao homem, onde a energia é limitada e as operações de substituição das baterias podem ser difíceis, demasiado onerosas ou até perigosas.

Tabela 1 – Técnicas para alcançar BP nas RSSF, fonte: autor

<b>Técnica</b>	<b>Consiste em:</b>
<b>Duty Cycling</b>	Colocar o nó em modo <i>sleep</i> /suspensão de forma a economizar energia. Apenas sai da suspensão em intervalos definidos para recolhas de dados ou comunicações.
<b>Gestão de energia</b>	Ajustar o consumo energético do nó sensor com base nos requisitos da aplicação.
<b>Recolha energética</b>	Usar a energia do ambiente, como a energia solar/térmica, para alimentar o NS. A recolha de energia pode prolongar a vida útil da bateria do nó, ou até mesmo extinguir a necessidade de substituição de baterias.
<b>Componentes de baixa potência</b>	Usar componentes e tecnologias de BP para reduzir o consumo energético dos nós. Isto pode incluir microcontroladores, módulos de comunicação sem fios e sensores com boa eficiência energética.

Existem vários fatores a ter em conta quando pensamos em construir uma rede de sensores sem fios como: a tolerância a falhas, a escalabilidade, os custos de instalação e de configuração, o ambiente de operação, a topologia, o meio de transmissão, os consumos de potência e as restrições de hardware.

A tolerância a falhas é a capacidade de sustentar as funcionalidades de rede sem qualquer interrupção, devido a falhas, dos elementos da mesma. O nível de tolerância a falhas está dependente da aplicação a ser suportada pela rede de sensores, ou seja, diferentes aplicações necessitam de diferentes níveis de precisão. A escalabilidade consiste no número de nós sensores dispostos na área de operação. Por fim, o custo de produção influencia principalmente RSSF que usam muitos sensores, como por exemplo, a monitorização de áreas extensas, nestes casos é importante um baixo custo de produção para tornar aplicação dos sensores viável [2].

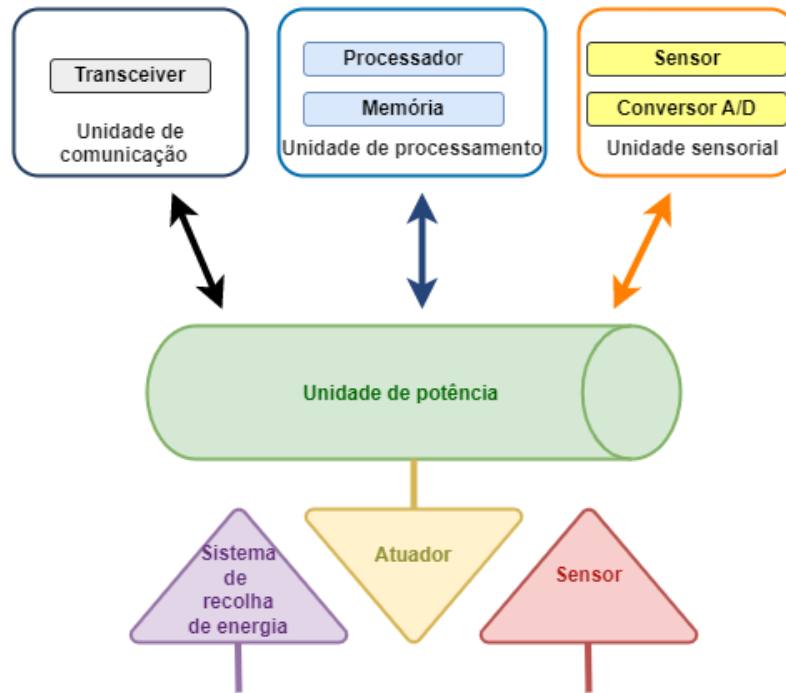


Figura 3 – Componentes de um nó sensor, fonte: autor, inspirado em [2]

É preciso ainda ter em conta outras restrições como por exemplo as limitações ao nível da potência de processamento, a memória, a gama de operação, a largura de banda, a confiabilidade, a segurança e o custo.

Como os nós sensor, por norma, são pequenos dispositivos de baixo custo, com poder de processamento, memória e capacidade de armazenamento limitados, logo não são capazes de executar cálculos complexos ou armazenar grandes quantidades de dados. Possuem também uma limitada faixa de transmissão, que resulta num intervalo restrito de comunicação, que pode afetar a cobertura e conectividade da rede.

As comunicações sem fios em RSSF são propensas a interferências e perdas de sinal, levando a uma comunicação não confiável. Ou seja, pode haver perda, atraso e/ou inconsistência dos dados. São também vulneráveis à segurança, devido às comunicações serem abertas e ao uso de nós.

## 2.2. Recolha Energética

A recolha energética - *Energy Harvesting* - consiste num conjunto de mecanismos que tornam algum equipamento que necessita de alimentação, autossustentável em termos energéticos. Ou seja, são mecanismos compostos por um conjunto de processos que transformam vários tipos distintos de energia em energia elétrica, podendo esta ser consumida no imediato ou armazenada para uso posterior.

No que à disponibilidade de energia diz respeito podem ser consideradas duas formas. Em ambas existe uma necessidade de disponibilidade de carga contínua, o que altera é a fonte. Na primeira condição essa exigência é colmatada na íntegra pelo circuito EH, ele é suficiente para que não se verifique uma falha no fornecimento de energia à carga, sendo fontes auxiliares, tais como, baterias ou conexão com a rede elétrica desnecessárias para o seu bom funcionamento. Estas características levam a uma redução dos custos de manutenção do sistema causado em grande parte pela não utilização de baterias.

Já quando o circuito EH não é suficiente para garantir disponibilidade contínua de energia, seja pela densidade de potência disponível, pelo comportamento de carga ou pela eficiência da conversão de energia, estamos perante uma segunda condição em que meios alternativos de abastecimento de energia, como fontes, não podem ser dispensados[3].

É possível então obter energia a partir de fontes: hidráulicas, eólicas, da força das marés, da decomposição química, vibrações, entre muitas outras. Pode-se ainda acrescentar a estas fontes um método de obtenção de energia bio inspirada, pois a própria natureza é uma produtora de circuitos AC e DC [4].

Esta forma de alimentação é bastante útil e cada vez mais utilizada em vários sistemas de forma a torná-los mais autónomos e menos carentes de manutenção. Induzindo, assim, uma maior vida útil a alguns equipamentos, bem como, a facilidade e aumento das suas aplicações, devido à diminuição da necessidade de manutenção energética destes [4].

Tem como objetivos principais:

- Remover a necessidade de fios de alimentação;
- Eliminar ou reduzir a dependência de fontes de alimentação;
- Aumentar a vida útil de aparelhos e sensores eletrónicos;
- Facilitar a instalação e a manutenção de aparelhos;
- Reduzir custos e desperdícios energéticos.

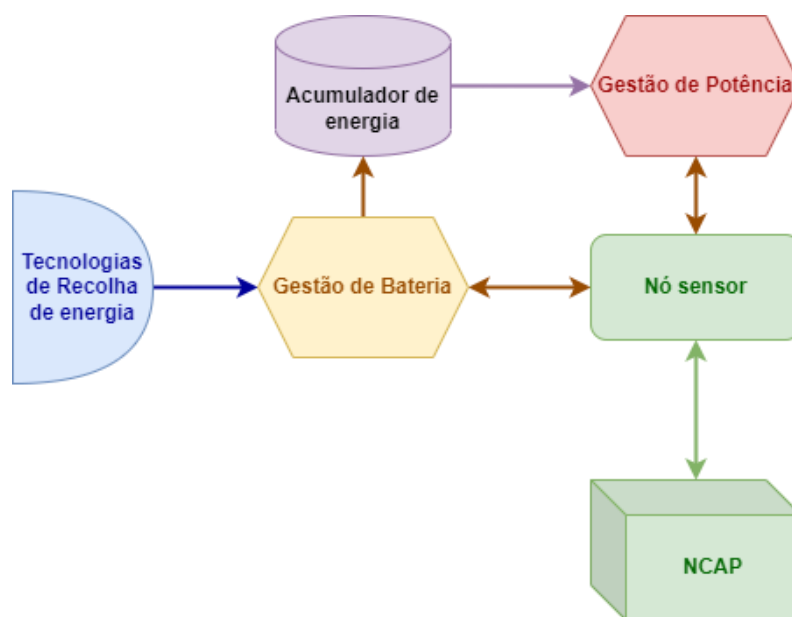


Figura 4 – Sistema de recolha e acumulação de energia [5]

### 2.2.1. Transformações de energia

Para transformar um tipo de energia noutra é necessário um conjunto de passos, sendo estes: uma fonte de onde provém a energia; um mecanismo de conversão; um sistema de armazenamento e, em alguns casos, um sistema de acumulação de energia (Figura 4). Pode-se ainda, estabelecer uma metodologia aplicável ao desenvolvimento base de um sistema de obtenção de energia, dividindo o processo em seis passos [4].

Primeiramente é necessário escolher qual o sistema de obtenção de energia a desenvolver, de acordo com o tipo de fonte que vamos utilizar. Segundo devem-se avaliar os diferentes parâmetros da energia a obter e identificar as condições em que essa obtenção pode ser maximizada, sendo que isso inclui parâmetros como: a distância entre o sistema de obtenção e a fonte de energia primária, a mensuração dessa energia, a orientação dos sensores presentes nos sistemas de obtenção, bem como a quantidade necessária e escolher os que estão aptos a atuar à temperatura ambiente no local onde serão aplicados [6].

Terceiro e quarto correspondem ao desenvolvimento do sistema de armazenamento (de energia) e o de monitorização (baseado em IoT). Em quinto vem a otimização do sistema para a maximização da energia obtida, e assim, o sistema estará concluído, como se pode observar de forma resumida na figura seguinte [4].

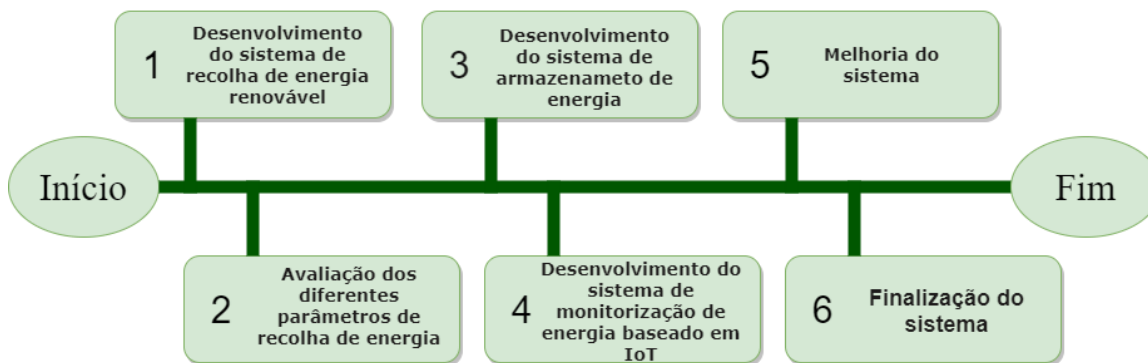


Figura 5 - Fluxograma de desenvolvimento de um sistema de RE, fonte: autor, inspirado em [6]

Atualmente existe uma grande diversidade relativamente aos mecanismos de conversão, sendo que alguns autores os classificam nas seguintes categorias: conversores de vibrações, de energia cinética, hídrica, eólica, de calor e de radiações. Sendo que esta divisão é feita de acordo com o tipo de energia que recolhem e transformam, em energia elétrica.

### 2.2.2. Tipos, fontes de energia e mecanismos de conversão

Os tipos e fontes de energia, são coisas disponíveis na natureza e possíveis de transformar em energia elétrica, assim sendo, todos os tipos de energia referidos a seguir passam por um processo de transformação, realizado por mecanismos eletrónicos, e posteriormente a energia recolhida é gasta e/ ou armazenada para uso futuro. Sendo que existem variados tipos e fontes de energia, são apresentados alguns em seguida, a título de exemplo.

#### 2.2.2.1. Energia da radiação:

A energia proveniente de radiações pode ser recolhida e transformada em energia elétrica, esta energia associada à radiação solar, é uma alternativa renovável o que contribui para a sustentabilidade do planeta. A operação de RE deste tipo, usa como fonte a luz solar e pode ser usada por diferentes tecnologias como o aquecimento solar, energia fotovoltaica e energia heliotérmica (passa primeiramente por um processo de transformação em energia térmica e posteriormente em elétrica). A radiação é recebida por painéis fotovoltaicos, sendo que, quanto maior a incidência da radiação, maior a quantidade de energia elétrica produzida [7].

- **Efeito fotoelétrico**

De forma resumida, a conversão da luz solar em energia elétrica é dada pelo efeito fotoelétrico, que ocorre quando os raios solares colidem com os átomos em repouso no interior do painel solar, que, conseqüentemente movimentam os elétrons, criando assim uma corrente elétrica [7].

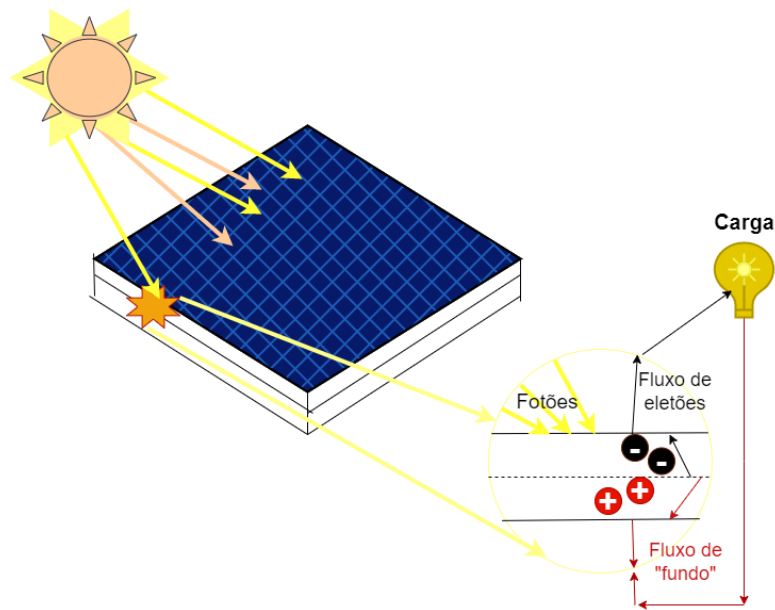


Figura 6 - Efeito fotoelétrico, fonte: autor, inspirado em [7]

#### 2.2.2.2. Energia cinética

A recolha de energia cinética ou do movimento, consiste em recolher algum tipo de energia de algo que se move, como o vento, as vibrações ou a força das marés. Os mecanismos electrocinéticos obtêm energia elétrica a partir de energia cinética. Esta categoria é constituída pelos triboelétricos (fricção), dispositivos nano fluídicos (cinética de iões), biofuelcells, células químicas e microbianas [4].

- **Energia eólica**

Quando falamos em energia do movimento causado pelo vento referimo-nos a energia eólica. Esta é também uma energia renovável e sustentável que é utilizada desde os tempos antigos, designadamente nos moinhos para fazer mover as suas pás e conseqüentemente induzir movimento nas mós, bem como para movimentar embarcações. Nos tempos atuais, continua a ser usada para acionar geradores de energia elétrica [7].

A principal forma de conversão é a utilização de ventoinhas eólicas (Figura 7), estas por sua vez são constituídas por três elementos principais, o rotor, o eixo e o gerador. Consoante o tipo de projeto a implementar possuem elementos secundários variáveis [7]. Este tipo de recolha pode ser anexado ainda a RE de fontes solares (Figura 9-b)

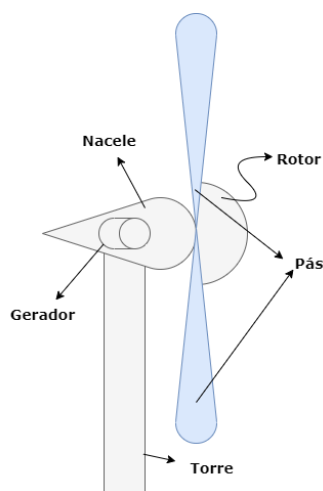


Figura 7 - Turbina eólica e principais constituintes, fonte: autor, inspirado em [7]

- **Piezoelétricidade**

O princípio da piezoelétricidade foi descoberto em 1880 pelos irmãos Pierre e Jacques Curie, no qual perceberam que alguns cristais ao serem submetidos a um dipolo elétrico se deformam dando origem a um diferencial de potencial na saída do circuito [7]. A piezoelétricidade é intrínseca a alguns materiais, que quando submetidos a uma tensão mecânica, tornam-se eletricamente polarizados, gerando assim uma diferença de potencial quando sofrem uma deformação mecânica [7].

Sensores piezoelétricos utilizam o efeito piezoelétrico para medir alterações de aceleração, temperatura, pressão, força ou tensão e converte-os em carga elétrica. O efeito piezoelétrico pode ser encarado como um potencial elétrico, que se mede entre dois lados de um cristal piezoelétrico (Figura 8) quando o material sofre uma extensão ou compressão.

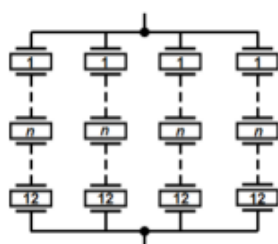


Figura 8 - Ligação de vários sensores piezoelétricos, fonte: [6]

Qualquer uma delas quando aplicada em ciclos (compressão e expansão) induz uma vibração nos sensores piezoelétricos, pode-se por isto designá-los também por conversores de vibrações [4].

- **Energia hidráulica**

A energia hidráulica é proveniente de fontes hídricas e do movimento que estas oferecem, ou seja, transformam a energia das marés em energia elétrica, por indução eletromagnética. Esta indução é provocada pela rotação das hélices do sistema. Sendo que este sistema de recolha pode ainda ser combinado com o sistema da recolha de energia eólica (Figura 9-a) [4], ou por exemplo aliando a RE de fontes, hidráulicas e solares (Figura 9-c).

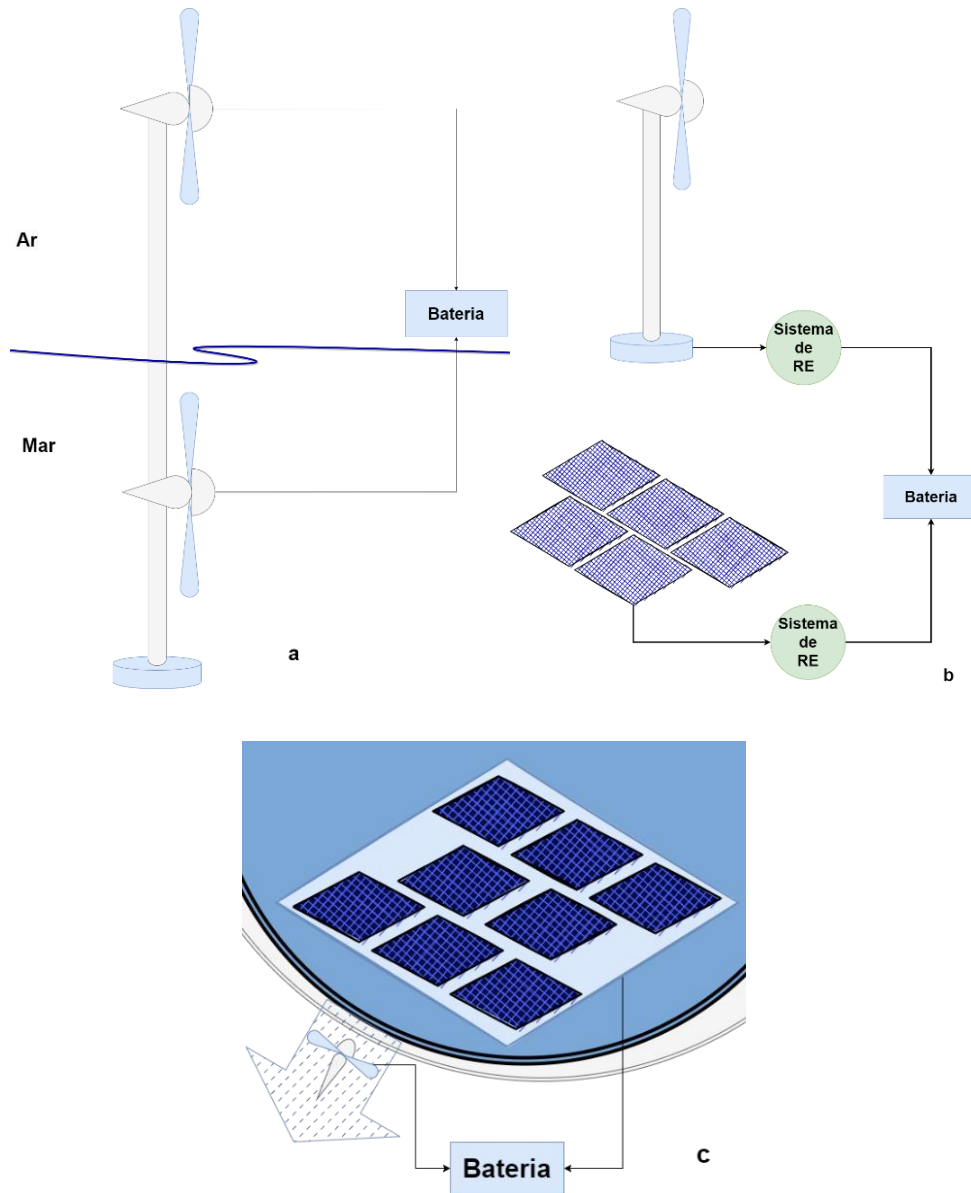


Figura 9 - Sistemas híbridos: a – hidráulico e eólico; b- eólico e solar; c- hidráulico e solar. fonte: autor inspirado em [7] e [8]

### 2.2.2.3. Energia térmica

A energia térmica ou energia calorífica, está presente em diferenças de temperatura entre objetos ou ambientes e podem ser convertidas por dispositivos chamados termo geradores que convertem essa diferença de temperatura em eletricidade. Isso é especialmente útil em aplicações onde há uma fonte de calor constante, como tubulações de calor residual em indústrias.

A conversão pode ser feita por diferentes processos, entre eles são enumerados alguns exemplos como o efeito termoelétrico, que consiste na conversão direta da diferença térmica em energia elétrica, o ciclo de Carnot inverso, que utiliza este princípio para a conversão e as turbinas a vapor, muito usadas para a conversão em centrais termoelétricas.

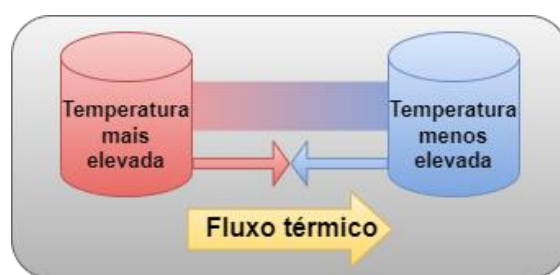


Figura 10 - Energia térmica, fonte: autor

O efeito termoelétrico baseia-se nas propriedades dos materiais termoelétricos, que são capazes de gerar uma corrente elétrica quando expostos a um gradiente térmico, formando os dispositivos, chamados geradores termoelétricos de energia ou TEGs, sendo então o seu uso mais comum as aplicações de RE.

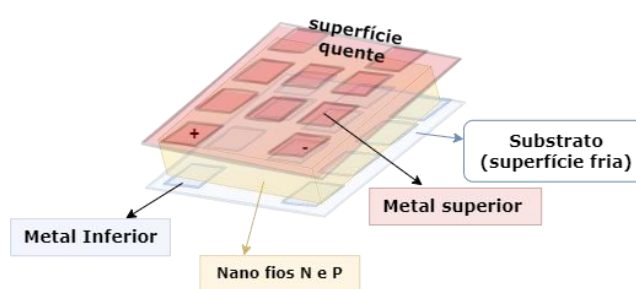


Figura 11 - Efeito termoelétrico, fonte: autor

O ciclo de Carnot inverso é um processo termodinâmico baseado no próprio ciclo que é composto por uma série de etapas, entre elas, a transferência de calor de uma fonte quente para um fluido, a expansão adiabática (totalmente isolada de transferências térmicas) desse fluido, a transferência de calor para uma fonte fria e a compressão adiabática desse fluido. Os dispositivos

que operam com base neste ciclo reverso têm alta eficiência teórica, no entanto são de difícil aplicação prática.

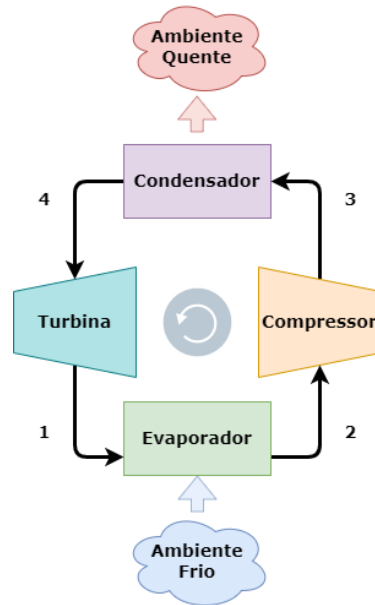


Figura 12 - Ciclo de Carnot Inverso, fonte: autor inspirado em [9], [10]

As turbinas a vapor das centrais, no processo de produção, usam a energia térmica para aquecer a água e gerar vapor de alta pressão. Este é direcionado para as pás da turbina, onde a energia cinética que surge é convertida em energia mecânica. Por sua vez, a turbina aciona um gerador elétrico que converte a energia mecânica em elétrica.

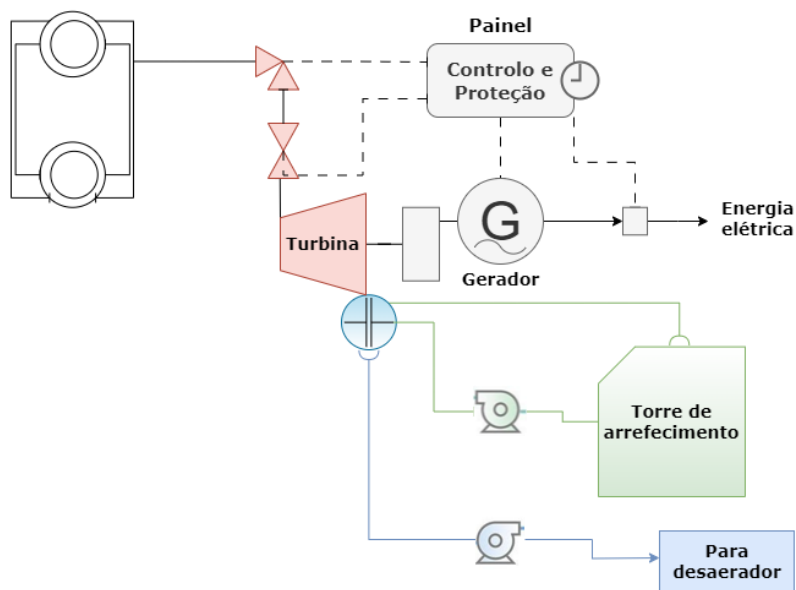


Figura 13 - Turbinas a vapor, fonte: autor inspirador em [9]

Em suma estes são apenas alguns exemplos de como a energia térmica pode ser convertida em energia elétrica e a escolha do método de conversão depende das necessidades de cada aplicação e da eficiência desejada.

#### 2.2.2.4. Energia Rádio Frequência

A energia RF é a energia presente nas ondas eletromagnéticas, como sinais de rádio, Wi-Fi e comunicações. Estas ondas são captadas por antenas especiais, provenientes das estações de TV, FM, *routers* WiFi ou torres de telecomunicações com o objetivo de converter essas ondas eletromagnéticas em corrente contínua para alimentar um dispositivo ou carregar uma bateria[7].

É possível observar na figura seguinte, um diagrama com os três principais blocos de conversão, onde o primeiro, desenhado de forma a melhorar a relação da impedância entre a antena e o diodo, permite maximizar a transferência de potência do sinal captado pela antena e transferi-lo para o diodo. O segundo bloco é formado por um diodo ou pontes retificadoras de diodos para converter ondas eletromagnéticas em corrente contínua. Por fim, no terceiro bloco, o condensador armazena energia e elimina o ripple da onda gerada pelo processo de retificação do diodo [7].

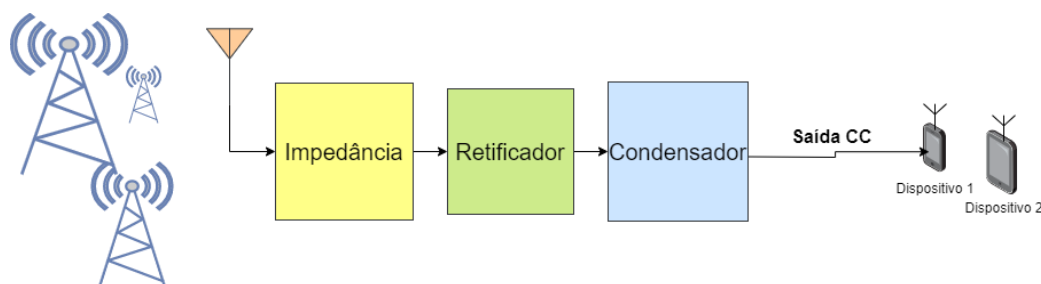


Figura 14 - Diagrama de blocos de conversão de energia de ondas eletromagnéticas, fonte: autor inspirado em [7]

O maior problema de captação e conversão de ondas eletromagnéticas em energia é a baixa densidade de potência disponível no ambiente. A potência disponível na antena depende de vários fatores como frequência, distância entre transmissor e receptor, densidade de potência transmitida, ganho das antenas do transmissor e do receptor e um circuito retificador de RF que seja capaz de captar o sinal de RF e fornecer uma tensão CC suficiente para alimentar uma carga [7].

#### 2.2.2.5. Energia química:

Alguns dispositivos de RE são capazes de aproveitar reações químicas para gerar eletricidade, por exemplo, células de combustível microbianas, usam bactérias para converter compostos orgânicos em eletricidade.

- **Células de combustível e baterias de células de combustível**

Células de combustível são dispositivos que convertem diretamente a energia química em energia elétrica através de reações eletroquímicas. Com base nas pilhas alcalinas surgiram as pilhas de combustível, sendo constituídas por uma célula unitária onde é convertida a energia. Esta por sua vez é constituída por um eletrólito em contacto com um ânodo (eléctrodo negativo) e um cátodo (eléctrodo positivo). Podem utilizar vários tipos de combustíveis como hidrogénio, metanol, etanol e gás natural, combinados com o oxigénio. A classificação das células é feita de acordo com os eletrólitos que as constituem e o combustível usado [10].

Tabela 2 - Classificação de células de combustível, fonte: autor inspirado em [10]

<b>Classificação</b>	<b>Definição</b>
<i>PEMFC</i>	Célula de combustível de membrana de permuta de protões ou membrana polimérica ácida à base de água como eletrólito e eletrólitos catalisados por platina. Utiliza como combustível hidrogénio puro ou gás natural. (<100°C)
<i>HT- PEMFC</i>	É uma PEMFC obtida pela mudança do eletrólito de um sistema à base de água para um à base de ácido mineral. (<200°C)
<i>DMFC</i>	Usa uma membrana polimérica como eletrólito e platina-ruténio como catalisador, este, no ânodo, utiliza diretamente o hidrogénio do metanol líquido como combustível.
<i>MCFC</i>	Usa um sal de carbonato fundido, suspenso numa matriz cerâmica porosa como eletrólito, e como combustível o gás natural (650°C)
<i>PAFC</i>	É constituída por um ânodo e um cátodo feitos de platina finamente dispersa.
<i>Catalisador em carbono</i>	É uma estrutura de carboneto de silício que contém o eletrólito de ácido fosfórico. (<200°C)
<i>SOFC</i>	Usa um eletrólito sólido de cerâmica.
<i>AFC</i>	Usa um eletrólito alcalino e como combustível, o hidrogénio e oxigénio puros.

Estas células são eficientes e têm um baixo impacto ambiental, são por isso usadas em várias aplicações e o seu uso tem vindo a intensificar-se, desde pequenos dispositivos eletrónicos até aos veículos elétricos [10].

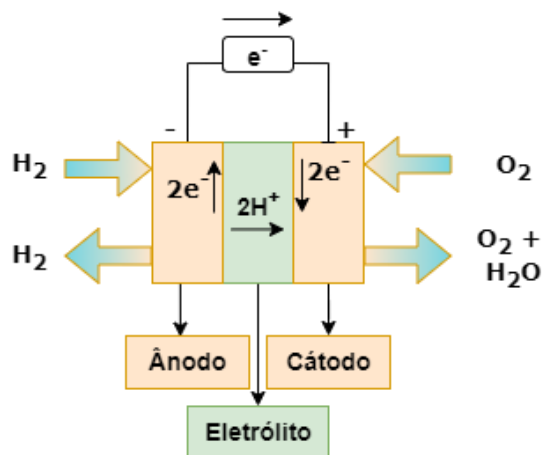


Figura 15 - Célula de combustível, fonte: autor

No caso de serem baterias a células de combustível, combinam o funcionamento das células com a capacidade das baterias de serem carregadas [10].

### 2.3. Compromissos entre o Custo de Energia e o Desempenho

Sabendo então a motivação deste trabalho, pode-se falar no compromisso entre o custo de energia e o desempenho do sistema, pois para funcionar, todo o sistema bem como os programas e funções associadas a este consomem energia. Visto que o objetivo central é diminuir o custo energético não se pode ignorar que, essa diminuição pode apenas ser levada até ao limite que permita ao sistema funcionar em plenas condições de desempenho e viabilidade. Isto é, não pode ser diminuído, o custo energético, para valores que não consigam comportar os gastos necessários ao bom desempenho e rigor. Disto advém o compromisso entre ambos, pois é necessário encontrar um valor intermédio para alcançar os dois.

### 2.4. Sincronismo Temporal e Deriva da Frequência de Relógio

Sincronismo temporal é fundamental nos sistemas constituídos por vários relógios, pois é este que permite compensar os desfasamentos entre os relógios dos diferentes dispositivos, ou seja, cada dispositivo que possui um relógio padece de um erro/desvio associado. Sendo que os dispositivos precisam de comunicar entre si, estes desvios vão causar desfasamentos que em determinado momento podem comprometer o alinhamento necessário às comunicações.

Assim, para combater e diminuir a possibilidade de falhas nas comunicações devido ao desacerto temporal, foram desenvolvidos métodos de sincronismo de forma a manter os relógios dos sistemas minimamente alinhados entre si, reduzindo a influência dos desvios temporais nas comunicações, sendo nesta secção que se incluem os RTCs (*real time clock's* ou relógio de tempo real).

### 2.4.1. RTC

O relógio de tempo real é um dispositivo computacional, por norma um circuito integrado, que indica a evolução temporal. Estes encontram-se em quase todos os dispositivos eletrónicos que padecem da necessidade de manter um controlo preciso do tempo. Estes circuitos são constituídos por um cristal de oscilação que opera numa determina frequência [1].



Figura 16 - Circuito integrado de um rtc, fonte: [11]

O tempo é medido em função do número de oscilações destes cristais num determinado intervalo de tempo, sendo com isto possível fazer uma correlação entre as horas, segundos ou minutos e as oscilações medidas pelo sensor, portanto relacionam o número de ciclos de relógio com o tempo, em segundos normalmente [1].

### 2.4.2. Fontes de erros no sincronismo

Existem diversas causas de erros temporais que dificultam o sincronismo, sendo as não determinísticas as principais fontes que levam ao assincronismo, no entanto existem ainda as determinísticas. Estas são compostas por seis variáveis, sendo elas: o tempo de envio; o tempo de acesso; os tempos de transmissão e receção; os atrasos na propagação e ainda o tempo gasto no processamento da mensagem, na receção [1].

Tabela 3 – Fontes de erros no sincronismo e definições, inspirado em [1]

Fonte de erro	Definição
Tempo de envio	Tempo usado no processo de transmissão de pacotes a partir da camada de aplicação até à camada final (RSSF ou MAC). Este não é determinístico pela imprevisibilidade do processo de carregamento de dados no sistema.

Tempo de acesso	Tempo gasto para encontrar o momento em que a transmissão se pode iniciar. Determinado pelo tráfego da rede e pela sua capacidade de execução na camada final.
Tempo de transmissão	Tempo gasto no envio de todos os dados do pacote. Determinado em função da largura de banda e do tamanho do pacote a ser enviado.
Tempo de receção	Tempo gasto na receção de todos os dados do pacote. Determinado em função da largura de banda e do tamanho do pacote a ser recebido.
Tempo de atraso de propagação	Causado pelo tempo gasto ao transmitir a mensagem desde o nó de origem até ao destino. No caso de ser conhecida a distância entre nós e a velocidade de transmissão do meio ser estável, trata-se de um atraso determinístico. No entanto, em RSSF, a distância entre nós nem sempre é conhecida e em alguns casos pode ser variável.
Tempo de processamento da mensagem de receção	Tempo gasto no envio da mensagem que parte da cada final para a aplicação. Não determinístico devido à imprevisão do processo de desencapsulamento dos dados no sistema operacional.

\*continuação da Tabela 3.

### 2.4.3. Sincronismo entre relógios

Como mencionado acima, cada cristal é responsável pela definição temporal, podendo este variar e marcar o tempo com um erro associado. Por isso é recomendável que todos se relacionem com um mesmo relógio: *masterclock*.

Todos os relógios possuem um desvio temporal associado, portanto é necessário sincronizá-los regularmente. Por norma não possuem muita precisão, por exemplo os que possuem um desvio “ideal” de apenas 0.001% de frequência acumulam, ao fim de um dia, alguns segundos de erro, sendo por isso o desempenho dos relógios medido em partes por milhão [1].

De forma a combater estas imprecisões e manter o baixo custo associado às RSSF, estabelece-se um limite aceitável de imprecisão do relógio, com o qual não se arrisca perder a capacidade de comunicação, fazendo uma sincronização temporal antes desse limite.

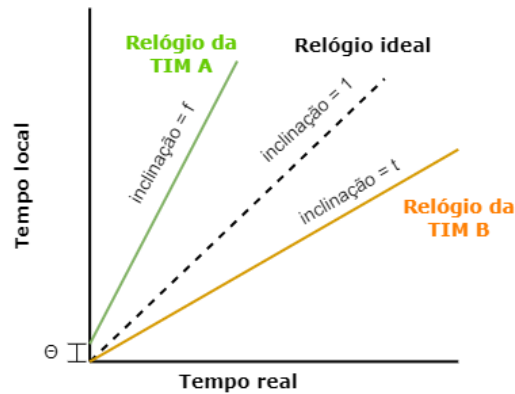


Figura 17 - Comportamento do relógio rápido, perfeito e lento [1]

Neste caso, como em alguns outros é possível estimar o desvio entre o relógio das TIM (*Transducer Interface Modules*) e o da NCAP (*Network-Capable Application Processor*), que serão aprofundados em 2.5.3, podendo assim contorná-lo alargando ou deslocando, de forma otimizada, a janela de comunicação.

## 2.5. Comunicações segundo a Norma IEEE 1451

As normas IEEE (*Institute of Electrical and Electronics Engineers* ou Instituto dos Engenheiros Eletrotécnicos e Eletrônicos) são normas técnicas desenvolvidas de forma a unificar e regular o uso de uma determinada tecnologia. Essas normas são conhecidas e adotadas por todo o mundo promovendo a interoperabilidade e homogeneidade, a segurança e a qualidade em várias tecnologias e sistemas.

O IEEE é uma organização internacional dedicada aos avanços da tecnologia relacionada com eletricidade, eletrônica, computação e áreas relacionadas. Fundada em 1884, é uma das maiores organizações técnicas do mundo, e a sua equipa é composta por engenheiros, cientistas, pesquisadores, profissionais da indústria e estudantes. Tem como objetivo principal promover o avanço tecnológico e a excelência em várias áreas já referidas.

Como organização global, constituem a sua equipa membros de mais de 160 países, oferecendo assim, uma rede de profissionais e pesquisadores de renome internacional. Com tudo isto, desempenha um papel imprescindível no avanço tecnológico, oferecendo uma plataforma de colaboração, partilha e difusão de ideias e de conhecimentos. Em prol de cumprir com este objetivo, o IEEE realiza uma vasta gama de atividades, estando entre elas o desenvolvimento e publicação de normas técnicas, organização de conferências/eventos, publicação de revistas científicas/técnicas e, claro, promove ainda a educação e a formação profissional.

Com as normas que fomenta, promove a definição de padrões técnicos que estabelecem diretrizes e requisitos para várias áreas da indústria. Além disto fornecem também uma base para

pesquisa e desenvolvimento contínuo, incentivando a inovação e colaboração entre profissionais da área a que se refere. Estas normas são revistas e atualizadas periodicamente para se manterem atualizadas relativamente aos avanços tecnológicos e às necessidades de evolução constantes da indústria.

### 2.5.1. O que é a norma IEEE 1451?

A família de normas IEEE 1451 estabelece um conjunto de interfaces normalizadas de hardware e software para interligar sensores/transdutores, microprocessadores e redes de comunicações. Tem como finalidade usar o conceito *plug and play* para o ramo dos transdutores permitindo o funcionamento imediato de um qualquer transdutor com um qualquer microprocessador ligado em rede. Em suma, o objetivo é que ao ligá-los entre eles, se reconheçam automaticamente com o mínimo de intervenção humana possível [12] e [13].

Isto tem a vantagem de evitar as tarefas de configuração manual bem como os erros que daí surgem, enquanto o processo de entrada em funcionamento dos equipamentos é acelerado [12].

Esta plataforma é constituída por uma família de normas/padrões desenvolvidos pelo IEEE para promover, mais uma vez, a interoperabilidade e a comunicação entre sensores e atuadores, em redes de sensores inteligentes. Esses padrões oferecem uma estrutura comum de conexão entre sensores e atuadores a sistemas de monitorização, promovendo a troca de informações de forma consistente e padronizada, independentemente do fabricante ou do tipo de dispositivo.

Esta plataforma inclui, até ao momento, sete normas (Tabela 4), sendo que cada uma foca-se num diferente aspeto das comunicações referidas e seis delas (.1; .2; .3; .4; .5 e .7) focam-se na interação sensor/ transdutor-atuador/microcontrolador [12].

Tabela 4 - Divisões da norma IEEE 1451, fonte: autor, inspirado em [12]

Divisões da norma	Define:
<b>IEEE 1451.0 - Conceitos fundamentais, terminologia e estrutura geral da família de padrões IEEE 1451.</b>	Introduz o conceito de TIM como sendo o periférico que faz a ponte entre os transdutores e os microcontroladores. define as funcionalidades comuns do TIM, a sua memória descritiva conhecida por transdutores eletrónicos de dados (TEDS), e uma interface de software para comunicar com ele.
<b>IEEE 1451.1 - Padrão de interface de rede para sensores e atuadores, especificando protocolos de</b>	Introduz o conceito de NCAP como sendo uma peça de hardware e software que faz a ponte entre os transdutores e a rede de comunicação. A norma define o comportamento da NCAP de acordo com um modelo de

<b>comunicação e formatos de dados.</b>	dados, dois modelos de comunicação, e um modelo de informação.
<b>IEEE 1451.2 - Padrão de interface de rede para sensores e atuadores, especificando protocolos de comunicação e formatos de dados.</b>	Concretiza o conceito de TIM para transdutores ligados ao microprocessador através de uma interface de comunicação ponto-a-ponto com fios.
<b>IEEE 1451.3 - Padrão para as comunicações sem fios entre sensores e atuadores e os seus sistemas de controlo.</b>	Concretiza o conceito de TIM para conjuntos de transdutores ligados ao microprocessador através de uma interface de comunicação multiponto com fios.
<b>IEEE 1451.4 - Padrão para gestão e controlo de sensores e atuadores em redes de sensores inteligentes.</b>	Define as funcionalidades, a memória descritiva e os circuitos de comunicação digital que podem ser adicionados a transdutores analógicos de forma a torná-los compatíveis com a plataforma IEEE 1451.
<b>IEEE 1451.5 - Padrão para acesso à web para sensores e atuadores, permitindo o controlo e a comunicação via internet.</b>	Concretiza o conceito de TIM para transdutores ligados ao microprocessador através de uma interface de comunicação sem fios.
<b>IEEE 1451.7 - Padrão para o acesso à rede de sensores inteligentes, usando Ethernet.</b>	Define estruturas de dados e protocolos de comunicação para dialogar com sensores acoplados a uma interface RFID.

\*continuação da Tabela 4

### 2.5.2. Comunicações segundo a norma IEEE 1451

Esta norma, mais especificamente a parte IEEE 1451.1, aborda comunicações entre sensores e atuadores em redes de sensores inteligentes, isto é, define um padrão de interface de rede, permitindo assim, a comunicação e troca de dados entre esses dispositivos.

O padrão de interface é baseado num modelo de comunicação cliente-servidor onde, o dispositivo (sensor ou atuador) é considerado como o servidor e o dispositivo de controlo/ sistema de aquisição de dados é considerado o cliente, isto numa comunicação de um para um. Em comunicações de um para vários, baseia-se no modelo publicador-subscritor. Além disto, define formatos de dados padronizados para a troca de informações entre os dispositivos, garantindo assim a consistência e a fácil compreensão dos dados pelos sistemas de controlo/aplicações que os recebem [12].

O modelo cliente-servidor (Figura 18) é um padrão de comunicação e interação entre sistemas de computadores, onde estes são divididos em dois grupos com papéis distintos. O cliente é responsável por solicitar serviços ou recursos, enquanto o servidor tem a responsabilidade de receber as solicitações, processá-las e fornecer as respostas adequadas.

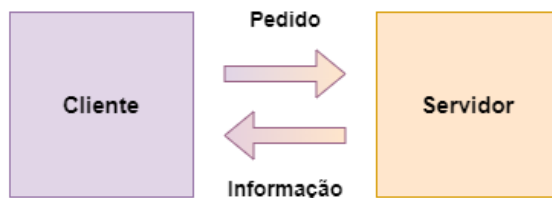


Figura 18 – Modelo Cliente- Servidor, fonte: autor

## 2.5.3. Organização das comunicações entre aplicação (NCAP) e Sensores (TIM)

### 2.5.3.1. O que são TIM

O TIM é um periférico que retém todo o hardware necessário para albergar os transdutores, incluindo circuitos de condicionamento e digitalização de sinal. A norma 1451.0 não impõe quaisquer restrições nestas áreas, tendo o fabricante liberdade total para escolher os seus transdutores e os respetivos circuitos de apoio. Onde a norma 1451.0 incide, é na definição de funcionalidades e tabelas TEDS comuns a todos os TIM de modo a tornar interoperáveis todas as interfaces 1451.X, Figura 19.

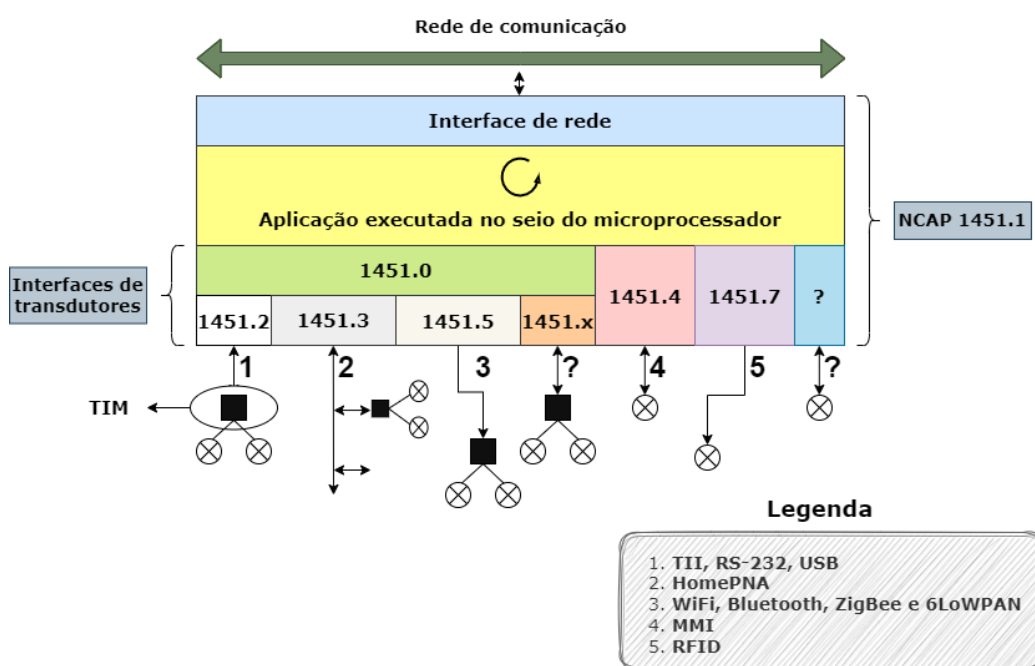


Figura 19 -Diagrama funcional da plataforma IEEE 1451, fonte: autor, inspirado em [12]

O TIM faz então a ponte entre os transdutores e o microprocessador, regido neste caso pelas normas 1451.0, 1451.2, 1451.3, 1451.4, 14.51.5, 1451.7.

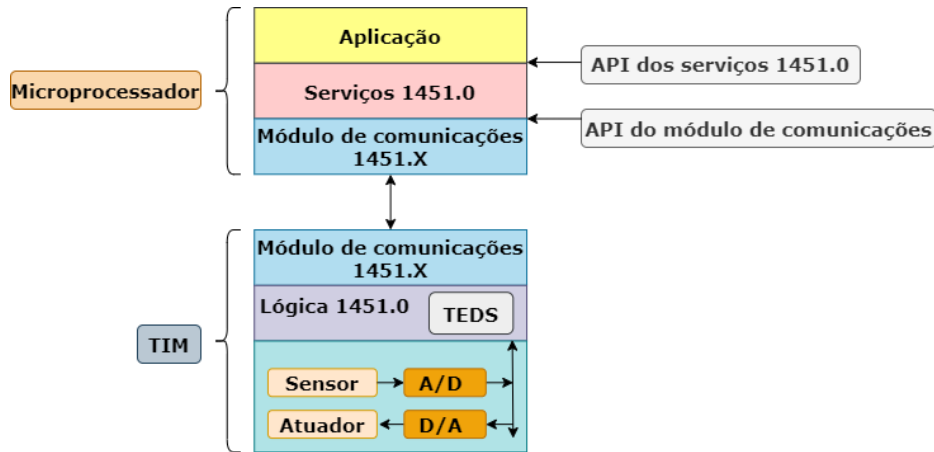


Figura 20 - Contexto funcional do TIM, fonte: autor inspirado em [12]

• **Transdutores**

Um transdutor é um dispositivo capaz de converter energia de um domínio para outro, na mesma ou em diferente forma. Os sinais, assim como a energia, podem ser conduzidos através do elemento transdutor. Um transdutor pode ser conectado a uma rede, a fim de que as informações que fornece possam ser partilhadas com/por outros dispositivos [13].

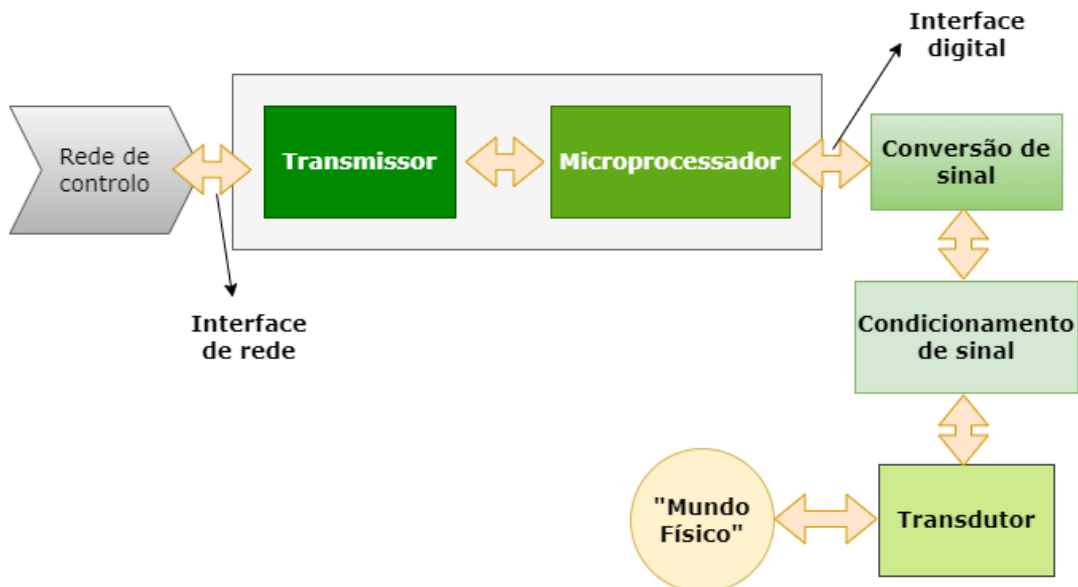


Figura 21 - Blocos funcionais entre a rede e o transdutor, fonte: autor inspirado em [13]

De forma a condicionar o sinal, existem circuitos de condicionamento, que executam funções como amplificação, filtragem, casamento de impedância, modulação e demodulação. Neste seguimento, o sinal precisa ser convertido do domínio analógico para o domínio digital (Figura 21).

Para isto usa-se um conversor analógico/digital (ADC – *Analog to Digital Converter*). O sinal em formato digital pode então ser enviado para uma unidade microprocessadora ou para um microcontrolador que são elementos encarregues de processar os dados. Por último, a informação pode ser enviada para a rede [13].

O TIM suporta três tipos de transdutores que por sua vez suportam cinco modos de amostragem dependendo da sua reação à ordem de disparo (Figura 22):

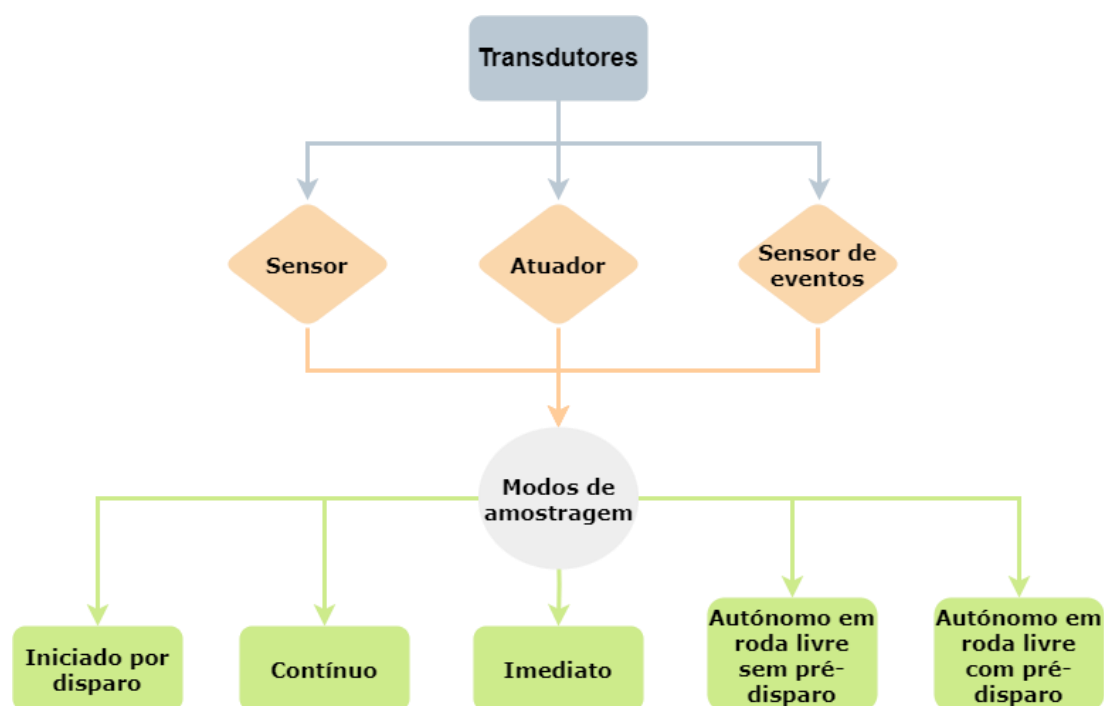


Figura 22 - Transdutores suportados pelo TIM e modos de amostragem, fonte: autor inspirado em [12]

○ **Tipos**

- **Sensor:** adquire uma ou mais amostras de uma grandeza física e devolve o resultado em formato digital. A aquisição das amostras depende de uma ordem de disparo.
- **Atuador:** altera o estado de uma saída de acordo com o valor digital que lhe é enviado. A atuação ocorre após a receção de uma ordem de disparo.
- **Sensor de eventos:** serve para sincronizar uma tarefa do TIM com a ocorrência de um determinado evento, que pode ser o flanco de um sinal digital ou a passagem de

um sinal analógico por um valor de limiar. Depois do disparado, o sensor de eventos fica em modo de espera até detetar o evento a que é sensível [12].

○ **Modos de amostragem**

- **Iniciado por disparo:** O processo de amostragem está parado e só arranca quando é recebida uma ordem de disparo. No caso dos sensores, um número pré-definido de 'n' amostras são adquiridas e transferidas para a memória de saída. No caso dos atuadores, as 'n' amostras armazenadas na memória de entrada são transferidas sequencialmente para a saída. Terminada a transferência, o processo de amostragem volta a parar.
- **Contínuo:** Este modo é idêntico ao anterior, com a diferença de que o processo de amostragem só se extingue por ordem do microprocessador.
- **Imediato:** Este modo também é idêntico ao primeiro, com a diferença de que a ordem de disparo está subentendida nas operações de leitura/escrita.
- **Autónomo em roda livre sem pré-disparo:** O processo de amostragem funciona de forma ininterrupta e autónoma. As amostras adquiridas são simplesmente ignoradas até que uma ordem de disparo seja recebida. Após a receção desta, 'n' amostras são adquiridas e transferidas para a memória de saída.
- **Autónomo em roda livre com pré-disparo:** O processo de amostragem funciona de forma ininterrupta e autónoma. As amostras adquiridas são continuamente armazenadas numa memória intermédia. Após a receção de uma ordem de disparo, 'n' amostras são transferidas para a memória de saída, das quais um número pré-definido 'p' foram adquiridas antes do instante de disparo, e 'n-p' foram adquiridas após o instante de disparo.

○ **Estrutura de dados e registos dos transdutores**

Cada transdutor possui uma estrutura de dados onde é mantida informação relativa ao seu funcionamento. O transdutor e a sua estrutura de dados constituem um canal do TIM.

Define-se o canal 'o' como sendo um canal virtual que representa o TIM como um todo. A estrutura de dados de um canal é constituída por uma memória e três registos de 32 bits. A memória é opcional e serve para armazenar as amostras manipuladas pelo transdutor (se existirem), neste caso a memória será do tipo circular, ou seja, a escrita de valores é feita de forma sequencial e quando esgotada a capacidade da memória, o valor mais recente ocupa o lugar do valor mais antigo. O trio de registos é obrigatório e inclui:

- **Registo de estado:** Assinala as condições de funcionamento do canal. O significado dos *bits* de estado, pode variar consoante o canal represente o TIM (canal 'o') ou um transdutor em particular [12].

- **Registo de eventos:** Memoriza a ativação dos *bits* de estado, se um *bit* de estado assumir o valor '1' e de seguida regressar ao valor '0' (porque a respetiva condição de funcionamento deixou de se verificar), o correspondente *bit* de evento mantém o valor '1', ou seja, memoriza o valor lógico '1' [12].
- **Máscara de interrupção:** Determina os eventos que levam o TIM a interromper o microprocessador. A máscara de interrupção é ligada bit a bit com o registo de eventos de tal forma que o valor 0 impede os bits de eventos de se propagarem para jusante e originarem um pedido de interrupção para o microprocessador. Os bits de pedido de interrupção de todos os canais são ligados para gerar uma única linha de interrupção [12].

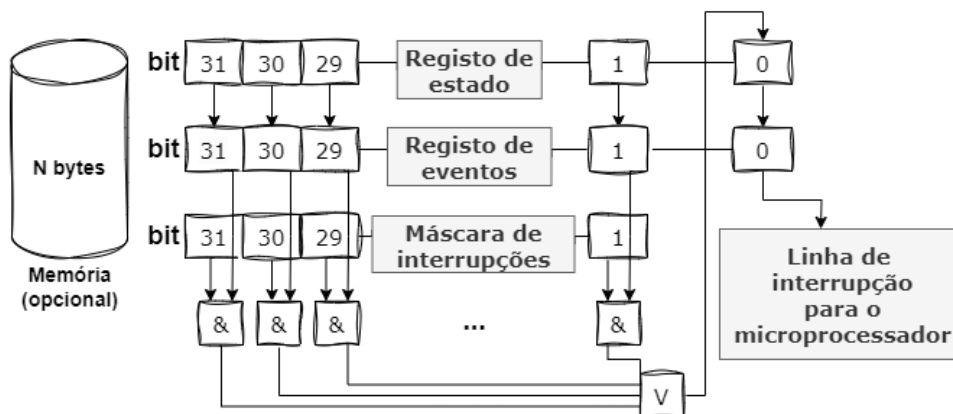


Figura 23 - Estrutura de dados de um canal do TIM, fonte: autor inspirado em [12]

A definição de grupos de canais é muito útil porque permite tratar o grupo como se fosse um único canal. As ações tomadas sobre o grupo afetam todos os canais que o constituem. Um exemplo clássico de aplicação desta funcionalidade é o disparo de um grupo de canais quando é necessário iniciar, em simultâneo, a aquisição de diversas variáveis de campo.

Tabela 5 - Bits de estado do TIM, fonte: autor inspirado em [12]

Bit	TIM (canal o)	Canal do TIM (não canal o)
0	Pedido de interrupção	Pedido de interrupção
1	TEDS modificado	TEDS modificado
2	Comando inválido	---
3	Comando rejeitado	Comando rejeitado
4	Dados ignorados ou evento descartado	Dados ignorados ou evento descartado
5	Dados disponíveis ou evento intercetado	Dados disponíveis ou evento intercetado
6	Falha no hardware	Falha no hardware
7	Não operacional	Não operacional
8	Erros nas comunicações com o microprocessador	---
9	Aquisição ou atuação consumada	Aquisição ou atuação consumada
10	Ocupado	Ocupado
11	Falha na calibração	Falha na calibração
12	Falha na rotina de autoteste	Falha na rotina de autoteste
13	Dados fora da gama	Dados fora da gama
14	Correções desativadas	Correções desativadas
15	Recursos esgotados	Recursos esgotados
16	---	Dados lidos mais que uma vez
17-23	Bits reservados para extensões futuras na norma 1451.0	
24-31	Bits de utilização livre por parte dos fabricantes	

○ **Mensagens:**

A norma 1451.0 demarca que a troca de dados entre o microprocessador e o TIM se faça através de três tipos de mensagens [12]:

- **Mensagem de comando** (microprocessador para TIM): Conjunto de bytes contendo toda a informação necessária para invocar um comando num determinado canal do TIM.
- **Mensagem de resposta** (TIM para microprocessador): Conjunto de bytes correspondentes à resposta do TIM a um comando anteriormente invocado pelo microprocessador.

- **Mensagem criada por iniciativa do TIM** (TIM para microprocessador): Conjunto de bytes análogos à informação não solicitada pelo microprocessador (por norma associada a um pedido de interrupção originado pelo TIM).

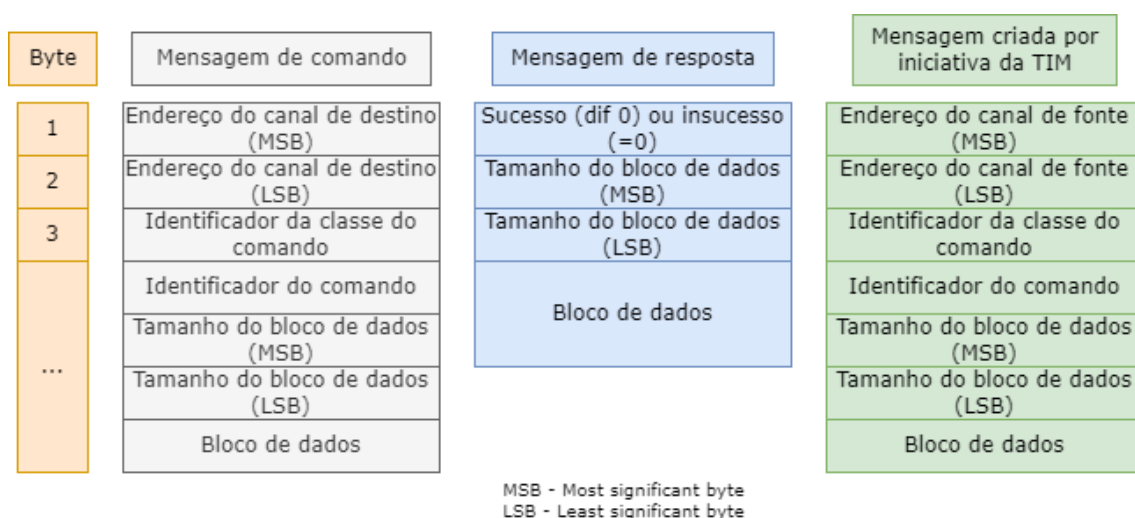


Figura 24 - Mensagens entre o microprocessador e o TIM, fonte: autor inspirada em [12]

#### ○ **Comandos:**

A norma 1451.0 define um vasto conjunto de comandos para interagir com o TI, estes são dirigidos a um canal de destino e são codificados em dois bytes. O byte mais significativo identifica a classe do comando e o byte menos significativo identifica o comando propriamente dito [13].

### 2.5.3.2. O que são NCAP

O NCAP ocupa uma posição de destaque pois é este que faz a ponte entre os transdutores e a rede de comunicação, segundo a norma 1451.1, sendo uma peça de hardware e software que faz a ponte entre os transdutores e a rede de comunicação (Figura 25). A aplicação fornece “inteligência” ao conjunto interagindo com o campo e com a rede de comunicação de acordo com uma lógica pré-programada [12].

A interação com o campo consiste tipicamente na execução de uma estratégia de controlo que recebe dados dos sensores, faz o seu processamento, e gera ordens de comando para os atuadores, ciclicamente e por esta ordem [12].

Do lado do campo implementa funções de alto nível para comunicar com transdutores e do lado da rede implementa funções de alto nível para comunicar com outros processadores. No seu seio, o NCAP executa a aplicação que processa os dados provenientes de ambos os lados (campo e rede) e gera ordens de atuação que decidem o estado seguinte do sistema[13].

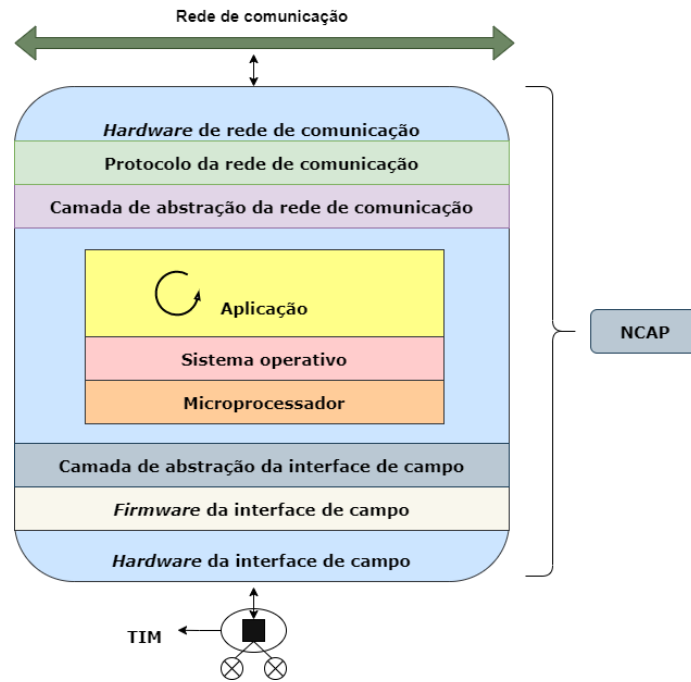


Figura 25 - Contexto funcional da NCAP, fonte: autor, inspirado em [12]

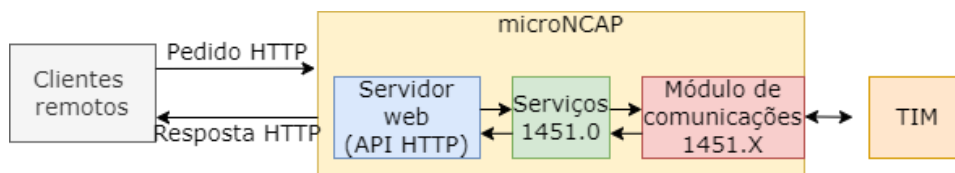


Figura 26 - Contexto funcional da microNCAP, fonte: autor, inspirado em [12]

Toda a arquitetura de software do NCAP é baseada em quatro modelos: um modelo de dados, dois modelos de comunicação, e um modelo de informação OO [12].

○ **Modelo de dados**

O modelo de dados da norma 1451.1 define os tipos de dados utilizados nas comunicações entre objetos, tanto os que residem no seio do NCAP, quer os que se encontram distribuídos pela rede. Este modelo suporta os tipos elementares seguintes [12] :

- Booleano (verdadeiro ou falso);
- *Byte* (combinação de oito bits que não deve ser interpretada como número);
- Inteiro (palavra binária interpretada como número inteiro, com ou sem sinal, que o tamanho varia entre 8 e 64 bits);
- Real (palavra binária interpretada como número real, de precisão simples ou dupla);
- *String* (conjunto de *bytes* interpretado como uma cadeia de caracteres).

Existe ainda um tipo de dados especial, que pode ser usado para transportar qualquer outro tipo de dados [12] :

- *Argument* (suporta estruturas e *arrays* do tipo elementares para representar grandezas mais complexas, tais como instantes de tempo, sistemas de coordenadas ou formas de onda).

- **Modelos de comunicação**

- **Cliente – servidor:**

Os modelos de comunicação são os referidos anteriormente em 2.5.2, primeiramente, o modelo cliente/servidor segue o paradigma RPC, segundo o qual um dado NCAP (o cliente) pode requerer remotamente uma função de outro NCAP (o servidor) e utilizar os resultados devolvidos [12].

Passo a passo do modelo, correspondente à Figura 27 :

1. Chamadas remotas: Porto de comunicação específico (porto cliente), o qual tem de ser previamente associado ao endereço de despacho do objeto servidor.
2. O cliente invoca a função *Execute* do “porto cliente”, indicando de forma explícita a operação que deseja executar no contexto do servidor, acompanhada dos respetivos parâmetros de entrada e de saída, ambos representados por ‘*arrays*’ do tipo ‘*argument*’.
3. A infraestrutura de rede invoca a função ‘*Perform*’ do objeto servidor.
4. A função ‘*Perform*’ reencaminha a chamada remota para a operação indicada pelo cliente fornecendo-lhe os parâmetros de entrada.
5. A operação é executada e os resultados são devolvidos para a função ‘*Perform*’ na forma de parâmetros de saída.
6. A infraestrutura de rede retorna os resultados para o “porto cliente”.
7. A função ‘*Execute*’ devolve os resultados ao cliente.

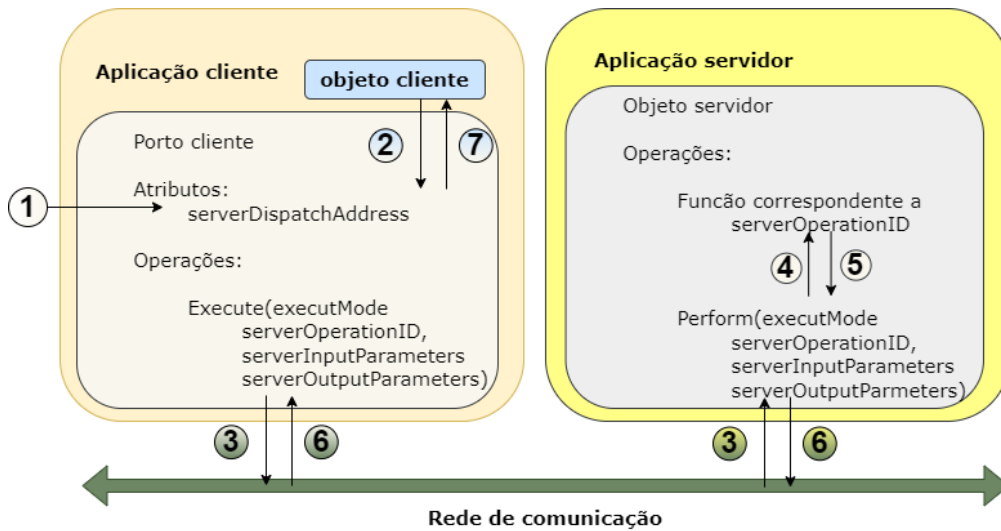


Figura 27 - Modelo de comunicação cliente-servidor, fonte: autor, inspirado em [12]

○ **Publisher – subscriber:**

O modelo publicador - subscritor segue o paradigma das mensagens, segundo o qual um dado NCAP (o publicador) pode emitir mensagens para a rede, sempre que tiver algo para enviar. Estas mensagens, denominadas publicações, contêm meta dados que descrevem o domínio, a semântica e a sintaxe da informação enviada. Com base na meta informação transmitida, qualquer NCAP remoto (subscritor) pode filtrar as publicações que são do seu interesse e usar o seu conteúdo [12].

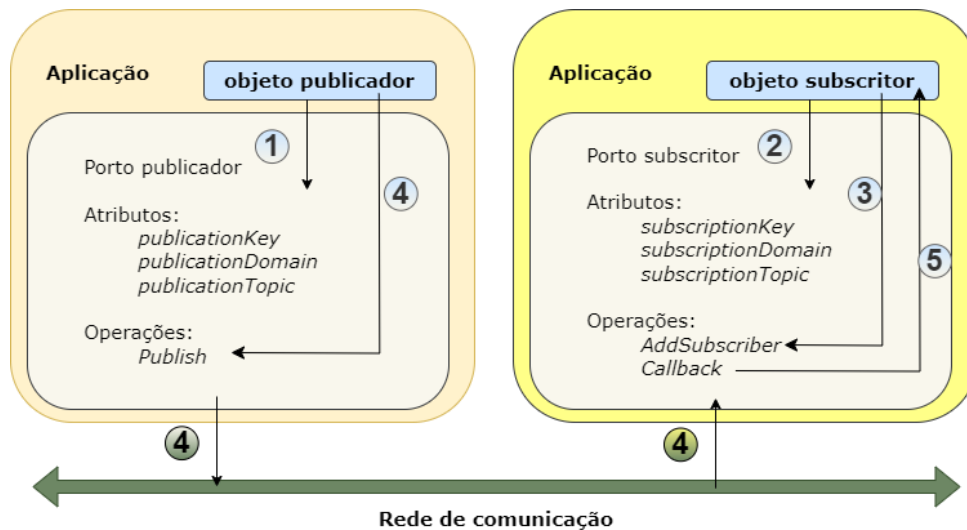


Figura 28 - Modelo de comunicação publicador-subscritor, fonte: autor inspirado em [12]

Este modelo pode ser descrito passo a passo, pela Figura 28 :

1. O publicador define os meta dados da publicação usando um porto de comunicação específico (porto publicador).
2. Paralelamente, o subscritor configura os meta dados da subscrição com recurso a um porto de comunicação específico, denominado “porto subscritor”.
3. O subscritor invoca a função *AddSubscriber* do porto subscritor para registar uma *callback* que será chamada sempre que uma publicação for filtrada.
4. O publicador emite uma publicação invocando a função *Publish* do porto publicador. O conteúdo da publicação é passado sob a forma de um *array* do tipo *argument*.
5. O porto subscritor vigia todas as publicações em trânsito na rede, mas filtra somente aquelas cujos meta dados coincidem exatamente com os seus (os da subscrição). Se a publicação passar no filtro, o porto subscritor invoca a *callback* previamente registada passando-lhe o conteúdo da publicação.

- **Modelo de informação OO**

O modelo de informação da norma 1451.1 é constituído por uma hierarquia de classes divididas em [12]:

- Blocos: Classes orientadas para adquirir e processar informação.
- Componentes: Classes orientadas para encapsular dados.
- Serviços: Classes orientadas para lidar com as comunicações da rede, incluindo tarefas de sincronização com outros processadores.

Todos os blocos, componentes e serviços partilham uma característica comum: são também entidades pois herdam da classe abstrata *Entity*, a qual define propriedades básicas para todos os objetos com visibilidade de rede, como se pode observar na Tabela 6.

Tabela 6 - Propriedades básicas de uma entidade, fonte: autor inspirado em [13]

<b>Propriedade</b>	<b>Tipo</b>	<b>Acesso</b>	<b>Unicidade de valores</b>	<b>Descrição</b>
<b>ID</b>	<i>Array</i> de bytes	Só de leitura	Obrigatória no contexto da rede	Identificador único que distingue sem ambiguidade a entidade no contexto da rede.
<b>Nome</b>	<i>String</i>	Só de leitura	Obrigatória no contexto da rede	Nome dado à entidade aquando da sua instanciação.
<b>Endereço de despacho</b>	<i>String</i>	Só de leitura	Obrigatória no contexto da rede	Endereço único que localiza sem ambiguidade a entidade no contexto da rede.
<b>Tag</b>	<i>Array</i> de bytes	Leitura/escrita	Recomendada	Etiqueta atribuída à entidade para facilitar a configuração do sistema. A <i>tag</i> serve também de pseudónimo para o endereço de despacho.

O modelo de informação é extensível permitindo a adição de novas classes para satisfazer os requisitos de uma aplicação particular.

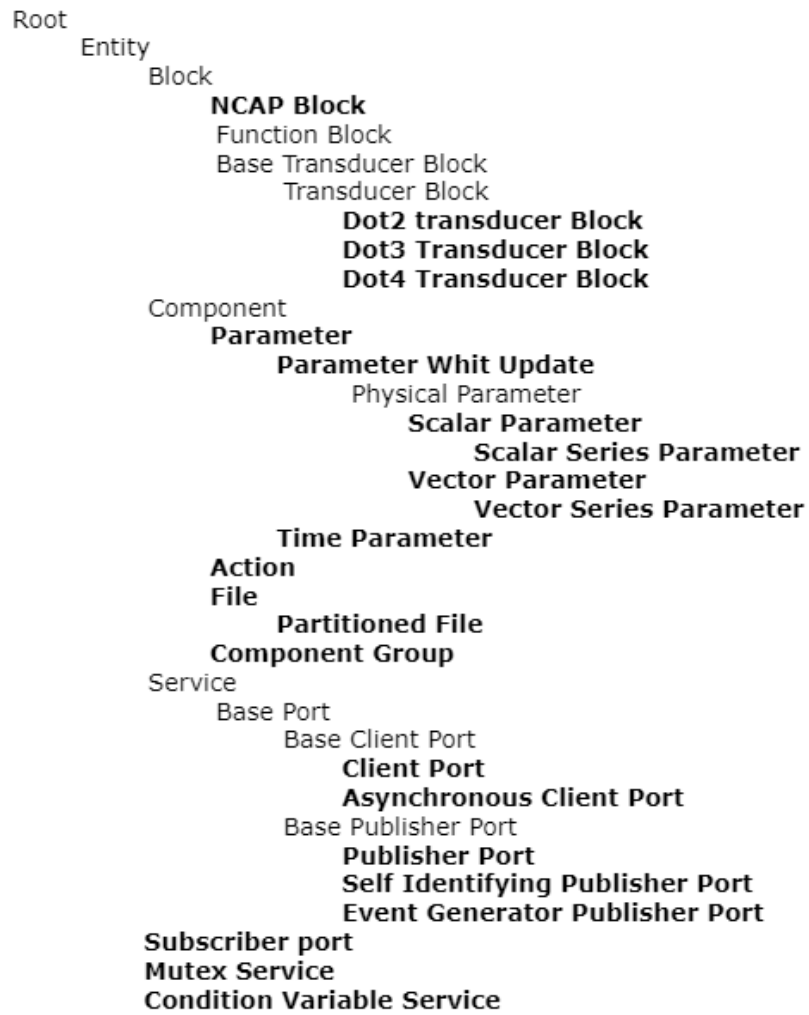


Figura 29 – Hierarquia de classes IEEE, fonte: autor inspirado em [13]

## 2.6. Algoritmos bio inspirados

Algoritmos bio inspirados consistem num conjunto de métodos computacionais inspirados em sistemas biológicos, como evolução, genética, redes de neurónios e inteligência de enxame (*swarm*). São projetados para resolver problemas de otimização complexos e de difícil solução, recorrendo a métodos tradicionais.

Pode-se observar os traços gerais característicos de cada tipo de algoritmo na tabela seguinte.

Tabela 7 - Traços de inspiração de alguns tipos de algoritmos [4]

<i>Tipo de algoritmo</i>	<i>Traços gerais de inspiração</i>
<i>Genéticos</i>	Processo de seleção natural e processo de evolução para otimizar soluções
<i>Redes neuronais</i>	Usa a estrutura e função cerebral para problemas de reconhecimento, classificação e regressão de padrões.
<i>Otimização de colónias</i>	Usa o comportamento de colónias para problemas de otimização de vários fatores, como por exemplo, otimização de caminhos.
<i>Otimização de enxame de partículas</i>	Inspira-se no movimento de bandos (pássaros) ou cardumes (peixes) para otimizar problemas com muitas variáveis.

Este tipo de algoritmos é utilizado em diversas aplicações, com sucesso, em várias áreas como engenharia, finanças, assistência médica, transporte, etc. De forma geral os algoritmos classificam-se em três grupos, os heurísticos, meta-heurísticos e híper – heurísticos (**Erro! A origem da referência não foi encontrada.**).

Os heurísticos usam uma estratégia de resolução de problemas baseada em técnicas práticas ou empíricas para encontrar uma solução que não é garantida como ideal, mas é suficiente para o problema em questão. De forma resumida é um atalho prático que pode ajudar a resolver um problema de maneira rápida e eficiente, sem exigir uma análise completa de todas as soluções possíveis. Por norma são usadas em situações em que o espaço do problema é muito grande ou demasiado complexo para permitir uma pesquisa completa de todas as soluções possíveis. Podem assim ser usados para orientação da procura de soluções, identificando as áreas / espaços de análise mais promissores de encontrarmos a melhor solução. São amplamente usados em campos como ciência da computação, matemática, economia, entre outra, onde podem resolver problemas como otimização, tomada de decisões e raciocínio.

Meta-heurísticos são uma classe de algoritmos projetados para resolver problemas de otimização difíceis de uma forma mais eficiente, relativamente as técnicas tradicionais. Podem ser aplicados a uma ampla gama de problemas sem exigência de conhecimento detalhado do problema, pois são métodos gerais de solução. Usam uma abordagem heurística para explorar o espaço de pesquisa e encontrar soluções ótimas ou quase ideais. Não garantem encontrar a melhor solução, mas são projetados para encontrar uma boa solução dentro de um período razoável. Exemplos destes são algoritmos genéticos, otimização de colônias de formigas, otimização de enxame de partículas entre outros. São amplamente utilizadas em vários campos, como engenharia, ciência da computação, pesquisa operacional e finanças, onde podem ser usadas para resolver problemas complexos, como agendamento, monitorização, alocação de recursos, e muitos outros. Uma das suas vantagens é a capacidade de lidar com problemas de otimização complexos, e a flexibilidade ao nível da representação de problemas e estratégias de pesquisa.

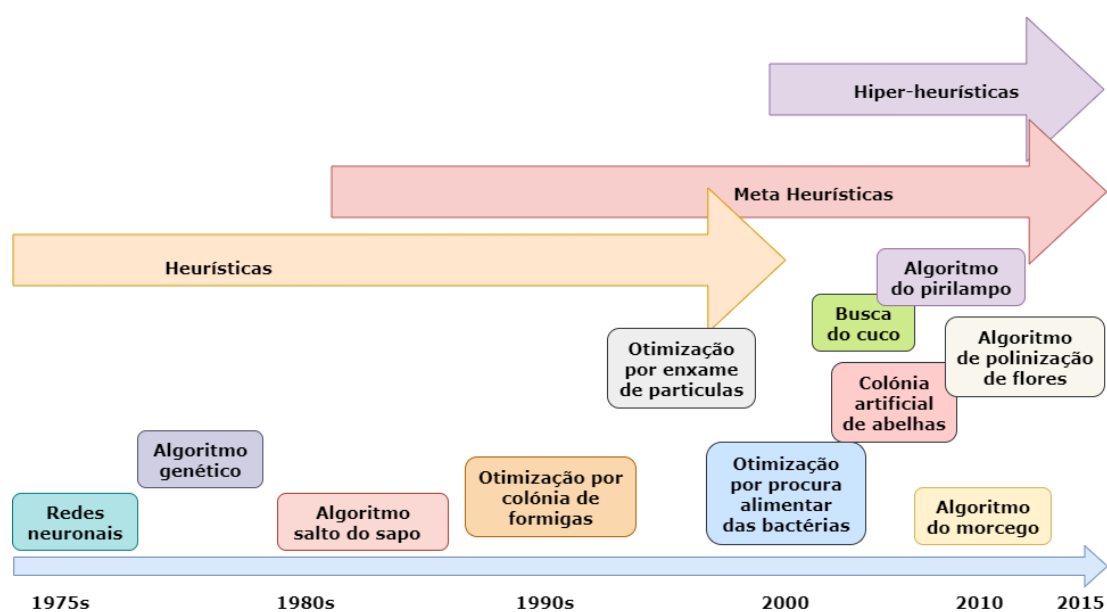


Figura 30 - Caracterização de algoritmos de otimização pelos mecanismos que usam, fonte: autor inspirado em [14]

Por fim, os híper-heurísticos derivam da meta heurística, usam um processo de pesquisa para gerar, selecionar e combinar automaticamente soluções meta-heurísticas de nível inferior, de forma a encontrar uma solução ideal ou quase ideal para o problema. Ou seja, um híper-heurístico é um algoritmo de pesquisa que opera além das meta-heurísticas existentes usando-as como blocos de construção para criar estratégias de pesquisa. Em vez de selecionar um único meta heurístico para um problema, seleciona e combina várias meta heurísticas para criar uma estratégia de pesquisa mais eficaz. No geral, as híper heurísticas são uma área promissora de pesquisa em otimização e inteligência artificial e têm o potencial de melhorar o estado da arte em muitas aplicações [4].

## **2.7. Resumo**

Uma rede de sensores sem fios recolhe e transmite dados para uma entidade central. São utilizadas na IoT em diversas aplicações, como controlo ambiental, monitorização de saúde, sistemas urbanos inteligentes, automação industrial e residencial. Tem uma grande vantagem devido à grande aplicação em locais de difícil acesso, mas enfrentam grandes desafios tais como a limitação de energia devido à grande dificuldade, geralmente encontrada, na substituição de baterias dos nós sensor, processamento e segurança.

Os nós sensor têm restrições de processamento, memória e largura de banda, tornando cálculos complexos e armazenamento de dados volumosos impraticáveis. Comunicações sem fio nas RSSFs são suscetíveis a interferências e perdas de sinal, e a segurança é uma preocupação devido à natureza aberta das comunicações e uso de nós.

Desta forma tanto a utilização de energia como a sua recolha são cruciais para a eficiência do sistema. Por um lado, protocolos eficientes no consumo de energia prolongam a vida útil do sistema, por outro, técnicas adequadas (eficientes) de recolha de energia podem aumentar essa mesma vida útil. Entre as variadas fontes existentes, tem-se fontes de energia: hidráulicas, eólicas, da força das marés, da decomposição química, vibrações, entre muitas outras. Com estes recursos é possível tornar um equipamento que necessita de energia, autossustentável em termos energéticos.

Claro que esse tipo de transformação de energia tem necessidade de alguns passos, sendo eles: uma fonte de onde provém a energia, um mecanismo de conversão, um sistema de armazenamento e em alguns casos um sistema de acumulação de energia. Essa recolha de energia pode ser adaptada conforme a necessidade do sistema ou a disponibilidade/geografia do local de aplicação.

Quando o método utilizado é o mais indicado é possível reduzir custos energéticos e em alguns casos impedir as falhas no sistema e redução do desempenho, originadas/ocorridas pelas falhas ou escassez de energia. A uma melhor gestão de energia acresce a disponibilidade da mesma, o que pode levar a um maior gasto, para a resolução de problemas como é o caso sincronismo temporal, sendo que também este, quando bem usado, pode permitir uma melhor gestão energética.

Tendo em conta tudo o que foi escrito nas secções anteriores, é oportuno fazer neste momento um balanço sobre os pontos fortes e fracos da plataforma IEEE 1451:

Tabela 8 - Pontos fortes e fracos da plataforma IEEE

Pontos fortes	Pontos fracos
A plataforma IEEE 1451 tem a ambição de integrar uma vasta gama de transdutores, tendo assim como ideia de fundo tornar todos os transdutores <i>plug and play</i> , sendo estes analógicos ou digitais, possuindo ou não fios de ligação ou ainda sendo ponto-a-ponto ou em rede.	Pouca intrusão no mercado devido á baixa oferta de suporte técnico e comercial (definida como uma solução demasiado académica e complexa).
É genérica ao nível do sistema operativo e neutra no que diz respeito a tecnologias de rede.	-----
Permite implementação de algoritmos e estruturas lógicas de autodiagnóstico, autoidentificação e autocalibração, acrescentando assim inteligência ao nível do campo.	-----

Sumarizando, a plataforma IEEE 1451 está em constante estudo e atualização, espera-se em breve uma publicação da norma IEEE P1451.639 que concretizará o conceito de TIM para conjuntos de transdutores ligados ao microprocessador através de uma rede CAN (*Controller Area Network*) [12].

Relativamente às normas já publicadas, elas seguirão o seu processo normal de revisão no sentido de se aproximarem das necessidades do mercado. O foco central da plataforma IEEE 1451 situa-se entre o microprocessador e os transdutores pois é aí que incide a maior parte das suas normas [12].

**Folha em branco**

## Capítulo 3 - Formulação do Problema e Descrição do Modelo/Algoritmo Proposto

### 3.1. Formulação do problema

O problema consiste em aumentar a eficiência energética de uma RSSF, constituída por nós sensor alimentados por sistemas de RE. Relativamente à forma como alcançar o objetivo desejado, existem vários pontos que esta formulação tem em conta, sendo eles, a calendarização das comunicações e a minimização da deriva temporal das janelas de comunicação (Figura 34) estabelecendo ainda condições de sincronismo apropriadas e calendarizadas.

O sistema é constituído por uma rede de sensores e uma aplicação que assegura a interface entre as tarefas agendadas e os transdutores. Estes últimos funcionam como sensores (recolha de dados) e/ou atuadores (realização de tarefas). Essa interface é realizada pelo NCAP (*Network-Capable Application Processor*) que é responsável pela interligação e gestão de um ou vários módulos de TIM (*Transducer Interface Modules*) ligados entre todos (Figura 31).

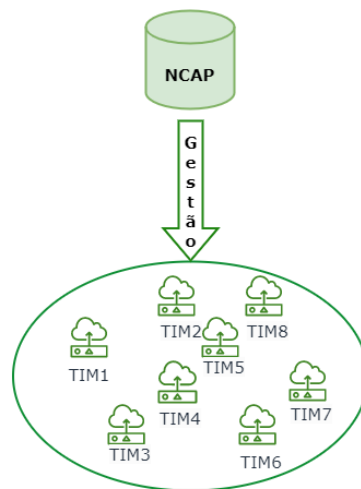


Figura 31 – Ilustração do sistema, fonte: autor

Outro fator muito importante a ter em conta é a própria entidade gestora, e que cada um dos sensores constituintes do sistema terão um *clock* e todos eles estarão a fazer uma contagem pelo seu, tal como cada um terá um erro associado. Sabe-se também que o erro dos *clock* é diretamente proporcional ao tempo decorrido desde a última sincronização, portanto, quanto mais tempo o sensor está em modo *sleep*, maior erro terá quando acordar para a sincronização/comunicação agendada. Além disto existem outros dois fatores que interferem no aumento do erro como os processos de fabrico e os fatores ambientais. A entidade gestora terá de ter acesso a esses dados

de forma a contornar possíveis desfasamentos temporais aquando das comunicações com os sensores.

O aumento dos erros entre os relógios da NCAP e dos TIM gera um aumento do desfasamento entre eles e conseqüentemente um maior gasto energético porque, a aplicação, no tempo  $t_1$  agenda uma comunicação com um sensor para  $t_1+x$  (Figura 32), se este valor 'x' for elevado (dentro dos termos do desvio) o erro de *clock* deste sensor irá ser grande/maior e isto fará com que a janela temporal, que irá ser aberta para a comunicação, tenha de ser aumentada de forma a colmatar o erro induzido pelos fatores referidos. Sendo esta janela maior, um determinado valor, tanto antes como depois do momento programado, como na Figura 33, haverá um período em que o sensor estará a gastar energia desnecessária, pois estará pronto para comunicar quando não existem comunicações, observável no canto superior esquerdo da Figura 36.

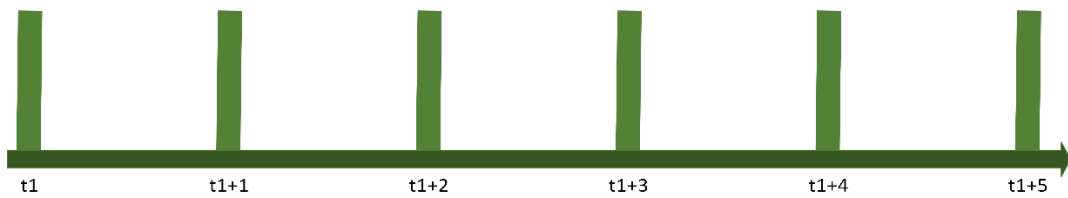


Figura 32 - Calendarização das comunicações em ordem a  $t_1+x$ , fonte: autor

A calendarização vai permitir à NCAP, com o perfil de recolha de energia, que será traçado a partir dos registos que os nós (TIM), lhe comunicam, juntamente com o custo associado a cada tarefa, estimar o tempo que cada nó necessita estar em modo adormecido, de forma a armazenar a energia necessária para realizar as tarefas que lhe estão destinadas. Além disto evitará que o nó acorde antes de ter recolhido energia suficiente, reduzindo assim o desperdício com ações desnecessárias. Partirá ainda da NCAP a responsabilidade de manter a sincronização, ou no caso de esta não ser necessária ou não se justificar devido à inatividade temporária do nó, tem de ser futuramente capaz de comunicar com este na devida janela temporal desviada.

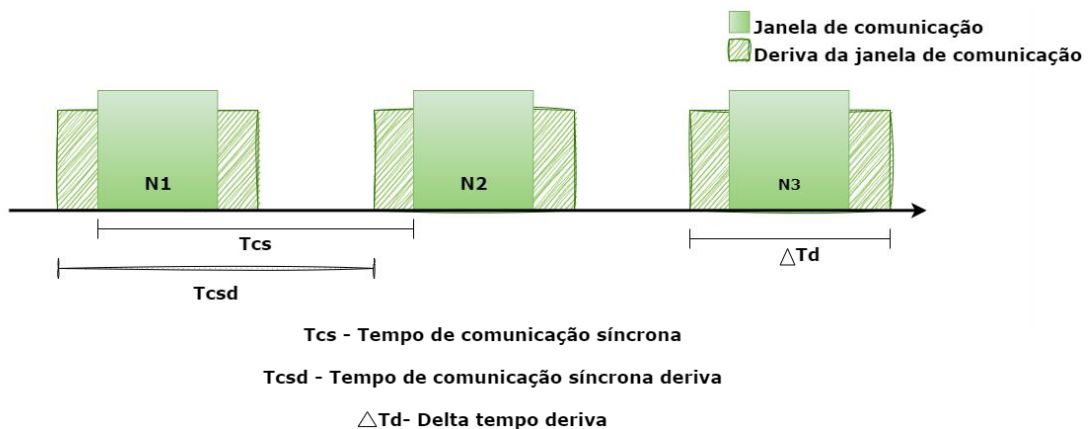


Figura 33 - Janelas e derivas temporais, fonte: autor

A deriva existe de forma a diminuir os erros de sincronização entre os *clock's* dos nós e da entidade gestora, com a diminuição desta deriva além de diminuirmos o gasto de energia, pela diminuição do tempo em que o nó está ligado em função total, aumentamos o tempo do modo *sleep*, logo, aumenta o tempo de recolha energética (Figura 34).

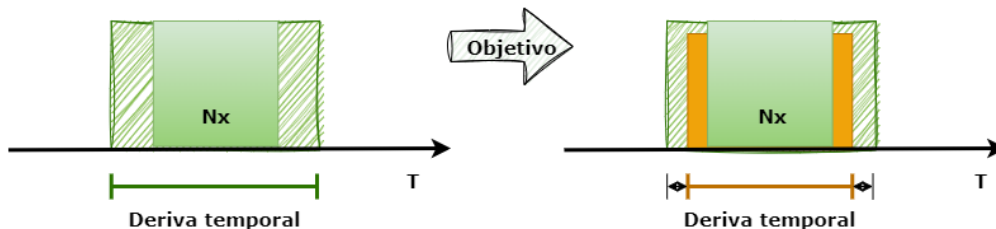


Figura 34 - Diminuição da deriva temporal, fonte: autor

Em suma, os NS enviam à NCAP, no início de cada comunicação, a potência disponível, esta devolve a tarefa que estes têm de realizar. Quando se dá o caso de o custo da tarefa ser superior à energia disponível na TIM, a NCAP indica-lhe um período de tempo no qual estará em modo *sleep*, a realizar RE, para posteriormente executar a tarefa pretendida, Figura 37. Em caso contrário, ou seja, quando a energia disponível é suficiente para concretizar a tarefa pedida, o NS realiza-a e devolve à NCAP informação sobre a energia que restou e o seu *clock* atual no fim da comunicação. Findo isto adormece o tempo que a NCAP lhe solicitar, como é possível observar na Figura 35.

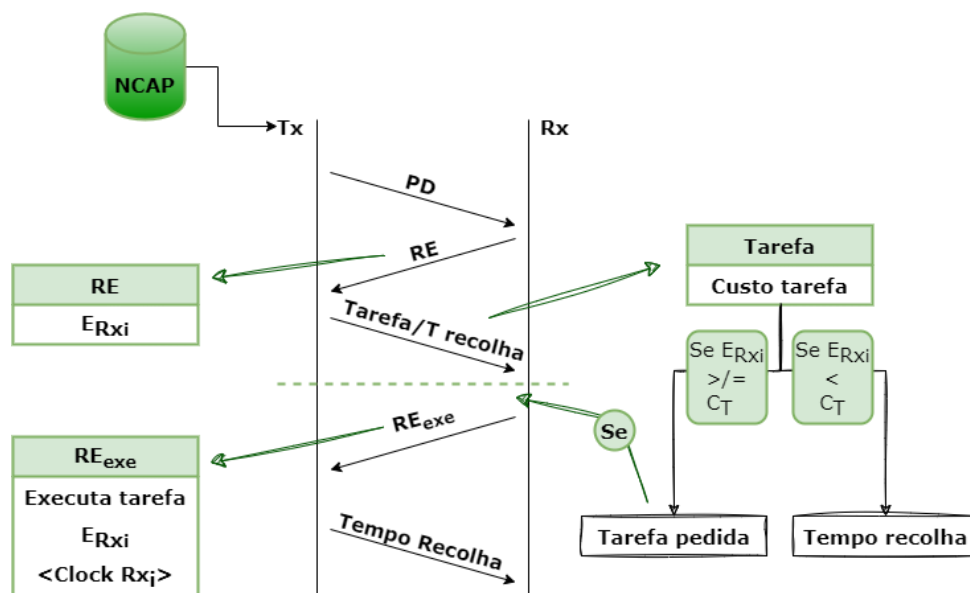


Figura 35 - Comunicações base entre a NCAP e TIM, fonte: autor

Considerando o sincronismo temporal como uma tarefa, aquando do pedido desta, no reconhecimento da execução/*acknowledge* (REexe), o TIM não devolve a informação do *clock* à NCAP, pois esta acabou de o sincronizar logo, tem conhecimento á cerca do mesmo.

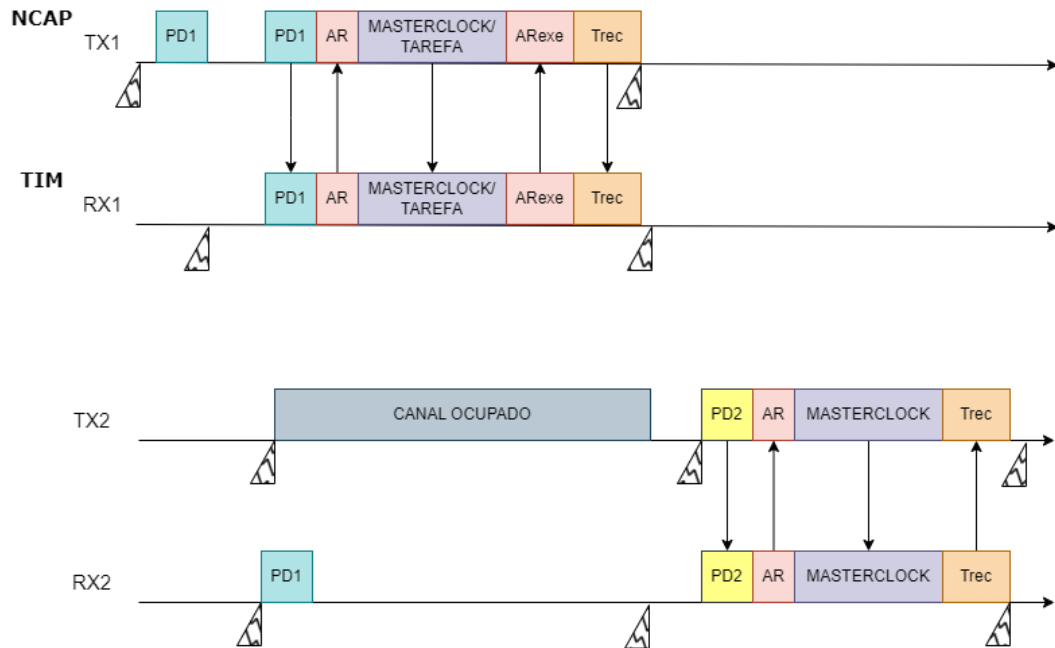


Figura 36 – Troca de comunicações entre NCAP e TIM, fonte: autor

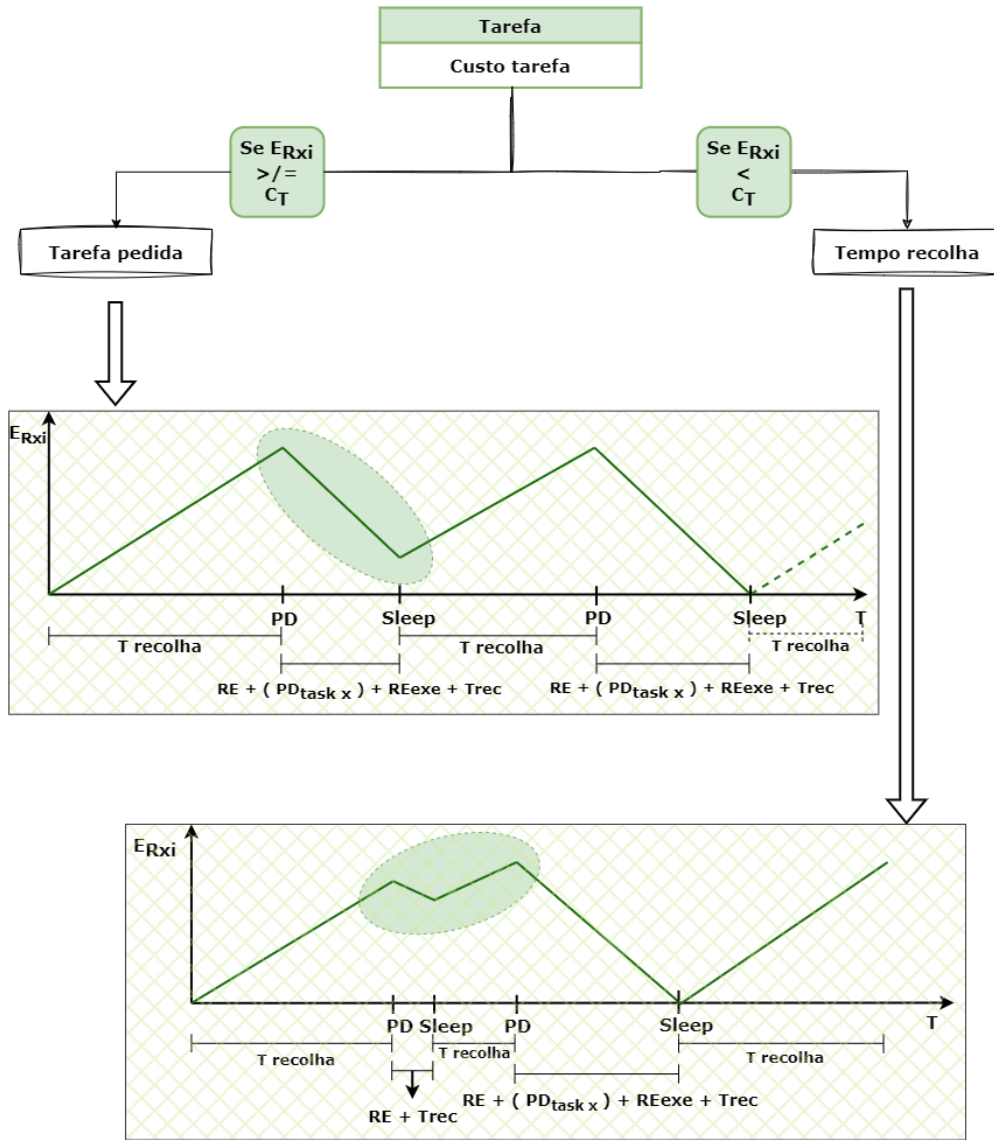


Figura 37 – Fluxograma de decisão de tarefas consoante energia disponível, fonte: autor

## 3.2. Descrição do modelo

### 3.2.1. Formulação teórica:

Esta formulação pode ser simplificada num diagrama de blocos, como ilustrado na Figura 38, que recebe três variáveis e origina uma saída, sendo que uma das entradas será uma “condição” definida, pois corresponderá a um valor pré estabelecido. Ou seja, recebe a informação de eficiência energética e de desvio temporal, das quais se alcança um histórico da energia obtida, em relação ao tempo que esteve a recolher a mesma. Seguidamente esta informação é analisada de acordo com a energia que será necessária para a realização de uma determinada tarefa, resultando assim, o intervalo de tempo que o sistema estará adormecido para recolha dessa mesma quantidade de energia, como referido anteriormente.

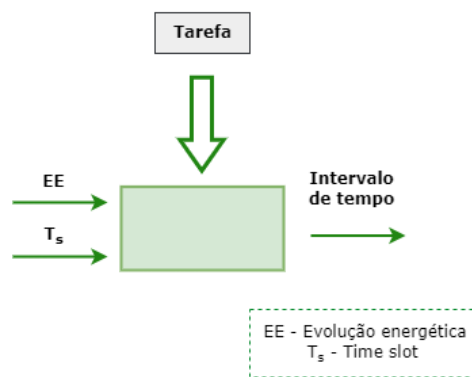


Figura 38 – Diagrama de blocos da formulação de base, fonte: autor

A evolução energética (EE) consiste num histórico da recolha de energia ao longo do tempo, permitindo traçar um gráfico e *time slot* ( $T_s$ ) define um intervalo de tempo correspondente ao tempo de avaliação da evolução energética. A tarefa, é a entrada que foi referida anteriormente como “condição”, pois esta é conhecida pelo sistema, que sabe as características energéticas que lhe estão associadas. No entanto sabe-se que pela imprevisibilidade dos fatores ambientais, ao nível da não linearidade da disponibilidade/intensidade, também a RE não é linear, podendo por isto existir erros de disponibilidade energética mesmo após a calendarização por parte da NCAP.

Resumindo, ao ser conhecida a quantidade de energia armazenada e o tempo que esteve a recolher a mesma, é possível calcular a potência disponível, obtendo assim informação da evolução energética individual. Com a potência e a energia necessária para realizar a próxima atividade/tarefa prioritária, pode ser calculado o tempo necessário que o nó tem de estar em repouso para obter essa quantidade energética (Figura 39).

No entanto, além desta calendarização do tempo do modo adormecido, tem de ser definido também um intervalo de tempo máximo para serem feitas sincronizações temporais e dos mapas energéticos.

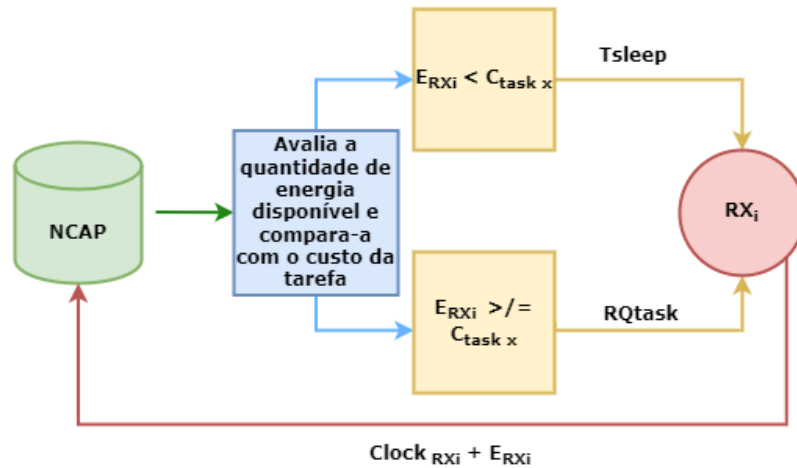


Figura 39 - Funcionamento das comunicações base entre a NCAP e as TIM, fonte: autor

A aplicação sabe qual o custo de cada atividade e vai atualizando a taxa de recolha de energia, assim calcula o tempo que o nó precisa de estar inativo para armazenar a potência mínima para realizar a tarefa e indica-lhe daqui a quanto tempo terá de acordar para estabelecer comunicação. No entanto, teremos sempre presente uma calendarização de sincronizações para tentar mitigar os erros associados aos desvios dos relógios.

Pretende-se então sincronizar os relógios da NCAP com os das TIM de forma a diminuir a deriva e dessa forma o tempo excedente que seria usado nas janelas de comunicação passa a ser usado para obtenção de energia. Por fim com a melhoria deste conjunto de parâmetros há ainda uma maximização do tempo disponível para as comunicações e para a realização de tarefas.

### 3.2.2. Formulação matemática:

Esta formulação tem como base a formulação matemática de [15], adaptada às variáveis escolhidas.

#### 3.2.2.1. Modelo de sincronização:

Custo total da tarefa de sincronização de acordo com o modelo de [15], tendo em conta a realização da tarefa no recetor ( $P_{R,COM}$ ) no transmissor ( $P_{T,COM}$ ) e o custo de estar no modo escuta, tendo em conta o desvio de relógio ( $P_{ESC\theta}$ ), sendo que todos estes representam o transdutor.

$$P_S = P_{R,COM} + P_{T,COM} + P_{ESC\theta} \quad (1)$$

Onde, para o custo de transmissão tem-se de relacionar a energia gasta nessa mesma transmissão ( $E_{T,COM}$ ) com o período entre comunicações ( $T_{COM}$ );

$$P_{T_{COM}} = \frac{E_{T_{COM}}}{T_{COM}} \quad (2)$$

Para o custo de recepção periódica de tarefas relacionamos a energia gasta para a recepção ( $E_{R_{COM}}$ ), o período ( $T_{COM}$ ) e o custo de estar em modo escuta ( $P_{ESC}$ ), com o período entre comunicações. Aqui também se adaptamos ao nosso problema substituindo uma constante da formulação original por uma constante  $k$  que neste caso varia entre 1 e 3; o parâmetro  $k$  representa o tamanho da deriva temporal na janela de comunicação sendo que é uma constante que multiplica o desvio associado ao *clock*,

$$P_{R_{COM}} = \frac{(E_{R_{COM}} + T_{COM} \cdot (k \cdot \theta) \cdot P_{ESC})}{T_{COM}} \quad (3)$$

E ainda, para o custo de estar em modo escuta por um tempo extra, causado pelo drift do *clock* ( $\theta$ ), tem-se a relação do período, o desvio e o custo de estar em modo escuta com o período de envio de dados ( $T_{DADOS}$ )

$$P_{ESC\theta} = \frac{T_{COM} \cdot k \cdot \theta \cdot P_{ESC}}{T_{DADOS}} \quad (4)$$

→ Vai haver sempre um custo mínimo associado ao  $P_{ESC}$  derivado do gasto energético que ocorre logo quando o NS sai do modo *sleep*.

Baseado na equação (1) o período ótimo de comunicação ( $T_{COM}$ ) pode ser encontrado usando a primeira derivada em relação ao ( $T_{COM}$ ) que deve ser igualado a zero de forma a encontrar o  $P_{Sinc}$ .

$$T_{COM} = \sqrt{\frac{T_{DADOS} \cdot (E_{T_{COM}} + E_{R_{COM}})}{(k \cdot \theta) \cdot P_{ESC}}} \quad (5)$$

Tendo como objetivo, neste caso, a otimização do desvio das janelas de comunicação, diminuindo o tempo da deriva das mesmas, tem-se as equações anteriores em ordem a ( $k \cdot \theta$ ) de forma a conseguirmos otimizá-los:

Numa comunicação, o custo de consumo de energia na recepção é determinado por:

$$P_{R_{COM}} = \frac{(E_{R_{COM}} + T_{COM} \cdot (k \cdot \theta) \cdot P_{ESC})}{T_{COM}} \Leftrightarrow$$

$$\Leftrightarrow P_{R_{COM}} * T_{COM} = E_{R_{COM}} + T_{COM} \cdot (k \cdot \theta) \cdot P_{ESC} \Leftrightarrow$$

$$\Leftrightarrow \frac{(P_{R_{COM}} \cdot T_{COM}) - E_{R_{COM}}}{T_{COM}} = (k \cdot \theta) \cdot P_{ESC} \Leftrightarrow$$

$$\Leftrightarrow k \cdot \theta = \frac{((P_{RCOM} \cdot T_{COM}) - E_{RCOM})}{(T_{COM} \cdot P_{ESC})} \quad (3.1)$$

O custo associado ao estado de escuta até receber um pedido de comunicação é dado por:

$$P_{ESC\theta} = \frac{T_{COM} \cdot (k \cdot \theta) \cdot P_{ESC}}{T_{DADOS}}$$

$$\Leftrightarrow P_{ESC\theta} \cdot T_{DADOS} = T_{COM} \cdot (k \cdot \theta) \cdot P_{ESC}$$

$$\Leftrightarrow \frac{P_{ESC\theta} \cdot T_{DADOS}}{T_{COM}} = (k \cdot \theta) \cdot P_{ESC}$$

$$\Leftrightarrow \frac{P_{ESC\theta} \cdot T_{DADOS}}{T_{COM} \cdot P_{ESC}} = (k \cdot \theta) \quad (4.1)$$

E o tempo entre as comunicações definido por:

$$T_{COM} = \sqrt{\frac{T_{DADOS} \cdot (E_{T\_COM} + E_{R\_COM})}{(k \cdot \theta) \cdot P_{ESC}}} \Leftrightarrow$$

$$\Leftrightarrow (T_{COM})^2 \cdot (k \cdot \theta) \cdot P_{ESC} = T_{DADOS} \cdot (E_{T\_COM} + E_{R\_COM})$$

$$\Leftrightarrow (k \cdot \theta) = \frac{T_{DADOS} \cdot (E_{T\_COM} + E_{R\_COM})}{(T_{COM})^2 \cdot P_{ESC}} \quad (5.1)$$

### 3.2.2.2. Modelo de custo energético:

O custo energético total, nesta formulação, divide-se em duas partes [15], sendo elas a energia despendida na receção ( $E_{TOTALRX}$ ) e na transmissão ( $E_{TOTALTX}$ ).

Tem-se então que a energia total necessária para realizar uma receção pode ser calculada por:

$$E_{TOTALRX} = f(t_{tb}, T_b, t_{esc}) =$$

$$= E_{TOTAL\_ON} + E_{RX\_JNICRAD} + E_{ESC} + E_{RX} + E_{RX\_FIMRAD} + E_{TOTAL\_SLEEP} \quad (6)$$

Ou, de forma mais compacta, a energia total para realizar uma transmissão é dada por:

$$E_{TOTALTX} = f(t_{tb}, T_b) =$$

$$= E_{TOTAL\_ON} + E_{TX\_INICRAD} + E_{TX} + E_{TX\_FIMRAD} + E_{TOTAL\_SLEEP} \quad (7)$$

Sendo  $t_{tb}$  o tamanho total em bits,  $T_b$  a taxa de bits em bits por segundo e  $t_{esc}$  o tempo em *idle listening*.

Sendo que quem recebe é quem transmite, ou seja, os sensores nó, logo o início e fim do rádio é o mesmo para um mesmo sensor:

$$E_{RX\_INICRAD} = E_{TX\_INICRAD} \text{ e } E_{RX\_FIMRAD} = E_{TX\_FIMRAD}$$

Para a energia total da comunicação tem-se a junção das anteriores:

$$\begin{aligned} E_{TOTAL\_RTX} &= f(t_{tb}, T_b, t_{esc}) = \\ &= E_{TOTAL\_ON} + E_{INICRAD} + E_{ESC} + E_{RX} + E_{TX} + E_{FIMRAD} + E_{TOTAL\_SLEEP} \end{aligned} \quad (8)$$

### 3.2.2.3. Modelo de *duty-cycling* síncrono:

Há necessidade de ter em conta a probabilidade de sucesso ( $Prob_{suc}$ ), pois a comunicação pode ser estabelecida fora da janela de comunicação e por isso não ser bem-sucedida. Assim tem-se ainda (BER - bit error rate) e a probabilidade de transmissão ( $Prob_{TX}$ ).

$$Prob_{suc} = Prob_{TX} * (1 - BER)^{t_{tb}} \quad (9)$$

Como foi referido anteriormente existe um tempo de deriva associado às comunicações onde os NS estão em modo *idle* a aguardar a comunicação da TIM, então:

$$t_{esc\_dados} = t_{ESCCOM} = \frac{(k \cdot \theta) \cdot T_{COM}}{Prob_{suc}} \quad (10)$$

Adaptando as equações 1, 2, 3 e 4 para *duty-cycling* síncrono é necessário considerar o custo total para *duty-cycling* que é determinado pela soma de várias parcelas de consumo de energia:

$$P_{duty\_cycle} = P_{RCOM} + P_{TCOM} + P_{SLOT\_ESC} \quad (11)$$

Em que o gasto de energia para transmissão é dado por:

$$P_{TCOM} = \frac{E_{TOTALTX}(t_{tbCOM}, T_b)}{T_{COM}} \quad (12)$$

O consumo de energia para receção por:

$$P_{R_{COM}} = \frac{E_{TOTALRX}(t_{tb_{COM}}, T_b, t_{esc_{dados}})}{T_{COM}} \quad (13)$$

E o consumo de energia por estar à escuta é determinado por:

$$P_{SLOT_{ESC}} = \frac{E_{TOTALRX}(0, T_b, t_{esc_{dados}} + t_{slot})}{T_{COM}} \quad (14)$$

### 3.3. Algoritmo selecionado

Tendo em conta a informação anterior sobre os diferentes tipos de algoritmos, para o caso em questão são abordados os algoritmos meta-heurísticos, pois parecem ser os mais adequados à otimização dos parâmetros referidos em 3.2.

O algoritmo meta-heurísticos dividem-se em dois grupos, os algoritmos baseados em indivíduos e algoritmos baseados em populações. Na primeira classe, apenas uma solução é gerada aleatoriamente e melhorada ao longo da otimização. Na segunda o algoritmo de otimização gera mais que uma solução aleatória (muitas) e melhora-as durante a otimização[16].

Assentam em duas técnicas usadas em combinação, numa certa proporção [4], [16]:

1. Intensificação (*exploitation/intensification*), (procura local) consiste na tentativa de construir soluções em regiões já exploradas, onde soluções de qualidade foram encontradas.
2. Diversificação (*exploration/diversification*), (procura global) consiste na tentativa de construir soluções pertencentes a regiões do espaço de procura inexploradas, que diferem significativamente das soluções já encontradas.

Portanto, o desempenho e a eficiência de uma meta-heurística dependem, profundamente, do equilíbrio entre as técnicas inerentes de diversificação e intensificação.

#### 3.3.1. Particle swarm optimization

Assim sendo e depois de serem observados vários algoritmos bio inspirados, o escolhido para este modelo foi o *particle swarm optimization* (PSO), ou seja, otimização por enxame de partículas. Inspirado no comportamento coletivo de animais sociais como os pássaros, peixes, formigas, abelhas entre outros, o conceito enxame (*swarm*), foi introduzido para investigar sistemas de inteligência distribuídos com vários agentes.

Um enxame é uma população de agentes homogêneos e simples que executam tarefas bastante elementares e interagem localmente entre si e o ambiente: o comportamento coletivo

surge como consequência da auto-organização e das interações locais (e parcialmente estocásticas). De facto, embora estes agentes (indivíduos do enxame) sejam relativamente pouco sofisticados, com capacidades limitadas por si só, a capacidade de atingir o objetivo final de resolução de problemas é fornecida pelos padrões comportamentais e de interação pela partilha de informações. Assim, algoritmos baseados em enxames surgiram como um conjunto de algoritmos baseados na natureza e baseados em populações.

Este algoritmo baseia-se no comportamento coletivo de um grupo de organismos, onde o grupo tenta atingir um objetivo comum, sendo que cada indivíduo tem em conta o feedback dos restantes membros. Portanto, um enxame é um grande número de indivíduos, que interagem entre si e com o ambiente em que se encontram, procurando obter uma solução global, o mais próximo possível de uma solução ótima. Este tipo de otimização, é frequentemente utilizada em problemas onde a função a otimizar é descontínua, não diferenciável e com vários parâmetros não lineares [14].

Isto traduz-se numa sequência de alguns passos iterativos, definidos com base no comportamento do organismo que está a simular. Cada membro/partícula tenta encontrar uma solução potencialmente ótima, comunica-a a todos e conseqüentemente, cada indivíduo terá uma solução ótima (solução ótima particular) e da análise de todas essas soluções ótimas surge a melhor solução geral (solução ótima global).

O processo de otimização começa com um conjunto de soluções criadas aleatoriamente. Além do conjunto de soluções, existe outro conjunto denominado de velocidade que é responsável por armazenar e definir a quantidade de movimento das partículas. Durante a otimização, a velocidade de uma partícula é atualizada com base na melhor solução obtida até ao momento, bem como na melhor solução encontrada pelo enxame. A interação social é direta, pois o movimento de cada partícula não é apenas influenciado pela própria memória (melhor solução encontrada até ao momento), mas também é direcionado para a melhor posição encontrada por outros agentes, ou seja, a melhor posição encontrada por todos, até ao momento [16].

Em suma, as partículas movem-se no espaço de busca, procuram soluções boas/ótimas, para um determinado problema, e usam a combinação das melhores soluções. O processo é repetido até que critério de cessação seja satisfeito [4]. Como as melhores soluções são guardadas no algoritmo PSO, há sempre, uma grande possibilidade de encontrar soluções melhores ao pesquisar em volta delas. Esta é a principal razão do sucesso deste algoritmo [16].

### **3.3.2. Formulação do PSO**

No PSO, cada solução candidata é chamada de “partícula” e representa um ponto num espaço D-dimensional, onde ‘D’ é o número de parâmetros a serem otimizados. Assim, a posição da  $i$ -ésima partícula pode ser descrita pelo vetor  $x_i$  [16]:

$$x_i = [x_{i1} \ x_{i2} \ x_{i3} \ \dots \ x_{iD}] \quad (15)$$

E a população de N soluções candidatas que constituem o enxame é [16]:

$$x_i = \{x_1, x_2, x_3, \dots, x_N\} \quad (16)$$

Na busca da solução ótima para o problema, as partículas assumem trajetórias (posições) no espaço de pesquisa baseadas na equação seguinte [16]:

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (17)$$

Onde t+1 representa duas iterações seguidas e  $v_i$  é o vetor que contém as componentes da velocidade da i-ésima partícula ao longo da dimensão D.

O vetor de velocidade define o movimento das partículas ao longo do espaço de procura e resulta da contribuição de três termos [16]:

- Inércia/momento que previne a mudança drástica da direção do movimento da partícula, mantendo a direção anterior;
- Componente cognitiva, explica a tendência que as partículas têm de voltarem á sua melhor posição encontrada anteriormente (C1);
- Componente social, propensão de uma partícula para se deslocar para a melhor posição global (C2).

As componentes, cognitiva e social, modulam a magnitude dos paços dados pelas partículas, na direção do melhor particular e do melhor global respetivamente [16].

Com base nisto, a velocidade é definida por [16]:

$$v_i(t + 1) = v_i(t) + c1 (p_i - x_i(t)).R_1 + c2 (g - x_i(t)).R_2 \quad (18)$$

Onde  $p_i$  é o melhor particular, e  $g$  é o melhor global. As constantes de aceleração  $c1$  e  $c2$  são valores reais por norma compreendidos entre 0 e 4, podem ser constantes, tipicamente,  $C1 = 2$  e  $C2 = 2$  com  $C1 + C2 \leq 4$  ou variáveis em função do número de iterações máximas permitidas (variante designada como *Particle swarm optimization with time varying acceleration coefficients*) [16].

$R_1$  e  $R_2$  são duas matrizes de números aleatórios entre 0 e 1, mantendo uma influência estocástica na regra de atualização de velocidade (4).

Após isto é feita a atualização das posições por meio de [16]:

$$x_i^{it+1} = v_i^{it} + x_i^{it} \quad (19)$$

Por fim de forma a evitar que as partículas viagem para fora do espaço de procura, ou seja, que se encontrem fora de algum dos limites (superior ou inferior), o movimento da partícula é modificado garantindo que o novo posicionamento se encontra dentro do espaço de procura [16].

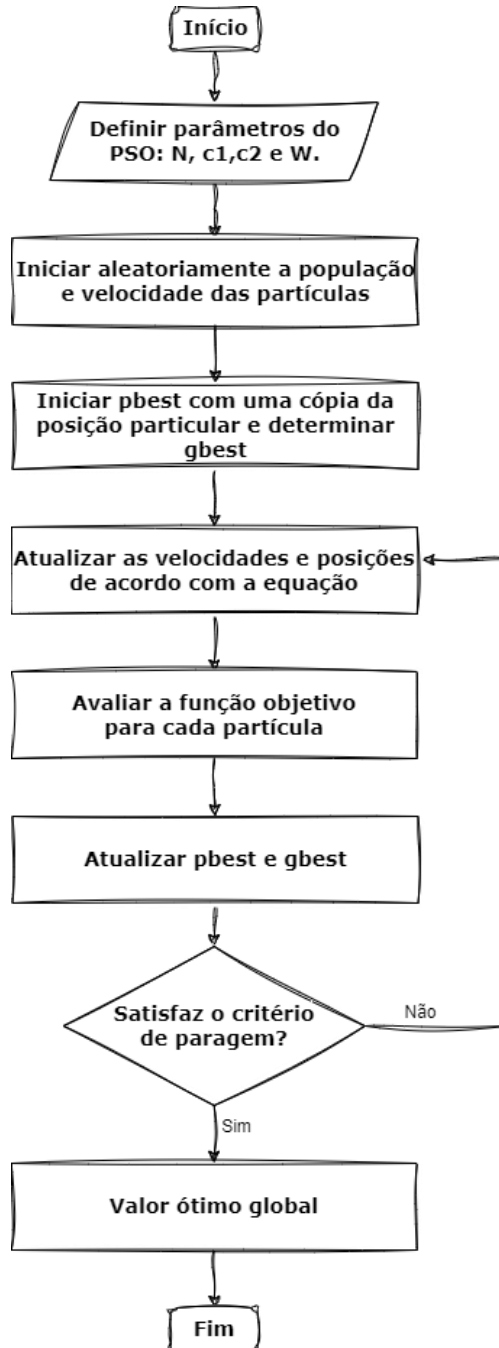


Figura 40 - Fluxograma do PSO, fonte: autor, inspirado em [16]

### 3.4. Modelo

#### 3.4.1. Conceitos base a ter em conta:

O ciclo de relógio é o intervalo de tempo entre o início da subida/descida de um pulso até ao início da subida/descida do pulso seguinte, o período é o tempo decorrido na realização de um ciclo e a frequência é a quantidade de ciclos por segundo. O número de ciclos de relógio vai corresponder ao número de pulsos que ocorrem por cada segundo.

A frequência em hertz é igual ao número de ciclos por segundo, logo, com conhecimento da frequência obtém-se o número de ciclos de relógio realizados por segundo. Para um determinado tempo, com o período, obtido através do inverso da frequência, obtém-se o número de ciclos realizados.

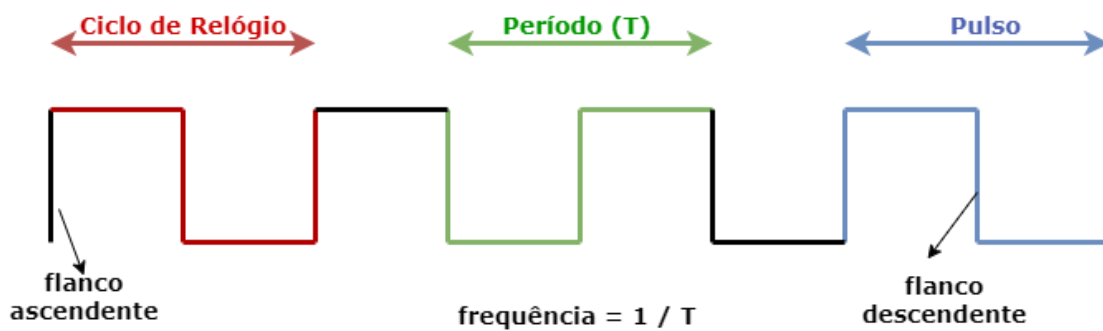


Figura 41 - Ilustração dos constituintes de uma função, fonte: autor

Por exemplo, se a frequência for de 1 MHz, ou seja, 1.000.000 Hz, são 1.000.000 ciclos por segundo e um período igual  $1/1.000.000 = 0,000001$  segundos. Em 600 segundos há  $600/0.000001 = 600.000.000$  ciclos de relógio.

#### 3.4.2. Recolha energética

Esta simulação está a ser realizada tendo em vista a sua aplicação a dispositivos que sejam alimentados por sistemas de recolha de energia. Portanto, como foi referido anteriormente, a NCAP será informada, por parte das TIM, sobre a quantidade de energia que dispõe a cada início de comunicação, e ainda quando a tarefa não for apenas de sincronismo; também informará no fim.

Com a junção dos dados sobre o tempo que a TIM esteve em modo adormecido (*sleep*) a efetuar recolha de energia, e os dados sobre a quantidade de energia disponível no início de cada comunicação, a NCAP consegue traçar um perfil de recolha de energia. Sabe-se que estes sistemas de recolha de energia do meio ambiente, seja qual for a fonte, dependem da disponibilidade de

energia existente bem como da “taxa” de disponibilidade, portanto é também sabido que essa disponibilidade não é linear nem constante.

A título de exemplo e para melhor compreensão tem-se, com base em [7], e supondo que o sistema de recolha escolhido fossem painéis fotovoltaicos, os valores de potência (P<sub>cc</sub>) obtidos são calculados com base em:

$$P_{cc} = \frac{E}{\frac{G_{poa}}{R}} \quad (20)$$

Sendo que ‘P<sub>cc</sub>’ é potência média necessária em corrente contínua (kW<sub>pcc</sub>), ‘G<sub>poa</sub>’ é o ganho dado pela radiação solar: médio mensal do total diário (kWh/m<sup>2</sup>/dia), ‘E’ é consumo médio diário durante o ano (kWh/dia) e ‘R’ é rendimento ou eficiência do sistema inversor e conexões, com valores aproximadamente entre 0,7 e 0,9, sendo que o autor adotou o valor 0,8 neste estudo. Do trabalho experimental realizado em [7] resultou a tabela seguinte, e de forma a estimar o valor da energia elétrica resultante do sistema fotovoltaico selecionado, foi usada a equação 25.

$$E = \frac{A * E_{ff} * G_{poa} * R}{100} \quad (21)$$

- A é a área total do sistema (m<sup>2</sup>)
- E<sub>ff</sub> é a eficiência de conversão

Tabela 9 - Tabela de irradiação, retirada de [7]

Ângulo β (graus)	0°	10°	15°	20°	25°	33°	90°
<b>Irradiação (kWh/m<sup>2</sup>/dia)</b>							
<b>Para α = -12°</b>	5.35	5.36	5.42	5.62	5.45	5.39	3.30
<b>Irradiação (kWh/m<sup>2</sup>/dia)</b>							
<b>Para α = 0°</b>	5.35	5.36	5.42	5.66	5.46	5.41	3.30

Neste caso como a aplicação deste trabalho seria em pequena escala vamos assumir um painel quadrado com 5 centímetros de lateral, logo seria um painel de 0.05 m de largura, portanto a sua área seria de 0.025 metros quadrados Figura 42. Usando então o valor de 0.8 para ‘R’, para E<sub>ff</sub> o valor de 2.27 e para G<sub>poa</sub>, tendo em consideração a cidade de Santa Maria da Feira que possui um valor de radiação solar de entre 145 e 150 Kcal/cm<sup>2</sup> segundo [7], selecionando o valor mais baixo de 145 obtêm-se o valor de 0,168 635 kW/m<sup>2</sup>/dia. Então:

$$E = \frac{0.025 \cdot 2.27 \cdot 0.168635 \cdot 0.8}{100} = 7.6560290000000001e - 05 \frac{kWh}{dia}$$

$$\Leftrightarrow 7.6560290000000001e - 05 \frac{kWh}{24} = 3.190012083e - 06 kWh$$

sendo que  $1 kWh = 3,6 \times 10^6 J$ , temos

$$3.190012083e - 06 kWh = 11.484 Joules \tag{21.1}$$

Portanto se for estipulado que uma tarefa precisa de 2J o tempo de *sleep*, tomando a radiação como constante e linear seria de, no mínimo  $2 / 11.484 = 0.174s$ .

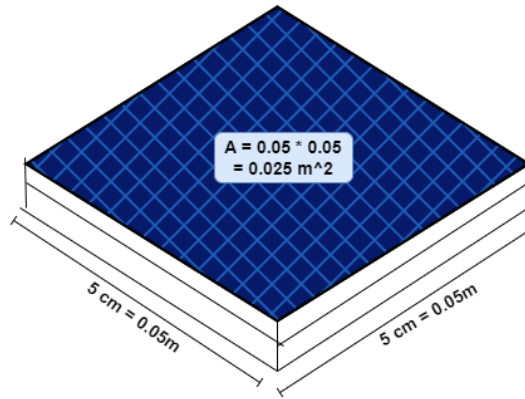


Figura 42 - Exemplo do painel fotovoltaico com medidas, fonte: autor

### 3.4.3. Adaptação da formulação:

A diminuição de ' $k \cdot \theta$ ' tem implicação na janela de desvio temporal, associada ao custo de escuta, pois é o fator que o multiplica, como é possível observar nas equações em 3.2.2. Da análise das formulações anteriores é possível observar que otimizando o valor de ' $k$ ' é possível diminuir o custo total da recepção na eq.(4.1), que conseqüentemente diminui o custo energético total da recepção eq.(6).

Função de custo usada como função objetivo:

$$F_{obj} = \frac{P_{ESC\theta} \cdot T_{DADOS}}{T_{COM} \cdot P_{ESC}} = (k \cdot \theta) \tag{4.1}$$

Para analisar essa otimização, pode-se adaptar a formulação de forma a manter apenas as variáveis que estão dependentes dos tempos de comunicação e da janela de deriva. Ou seja, vamos ignorar os parâmetros que são constantes em ambas as equações eq.(6) e (7). Assim, tendo em conta apenas o tempo útil de comunicação, obtém-se o seguinte. Além disto, de forma a facilitar as simulações é considerado que a probabilidade de sucesso é 1, ou seja, que a comunicação acontece sempre sem falhas.

Para a energia total da comunicação tem-se a junção das equações (6) e (7):

$$\begin{aligned}
 E_{TOTAL_{RTX}} &= f(t_{tb}, T_b, t_{esc}) \\
 &= E_{TOTAL_{ON}} + E_{INIC_{RAD}} + E_{ESC} + E_{RX} + E_{TX} + E_{FIM_{RAD}} + E_{TOTAL_{SLEEP}}
 \end{aligned} \tag{22}$$

Sendo que quem recebe é quem transmite, ou seja, os sensores nó, logo o início e fim do rádio é o mesmo para um mesmo sensor:

$$E_{RX_{INIC_{RAD}}} = E_{TX_{INIC_{RAD}}} \text{ e } E_{RX_{FIM_{RAD}}} = E_{TX_{FIM_{RAD}}}$$

E ainda a  $E_{TOTAL_{ON}}$  terá sempre o mesmo valor pois não depende dos tempos envolvidos nas comunicações, tem-se:

$$\begin{aligned}
 E_{TOTAL_{otimizada}} &= f(t_{tb}, T_b, t_{esc}) \\
 &= E_{ESC} + E_{RX} + E_{TX} + E_{TOTAL_{SLEEP}}
 \end{aligned} \tag{23}$$

Daqui sabe-se que  $E_{ESC}$  depende de ' $k \cdot \theta$ ' e do tempo que estará à escuta,  $E_{RX}$  terá um valor estabelecido pois, a troca de mensagens vai ser predefinida mais à frente, Figura 45, e neste caso será constante. A  $E_{TX}$  estará dependente do tamanho da mensagem a enviar, ou seja, da quantidade de bytes bem como o tempo associado a esse envio e por fim,  $E_{TOTAL_{SLEEP}}$  está dependente do tempo que a NCAP mandará a TIM para o estado *sleep* entre comunicações.

Dito isto e observando as equações apresentadas anteriormente, vai ser utilizada como função objetivo o custo de escuta, em ordem de ' $k \cdot \theta$ ' relativamente à otimização, no algoritmo proposto no ponto a seguir e que será explicado de forma mais aprofundada mais á frente no documento.

### 3.4.4. Adaptação do algoritmo:

Da análise da formulação do algoritmo e tendo em conta os parâmetros que são importantes considerar o algoritmo foi adaptado em alguns aspetos, passando assim a formular-se da forma apresentada a seguir, Figura 43.

- Número de partículas será representado pelo número de TIM;
- As posições no espaço serão as janelas de deriva temporal multiplicadas por ' $k$ ';
- Não é usada a função da velocidade pois pretende-se que a procura seja restrita às posições iniciais apenas variando o  $k$  pelo qual são multiplicadas;
- Atualização de posições com recurso à velocidade será dispensável pois advém da soma da velocidade a esta;
- Por fim a verificação dos limites ocorre de igual forma ao original.

De forma a simplificar o problema foi assumido que a recolha de energia é linear e que a quantidade de energia disponível é sempre, no mínimo igual, à quantidade de energia necessária para realizar as tarefas. Além disto também foi considerado que as comunicações são sempre bem sucedidas, ou seja, que não existem falhas de comunicação.

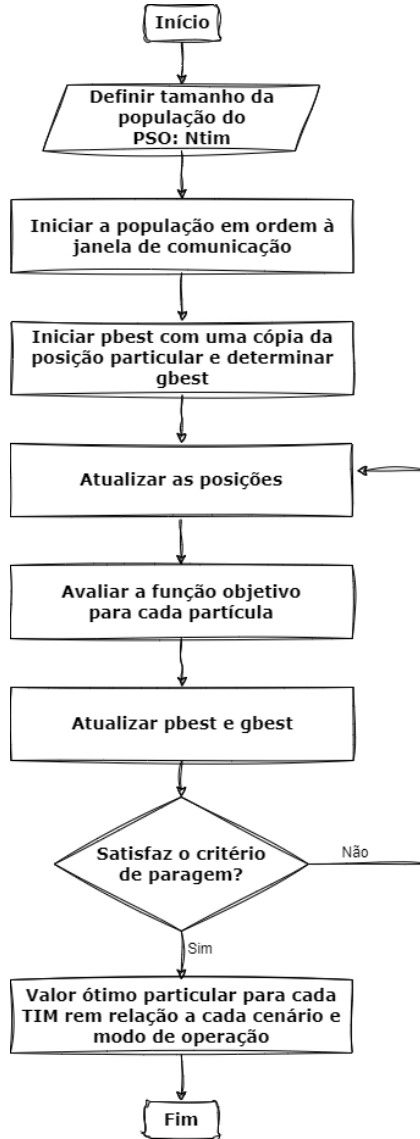


Figura 43 - Fluxograma do algoritmo proposto, fonte: autor

### 3.4.5. Definição dos parâmetros:

Inicialmente é necessário definir o número de TIM, bem como as frequências destas e da NCAP. É encontrado aleatoriamente um valor de desvio compreendido entre 10 e 50 ppm, sendo este multiplicado por um coeficiente 'k' que varia entre 3 e 1 em decrementos de 1 valor. Depois são definidos os tempos das tarefas e calculados os períodos nominais da TIM e NCAP bem como os períodos das TIM desviadas.

Então, usando o tempo das tarefas e os períodos calculados, obtém-se o número de ciclos de relógio que cada elemento realiza nesse tempo, comparam-se entre eles e daí resulta o tempo de discrepância, para a mesma tarefa, entre a NCAP e as TIM (nominal e desviadas). Tudo isto deve ser tido em conta para cada TIM individualmente, envolvendo os erros particulares de cada uma destas.

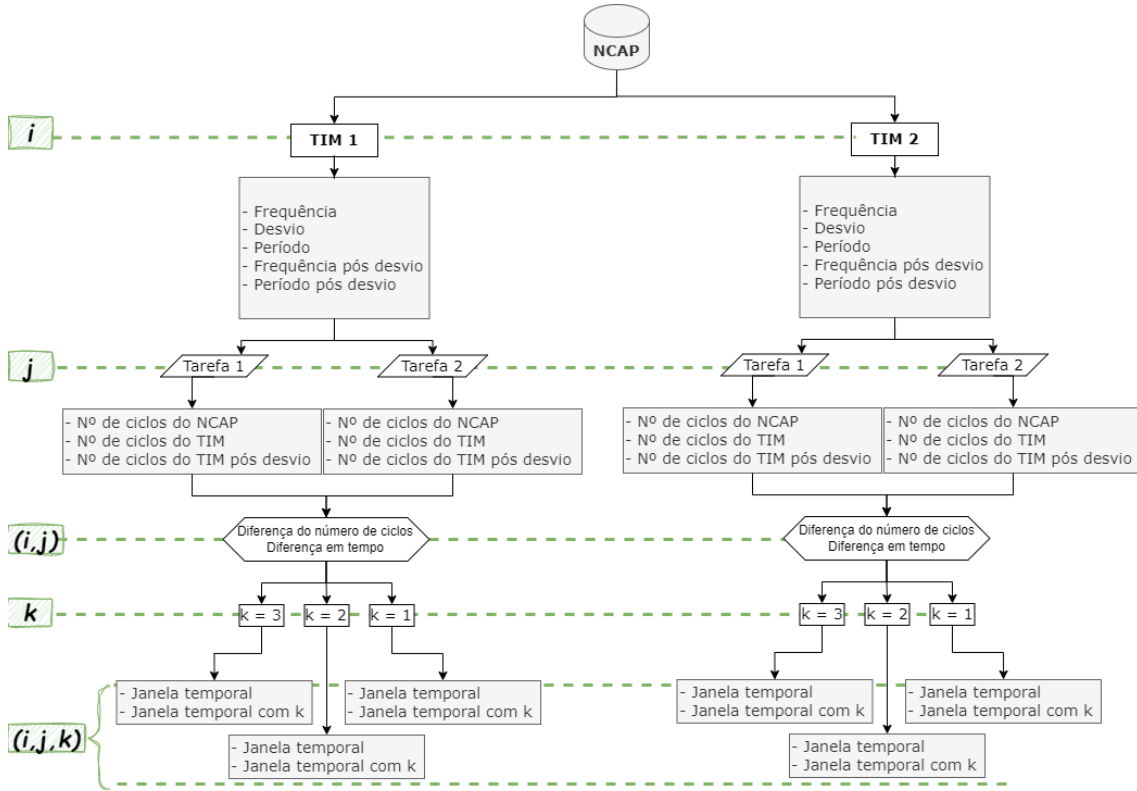


Figura 44 - Fluxograma da primeira parte do código, fonte: autor

Após calcular a diferença do número de ciclos, são calculadas as diferenças temporais em conjunto com o período correspondente, e daí obtém-se (dobrando o valor) as janelas de deriva correspondentes ao desvio de cada. O valor é dobrado pois a cada comunicação corresponde um desvio positivo e um negativo, ou seja, para cada lado da janela de comunicação que é aberta. Após isto e multiplicando pelo 'k' tem-se o valor da janela de deriva a otimizar. Por fim são definidos os parâmetros relativos ao algoritmo, necessários para o seu funcionamento. Tudo isto está representado na Figura 44.

De forma a simplificar o problema foi assumido que a recolha de energia é linear e que a quantidade de energia disponível é sempre, no mínimo igual, à quantidade de energia necessária para realizar as tarefas. Além disto também foi considerado que as comunicações são sempre bem sucedidas, ou seja, que não existem falhas de comunicação.

### 3.5. Caso em estudo:

Para este estudo foi escolhido o rádio CC2500 que opera a 2.4 GHz, com um ritmo de comunicação de 250 KBaud, com uma tolerância do relógio de +/- 40 ppm e um atraso de arranque de 0.015 segundos. Este rádio funciona com uma tensão de alimentação de 3.3 V, taxas de latência de 9 *bits* em RX e 8 *bits* em TX e por fim possui diferentes consumos de corrente conforme os modos de operação ativos [17].

Dos vários modos de operação disponíveis, foram seleccionados dois para serem testados, um com o menor gasto energético e outro com o maior gasto, sendo que esta seleção foi feita a partir da informação disponibilizada no *datasheet* do dispositivo referido [17]. Para uma melhor e mais sucinta compreensão dos modos escolhidos foi elaborada a Tabela 11, apresentada no ponto 3.6.

Foram também seleccionados dois cenários de quantidade de dados a enviar, sendo um o melhor que vai ser identificado como cenário A e o pior como cenário B, sendo respetivamente 3 *bytes* e 99 *bytes*. Com base em [18] foi estabelecida uma troca de mensagens padronizada, mostrada na Figura 45.

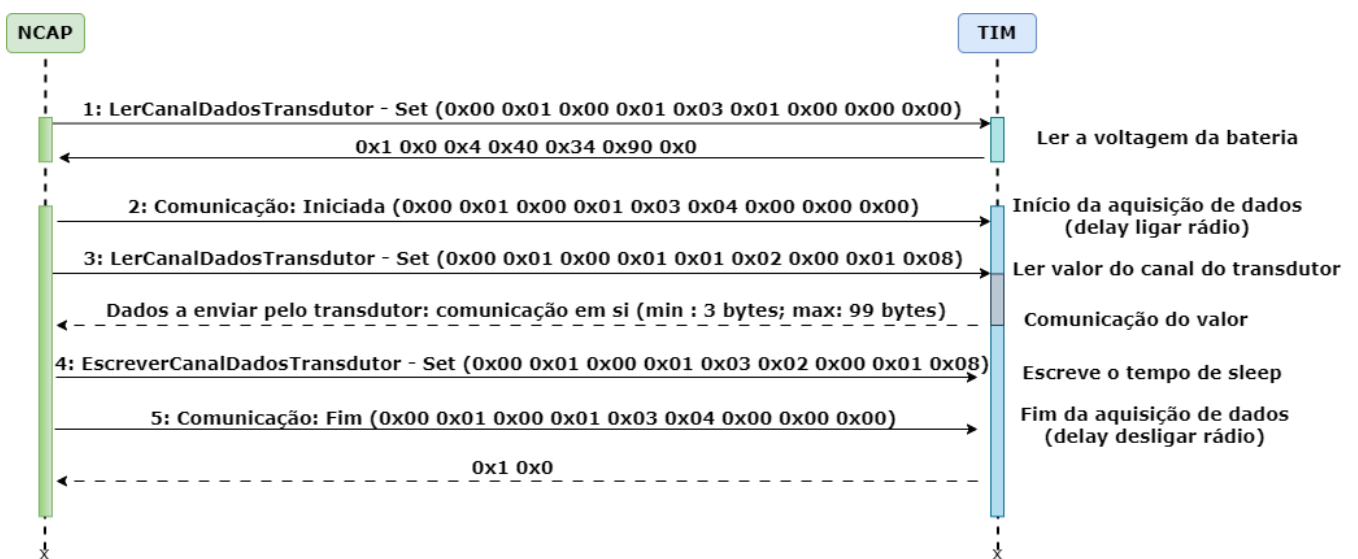


Figura 45 - Troca de mensagens numa comunicação, adaptado de [18]

É sabido também que este dispositivo tem uma padronização do tamanho de tramas de comunicação de 9 bytes para comunicações NCAP para TIM e no sentido contrário varia entre 3 e 99 bytes. O ritmo de comunicação ( $v_c$ ) é igual á divisão do número de bits ( $n_{bits}$ ) a enviar pelo tempo que este envio dura ( $\Delta T$ ).

$$v_c = \frac{n_{bits}}{\Delta T} \quad (24)$$

Deduz-se assim que o tempo necessário de envio é:

$$\Delta T = \frac{n_{bits}}{v_c} \quad (25)$$

Tem-se que, 1 byte = 8 bits, 1 kbit = 1024 bits, 1 kBaud = 1000 bits/s e 250 kBaud = 250000 bits/s. A partir destas conversões é possível calcular o tempo necessário para enviar um determinado número de bytes.

Seguindo a lógica de comunicações da figura anterior e Figura 45 considerando a quantidade de bytes a enviar para os cenários A e B, calcula-se o tempo necessário para cada. Sabe-se que o tempo do modo de transmissão é o mesmo para ambos os casos, pois as mensagens da NCAP para a TIM seguem o padrão referido, o que varia é o tempo de transmissão da informação por parte do TIM para a NCAP.

Tabela 10 – Cálculos para vários cenários/quantidades de bytes

	<b>Soma das tramas da comunicação (bytes)</b>	<b>Total de bytes</b>	<b>Conversão</b>	<b>Cálculo temporal</b>
<b>Cálculo geral</b>	-----	n bytes	$8 * n = x \text{ bits}$	$250000 \text{ bits} - 1 \text{ s}$ $8n \text{ bits} \text{ ---- } x_g \text{ s}$ $\Leftrightarrow x_g = \frac{8n}{250000} \text{ s}$
<b>Cenário A e B - RX</b>	$RX = 9 + 9 + 9 + 9$ $+ 9$ (NCAP ->TIM)	45 bytes	$8 * 45 = 360 \text{ bits}$	$x_{RX} = \frac{360}{250000}$ $\Leftrightarrow x_{RX} = 0.00144$ $= 1.4E - 3 \text{ s}$
<b>Cenário A - TX</b>	$TX = 7 + 3 + 2$ (TIM -> NCAP)	12 bytes	$8 * 12 = 96 \text{ bits}$	$x_{mTX} = \frac{96}{250000}$ $\Leftrightarrow x_{mTX} = 3.84E - 4$ $= 3.8E - 4 \text{ s}$
<b>Cenário B - TX</b>	$TX = 7 + 99 + 2$ (TIM -> NCAP)	108 bytes	$8 * 108 = 864 \text{ bits}$	$x_{pTX} = \frac{864}{250000}$ $\Leftrightarrow x_{pTX} = 0.00346$ $= 3.5E - 3 \text{ s}$

### 3.5.1. CC2500 – Modos de operação selecionados:

Como referido anteriormente foram usados os parâmetros presentes no *datasheet* do rádio CC2500, como este possui vários modos de operação e consequentemente diferentes gastos energéticos, foram selecionados os que foram entendidos como um melhor e um pior cenário dentro dos valores fornecidos pelo fabricante. A Tabela 11 permite obter uma visão geral dos cenários selecionados, com valores obtidos das tabelas 1 até 7 do *datasheet* do rádio CC2500 [17].

Tabela 11 - Cenários de gasto energético selecionados, fonte[17]

Modo	Parâmetro	Mín.	Typ.	Máx.	Unid.	Condição/Nota
<b>FONTE</b>	Fonte de alimentação	-0.3	.	3.9	V	Todos os pinos devem ter a mesma tensão
<b>RT</b>	<i>Data rate</i>	1.2	.	250	kBaud	GFSK and OOK
<b>SLEEP</b>	Consumo de corrente no modo <i>power down</i> (min)	.	92	.	µA	Regulador de corrente da parte digital <i>off</i> , regista valores retidos, XOSC ligado (estado <i>SLEEP</i> com MCSMo.OSC_FORCE_ON ligado)
	Consumo de corrente no modo <i>power down</i> (max)	.	900	.	nA	Regulador de corrente da parte digital <i>off</i> , regista valores retidos, oscilador ligado (estado <i>SLEEP</i> com WOR ligado)
<b>IDLE</b>	Consumo de corrente	.	1.5	.	mA	Apenas regulador de tensão e oscilador ligados
<b>RECEBER (RX)</b>	Consumo de corrente, RX <i>State</i> (min)	.	13.3	.	mA	Modo de receção, 250kBaud corrente otimizada, <i>input</i> bem acima do limite, MDMCFG2.DEM_DCFILT_OFF=1
	Consumo de corrente, RX <i>State</i> (max)	.	18.8	.	mA	Modo de receção, 250kBaud, <i>input</i> no limite, MDMCFG2.DEM_DCFILT_OFF=0

<b>TRANSMITIR (TX)</b>	Consumo de corrente, TX State (min)	.	11.1	.	mA	Modo de transmissão, -12 dBm potência de saída
	Consumo de corrente, TX State (max)	.	21.5	.	mA	Modo de transmissão, +1 dBm potência de saída
<b>LATÊNCIAS</b>	Receção	.	9	.	bit	Operação em série. Tempo desde que inicia a receção até aos dados estarem disponíveis no pin de output de receção.
	Transmissão	.	8	.	bit	Operação em série. Tempo desde que os dados entram até que os resultados são observados nas portas de saída.
<b>ATRASSO</b>	Atraso no arranque	.	150	.	µs	Medido no CC2500EM usando um cristal AT.41CD2 de NDK. Este parâmetro é amplamente dependente do cristal usado.

\*continuação da Tabela 11.

### 3.5.2. Cenários considerados

Atendendo à informação fornecida anteriormente, e considerando o cenário A e B, é possível auferir que de modo geral o cenário 'A', relativamente aos custos energéticos associados a este, será mais favorável que o cenário 'B', pois a quantidade de dados a enviar é menor. Sendo que foram selecionadas as duas configurações ilustradas na tabela anterior, ao nível do modo de operação do cc2500, uma melhor e outra pior, conjugando estas com os dois cenários referidos (A e B), surgem dois cenários gerais, um melhor e um pior de entre todos.

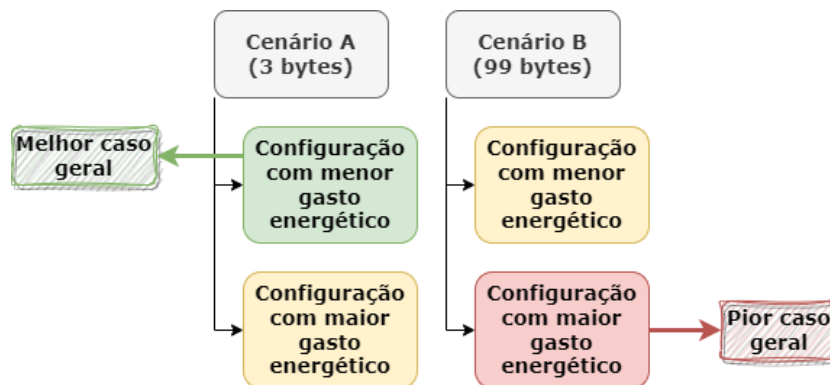


Figura 46 - Cenários de estudo, fonte: autor

Desta informação é possível auferir que o melhor cenário corresponde ao uso do modo de operação com menor gasto energético, para o envio de 3 bytes e o pior é referente ao modo de operação com maior gasto energético usado para o envio dos 99 bytes. Surgem ainda dois outros cenários que são o envio de 3 bytes com a configuração rádio de maior gasto energético e o envio de 99 bytes com a configuração rádio de menor gasto, estando tudo isto compilado na Figura 46.

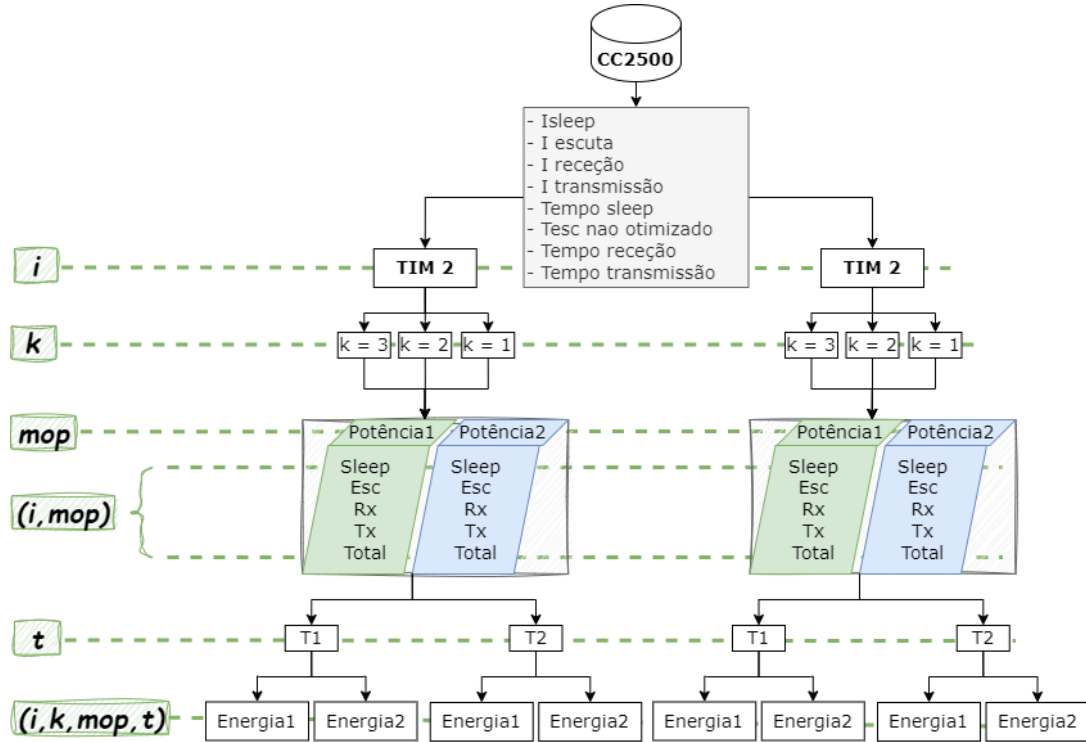


Figura 47 - Fluxograma do algoritmo, fonte: autor

**Folha em branco**

## Capítulo 4 - Resultados e Validação do Modelo

### 4.1. Resultados e Análise

#### 4.1.1. Análise prévia

Da primeira parte do código surge o resultado relativo à janela de comunicação que seria usada sem otimização do tempo de escuta. Esta é calculada tendo em conta os erros associados ao *drift* dos relógios dos TIM relativamente à NCAP, portanto resulta a janela de comunicação que seria necessária para colmatar o erro produzido por estes fatores. Para o cálculo são também usadas dois tempos de tarefas 600 e 900 segundos.

Esta janela de comunicação resultante tem de ter o seu valor dobrado para representar a compensação positiva e negativa da deriva, como já foi referido e observado na Figura 34, a deriva tem uma componente positiva e outra negativa de forma a alargar a janela em ambos os sentidos. Após isto ocorre a multiplicação pelos valores de 'k' e assim obter a janela de deriva multiplicada pelo fator de forma a poder ser otimizada no algoritmo.

O passo seguinte integra todo o algoritmo descrito e a formação dos gráficos associados às variáveis em estudo. Na parte inicial do algoritmo são calculadas a potência de modo *sleep*, receção, transmissão e escuta para cada corrente correspondente aos dois modos de operação selecionados, bem como a energia, decorrente da multiplicação da potência (eq.(23)) pelo tempo correspondente, sendo usado como tempo de escuta o valor correspondente da janela temporal anterior. É calculada a potência e energia total nos mesmos moldes obtendo assim informação pré- otimização, para mais tarde ser comparada.

$$E = P * T \quad (26)$$

Depois de aplicado o algoritmo, como referido nos pontos anteriores, obtem-se o valor de tempo de escuta otimizado o que se traduzirá na otimização da energia gasta em modo de escuta e consequentemente o gasto total de energia de comunicação.

#### 4.1.2. Resultados

Da execução do código obtêm-se vários gráficos que tornam possível comparar as energia antes e depois da otimização, esses gráficos são apresentados a seguir, neste caso as simulações foram feitas para dois TIM. O modo de operação '1' representa o modo de operação do rádio com maior gasto energético e o '2' representa o modo com menor gasto energético e ainda o cenário 'A' corresponde a 3 bytes e o 'B' a 99 bytes.

4.1.2.1. Valores comuns a todas as simulações:

- **Potência Total**

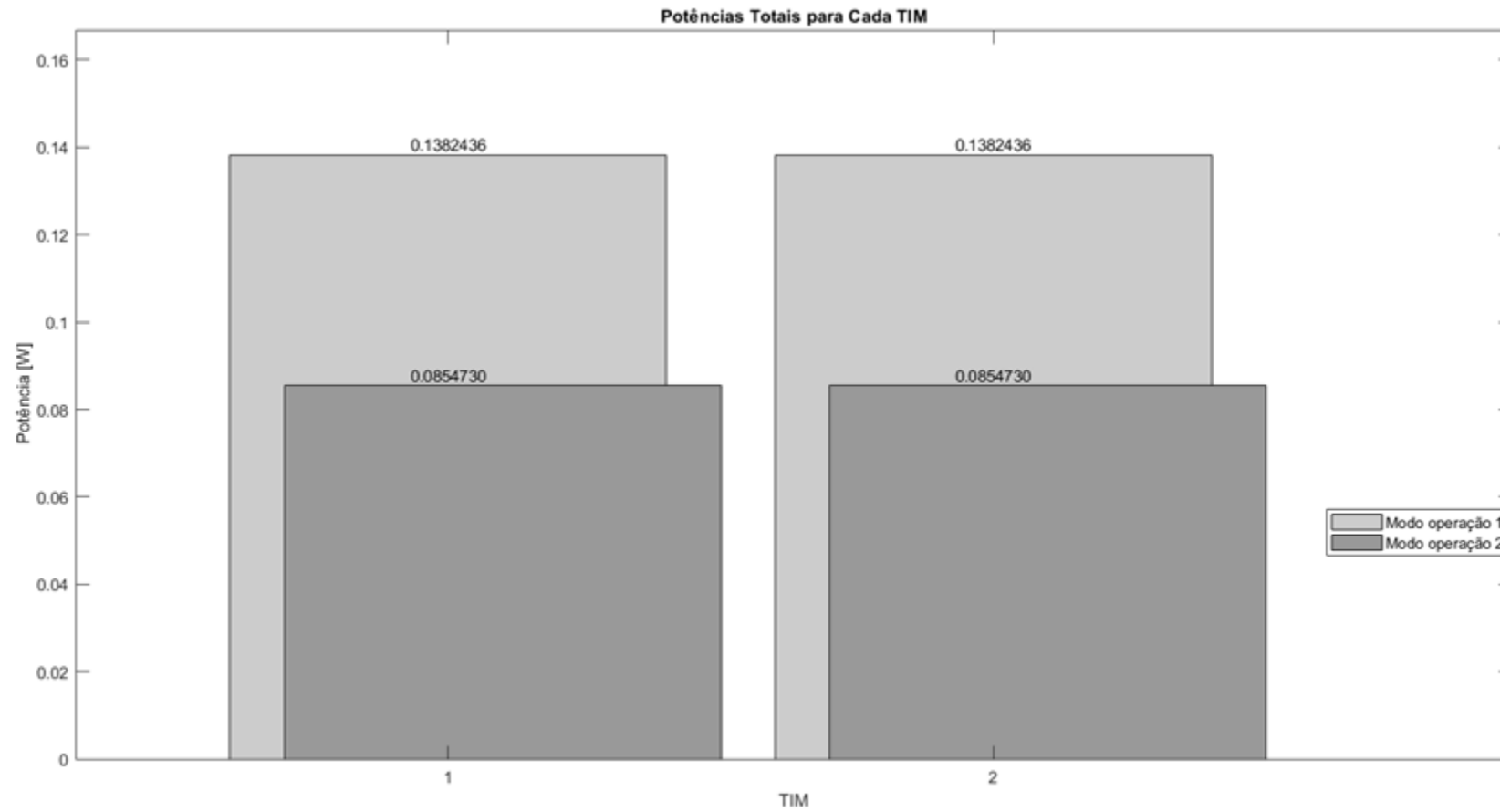


Figura 48 – Resultados do Cálculo da Potência total para cada modo de operação

- Componentes constituintes da potência total (igual para todos os TIM):**

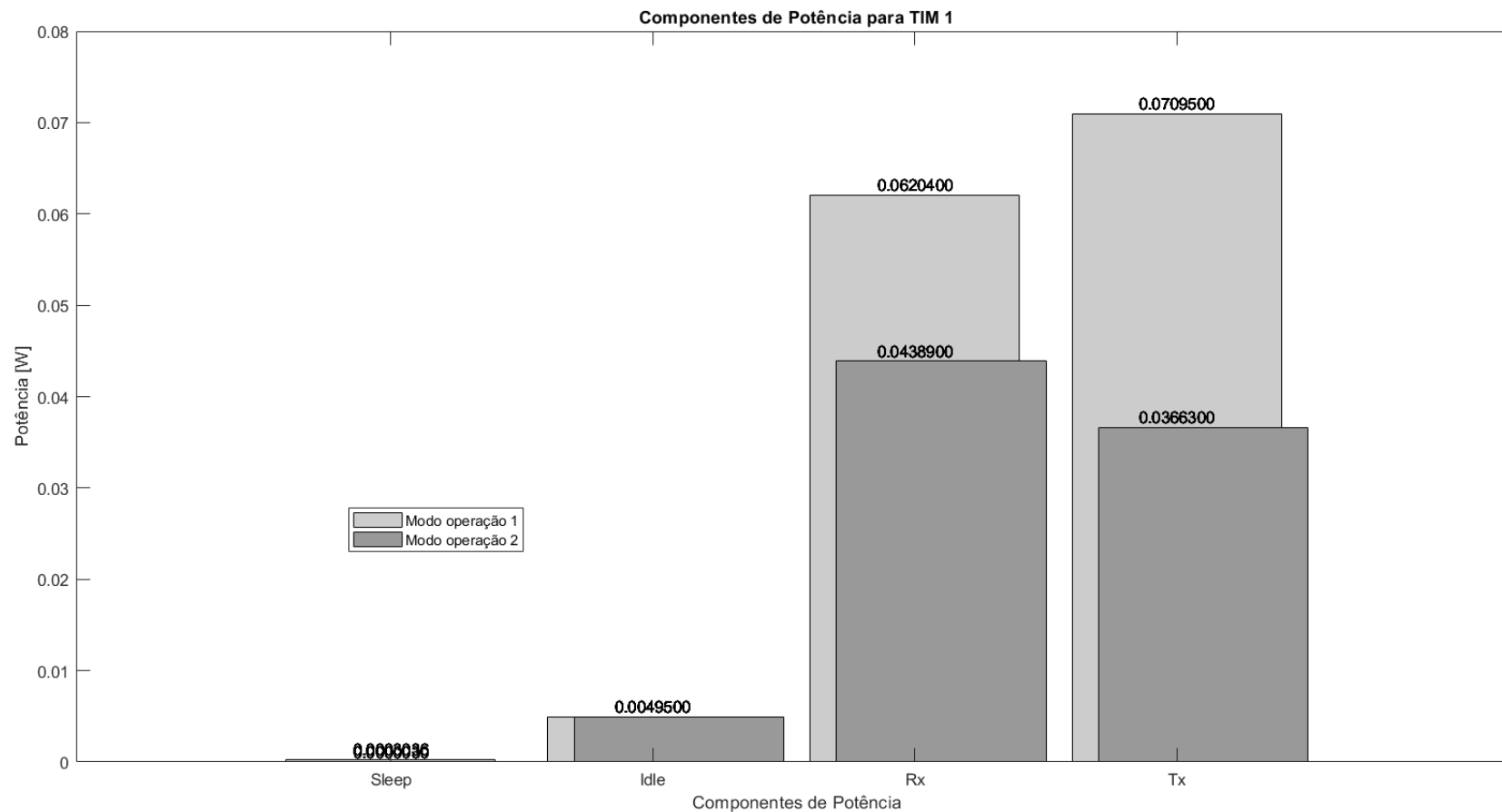


Figura 49 – Valores das componentes que compõem a potência, para cada modo de operação, para Tim 1

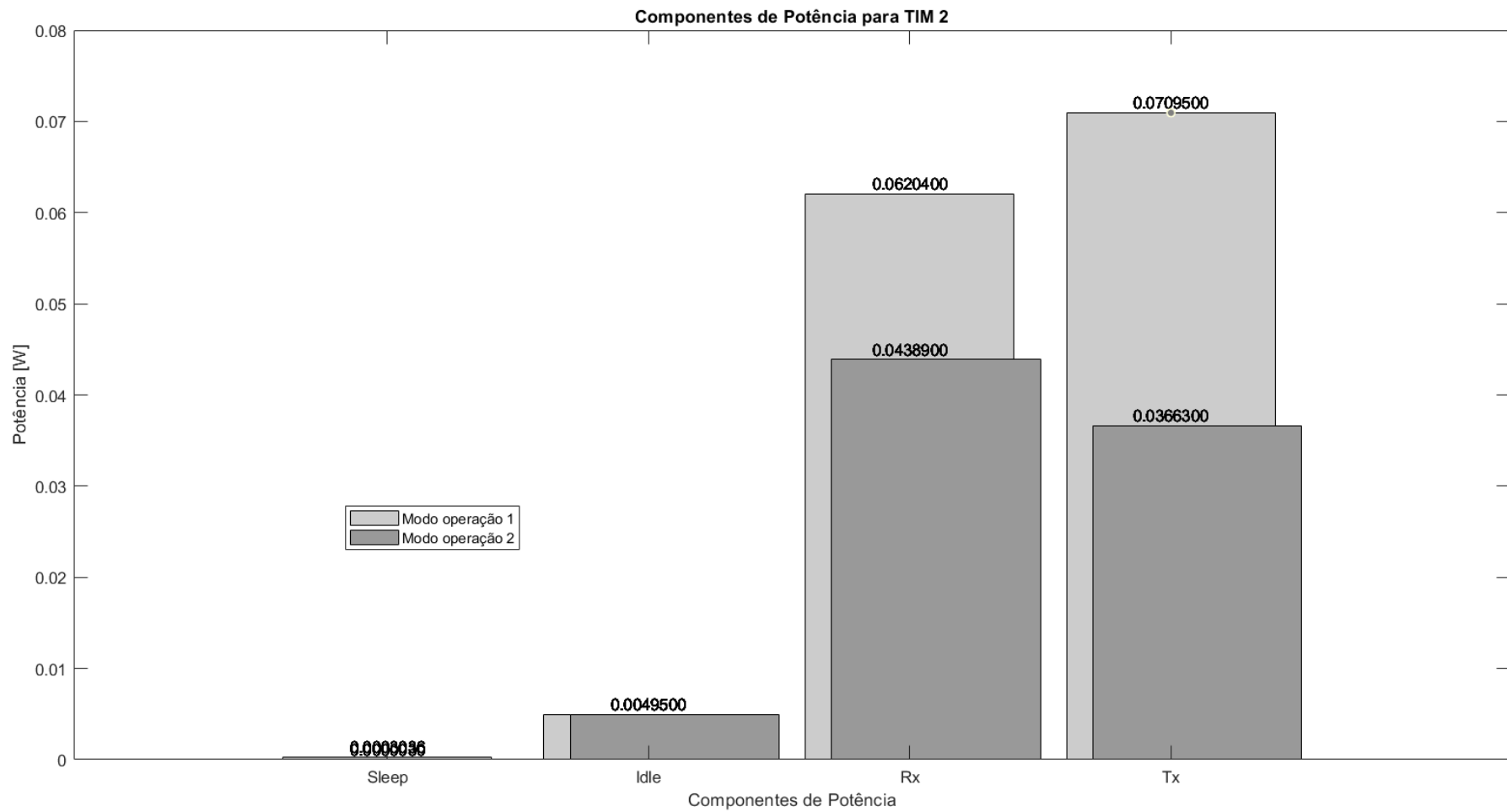


Figura 50 - Valores das componentes que compõem a potência, para cada modo de operação, para Tim 2

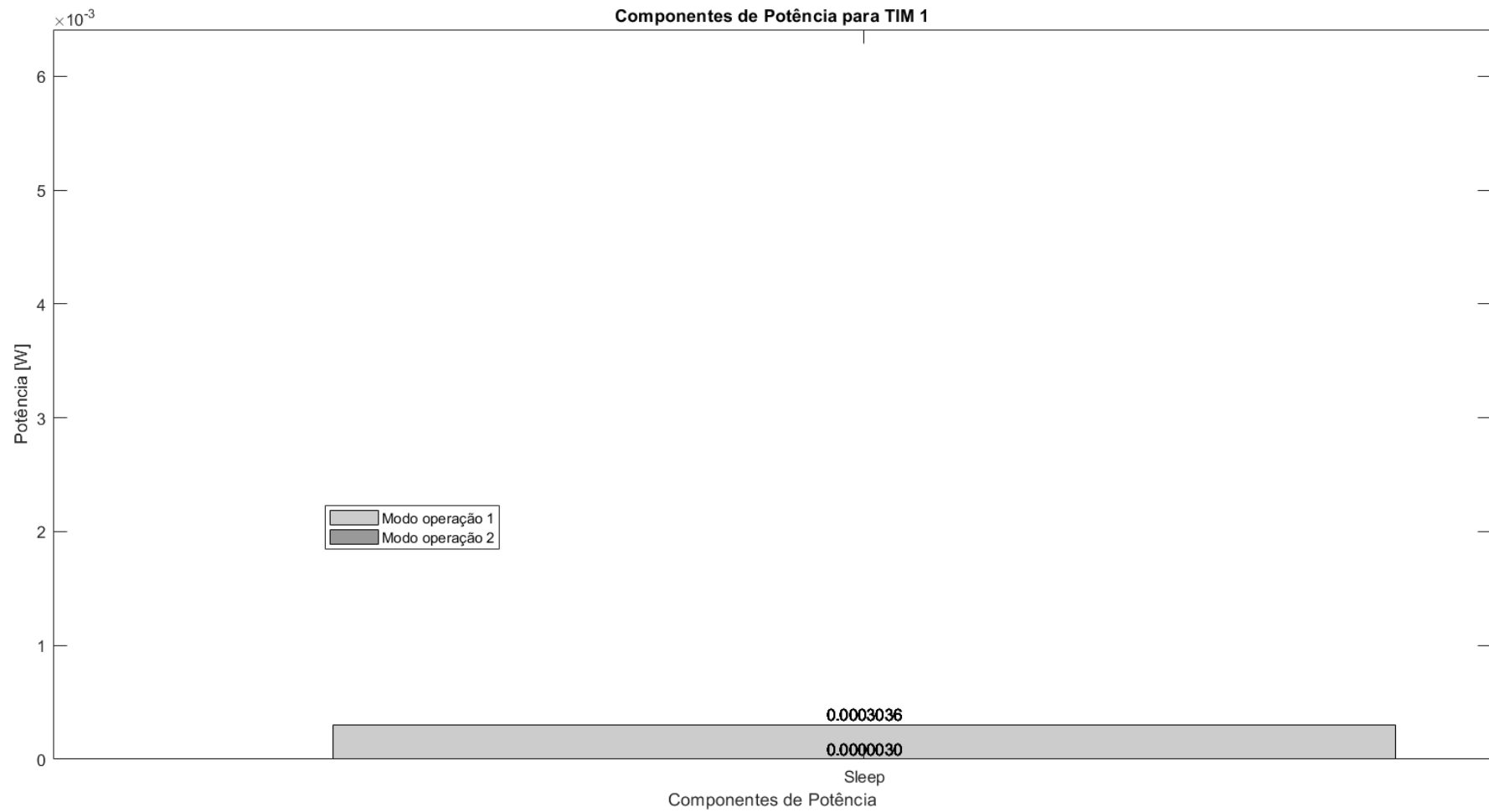


Figura 51 - Valores das componentes do modo sleep, para cada modo de operação, para Tim 1

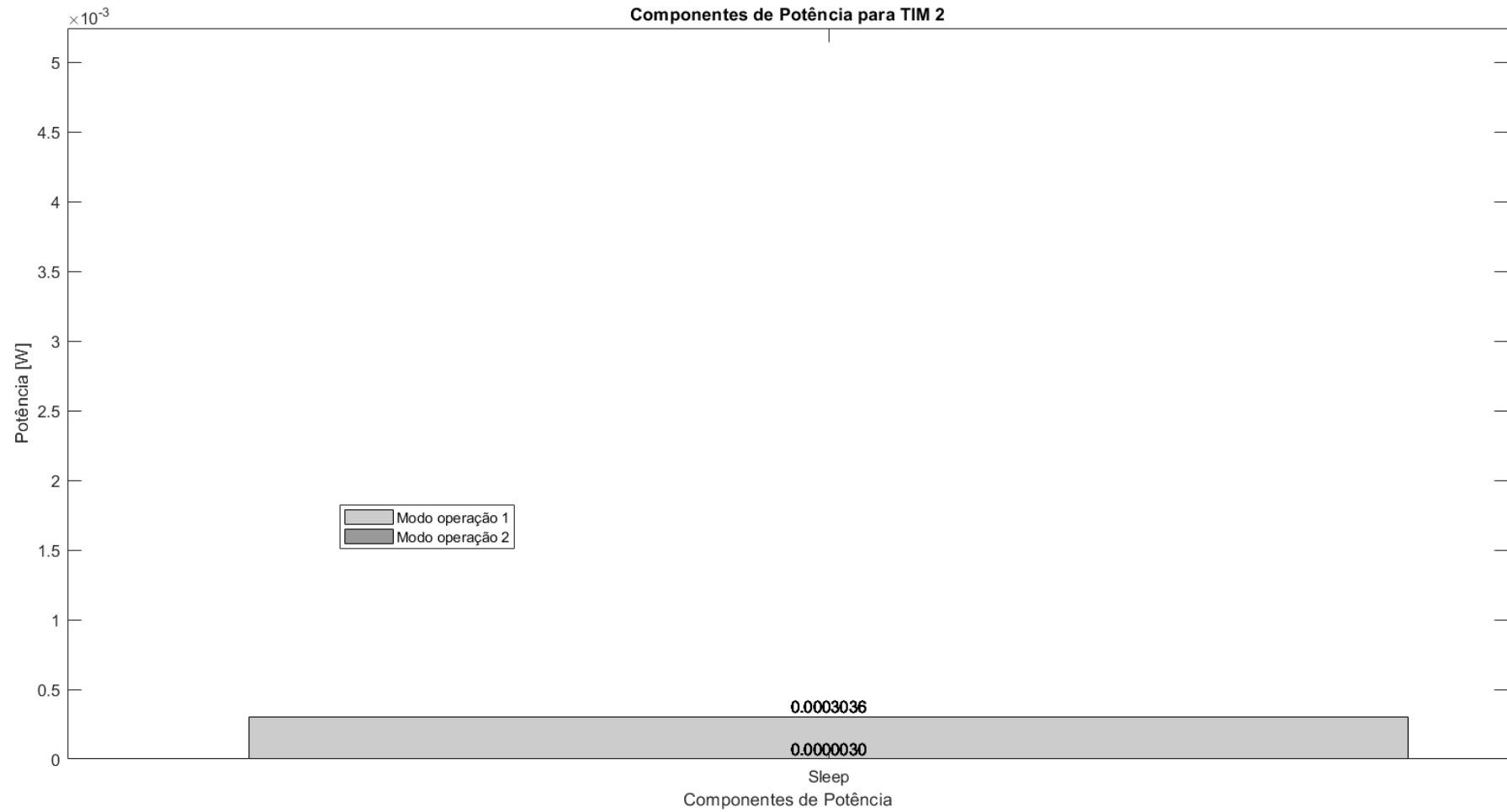


Figura 52 – Zoom dos valores das componentes do modo sleep, para cada modo de operação, para Tim 2

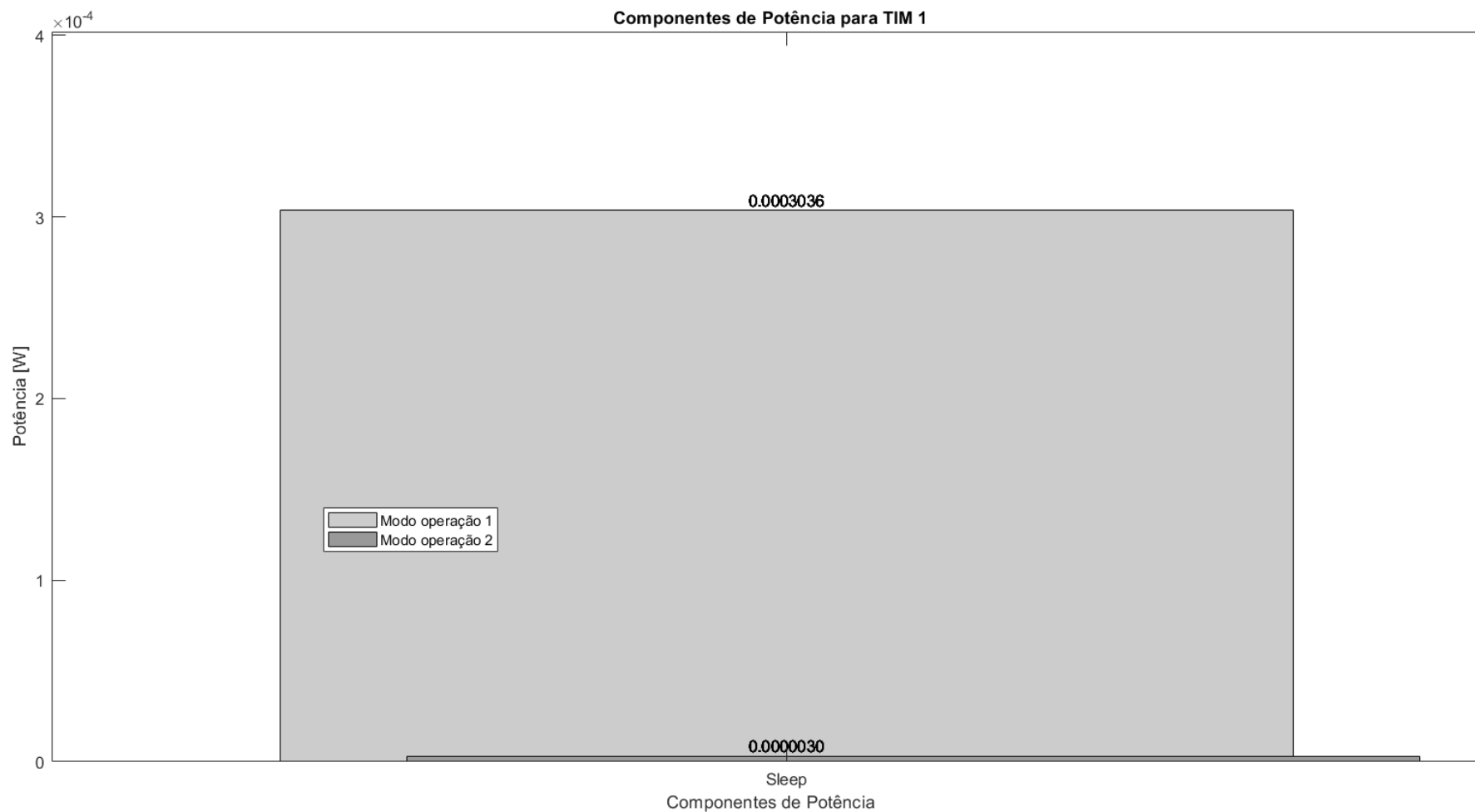


Figura 53 - Zoom dos valores das componentes do modo sleep, para cada modo de operação

### 4.1.2.2. Simulação 1

- Valores de  $k * \Theta$  otimizado

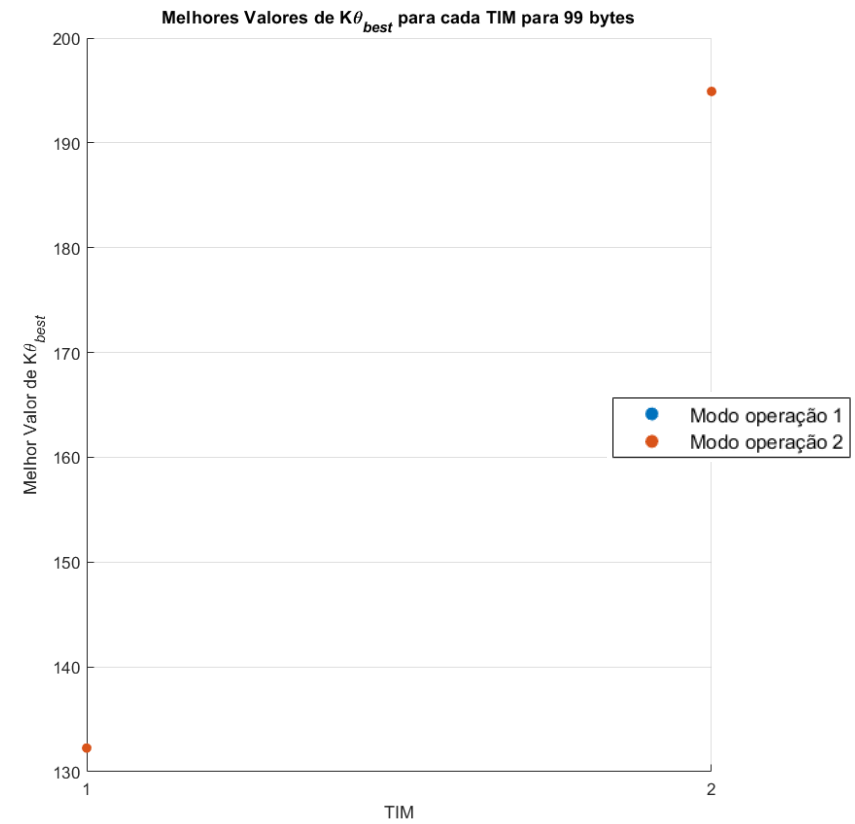
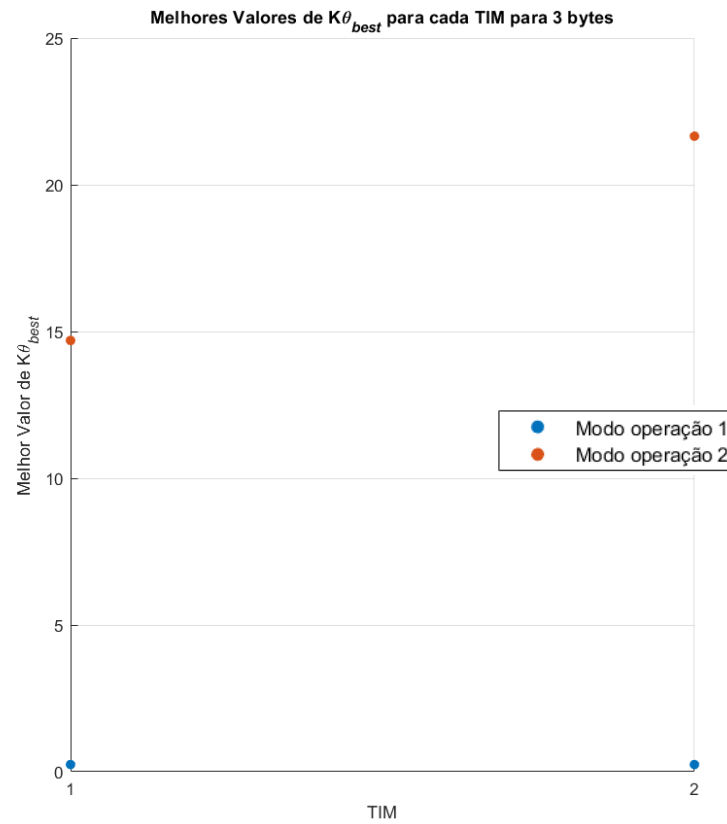


Figura 54 – Valores otimizados de  $k \cdot \Theta$ , para cada modo de operação e cada cenário A e B

- Comparação energética**

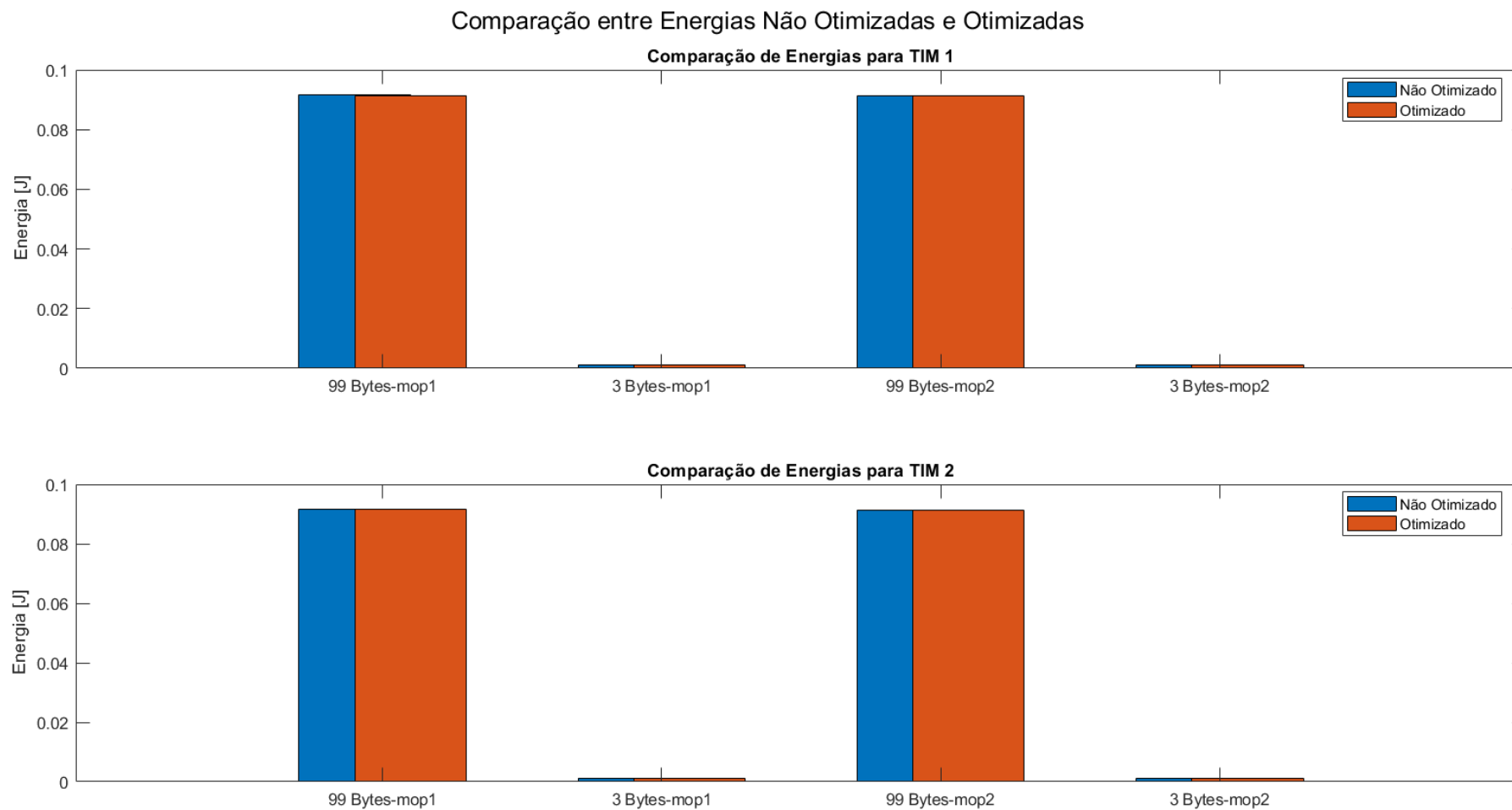


Figura 55 – Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação

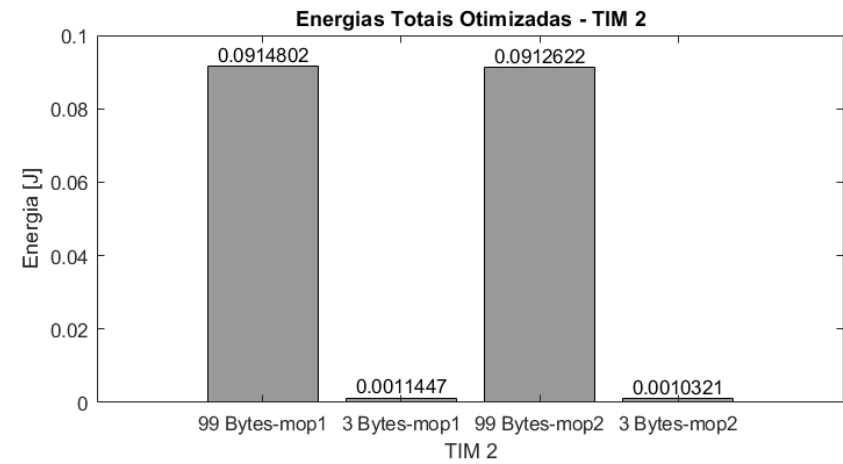
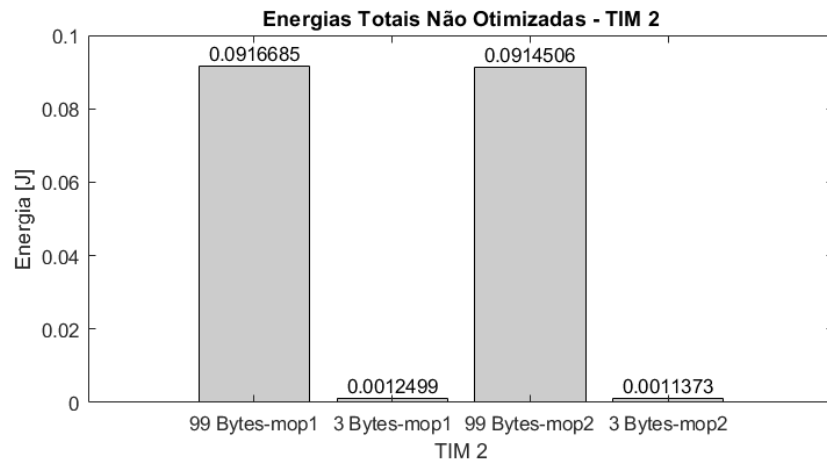
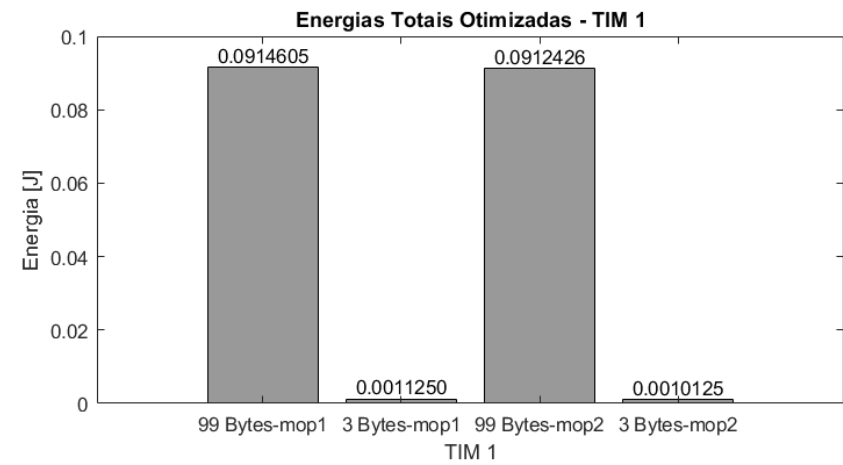
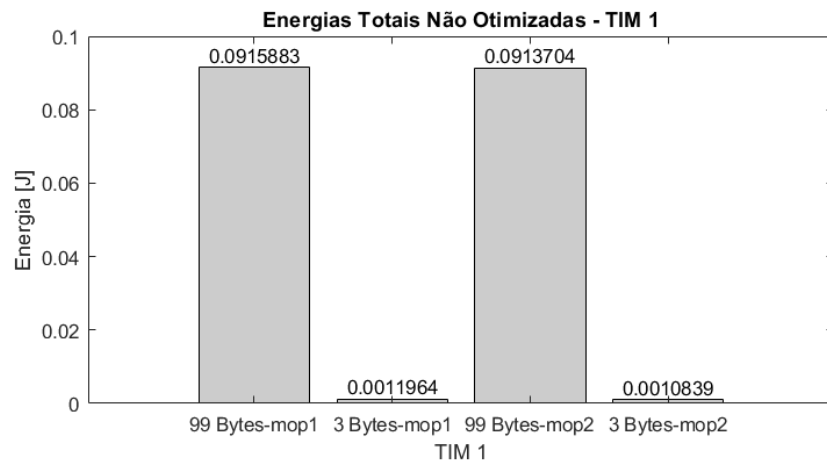


Figura 56 – Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B

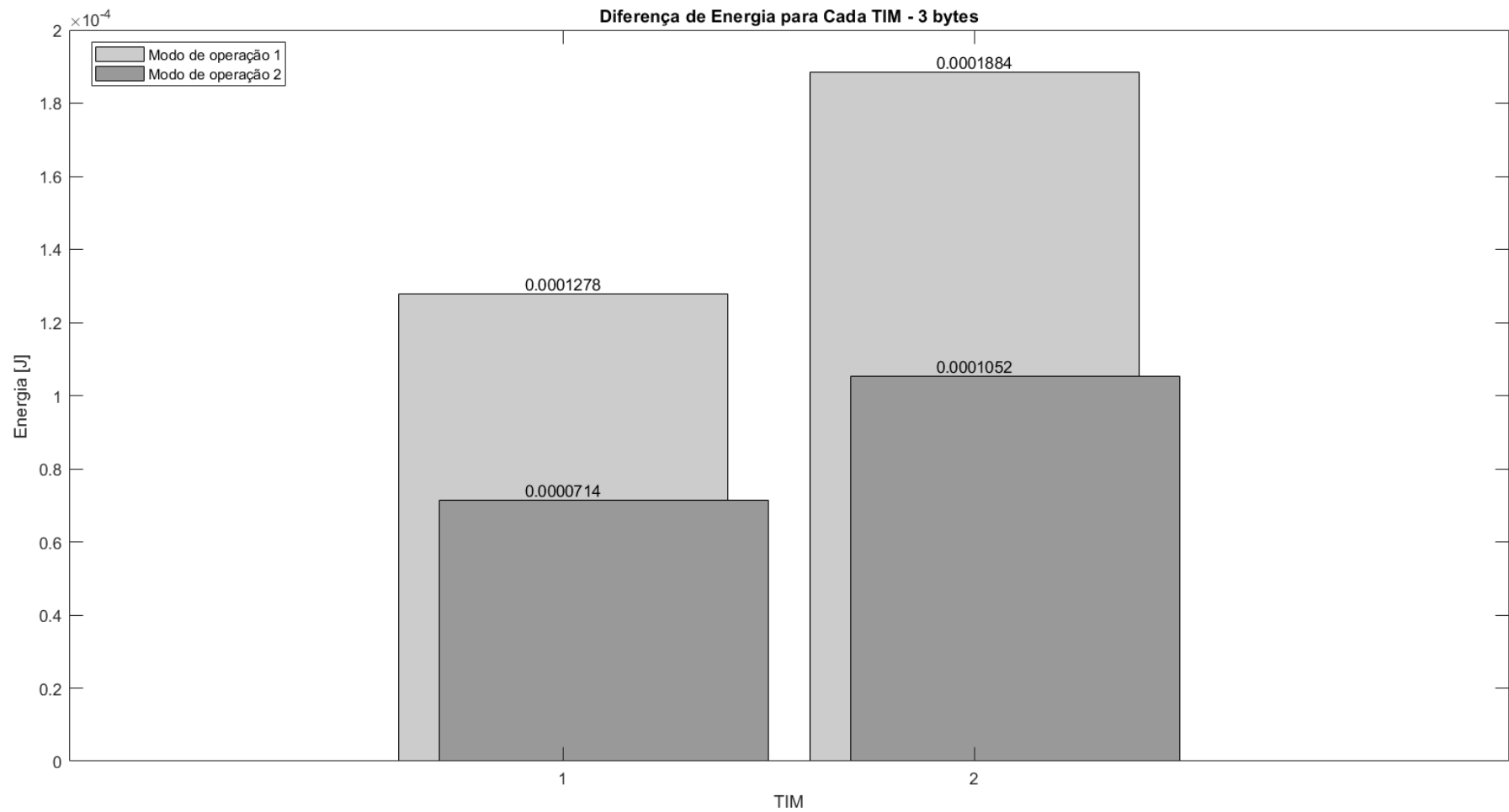


Figura 57 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM

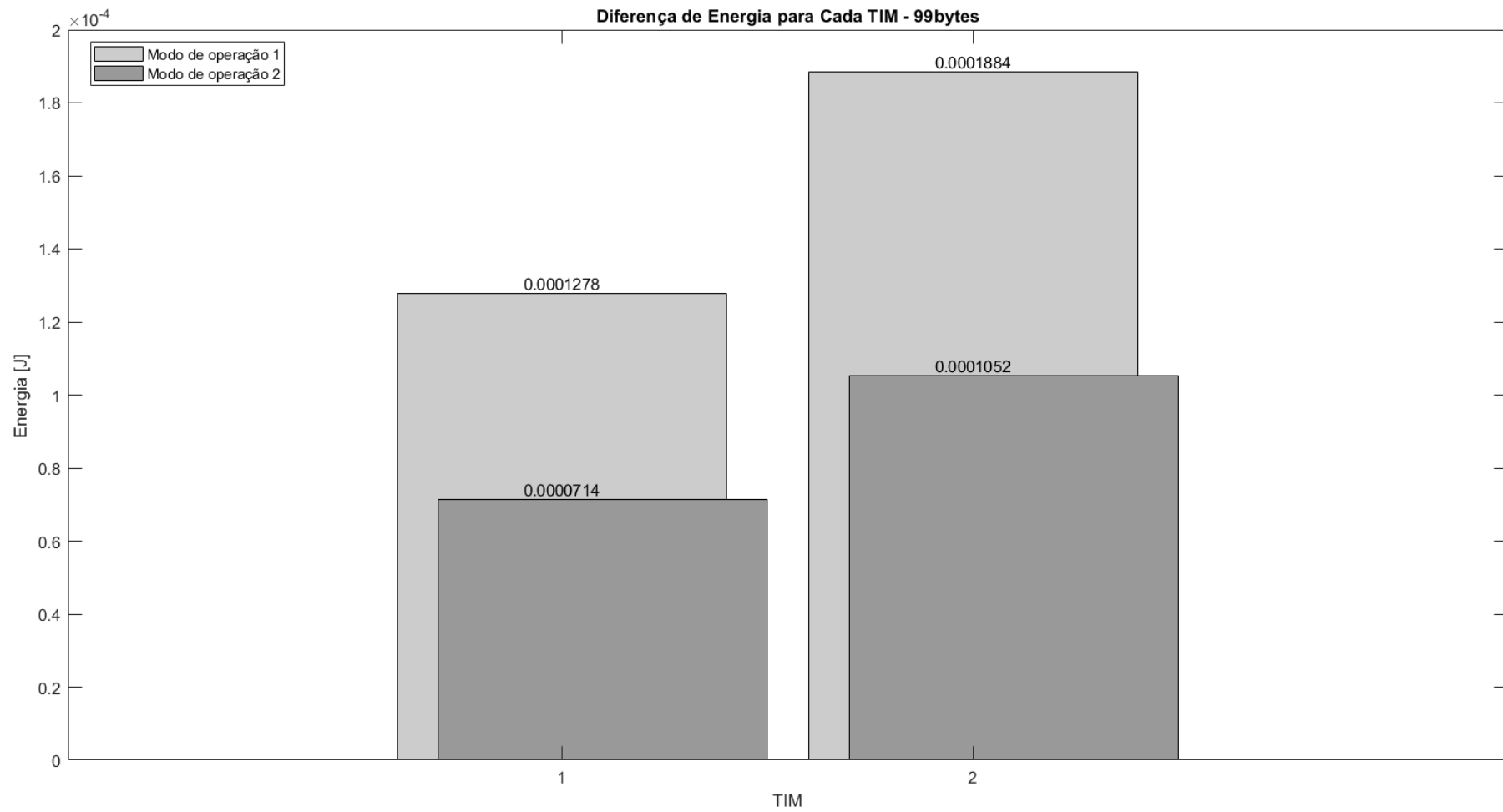


Figura 58 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM

### 4.1.2.3. Simulação 2

- Valores de  $k * \Theta$  otimizado

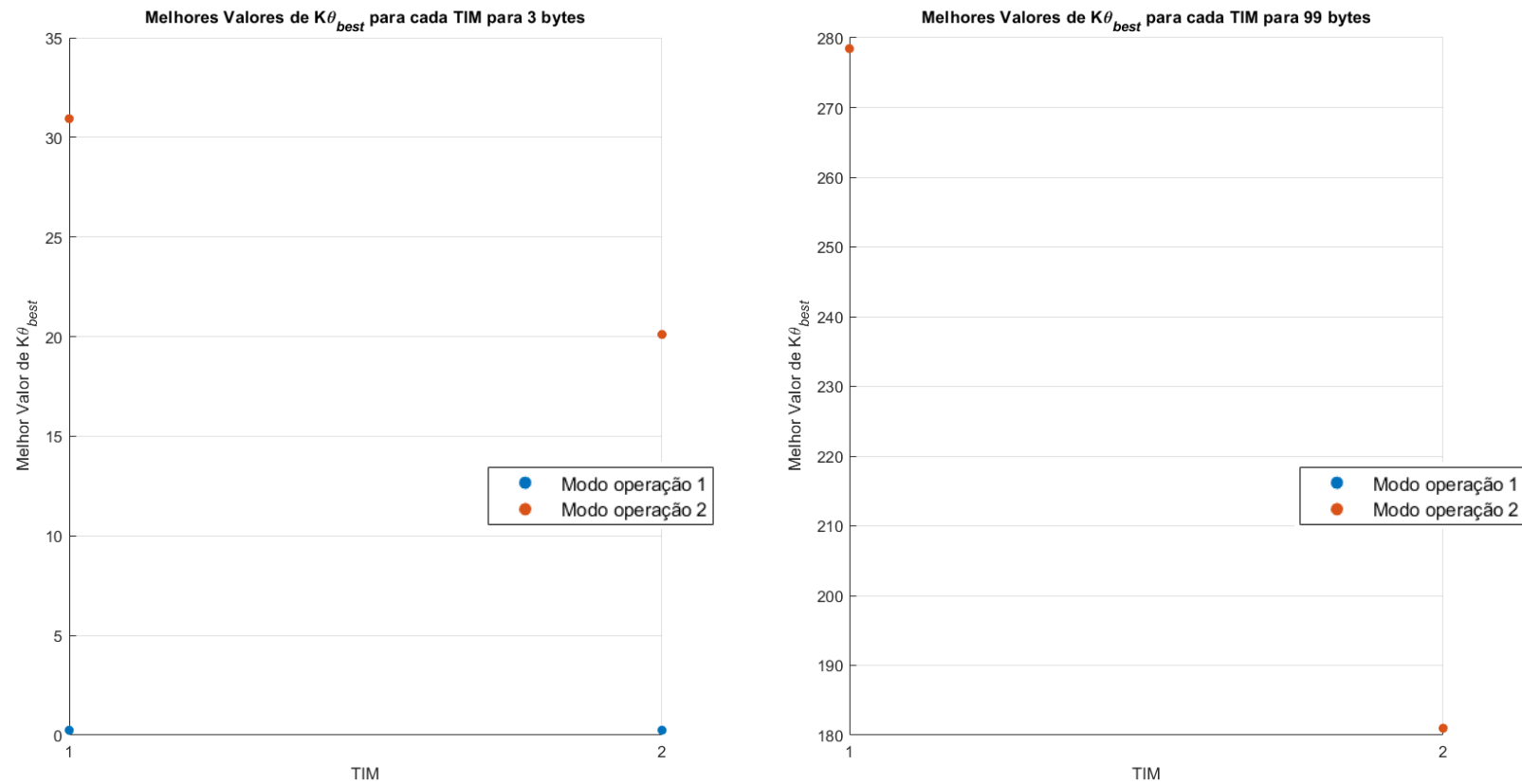


Figura 59 - Valores otimizados de  $k * \Theta$ , para cada modo de operação e cada cenário A e B

- Comparação energética**

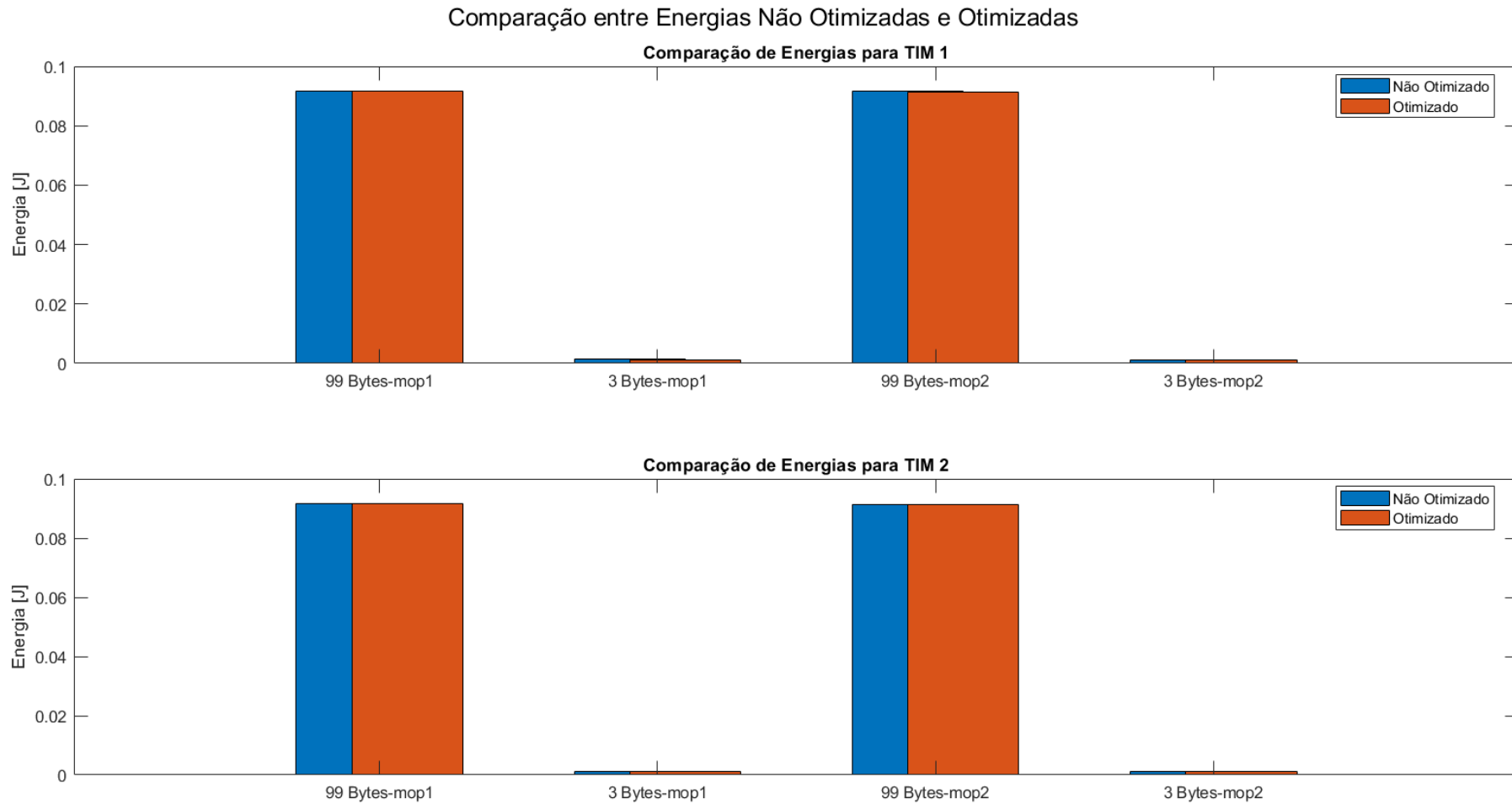


Figura 60 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação

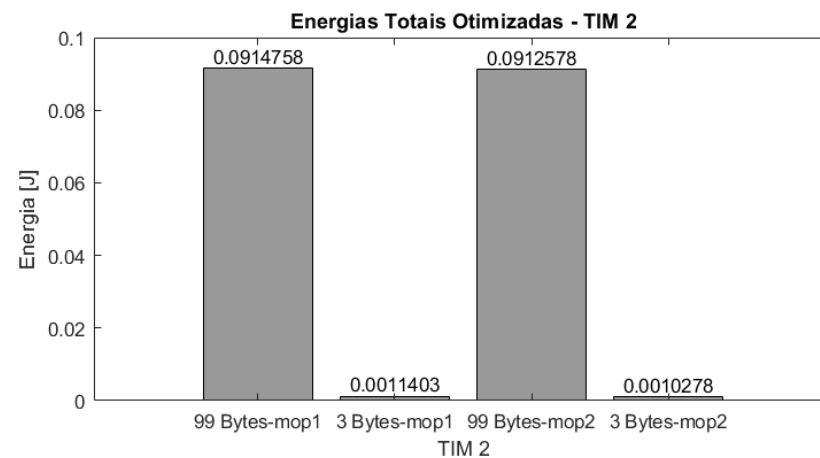
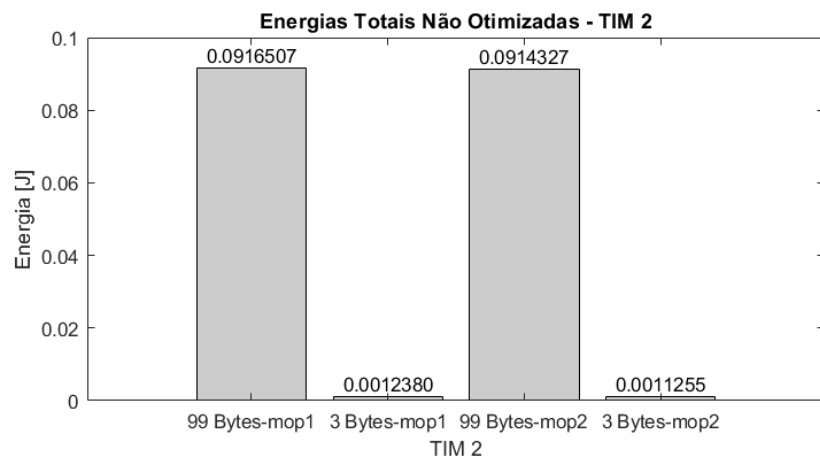
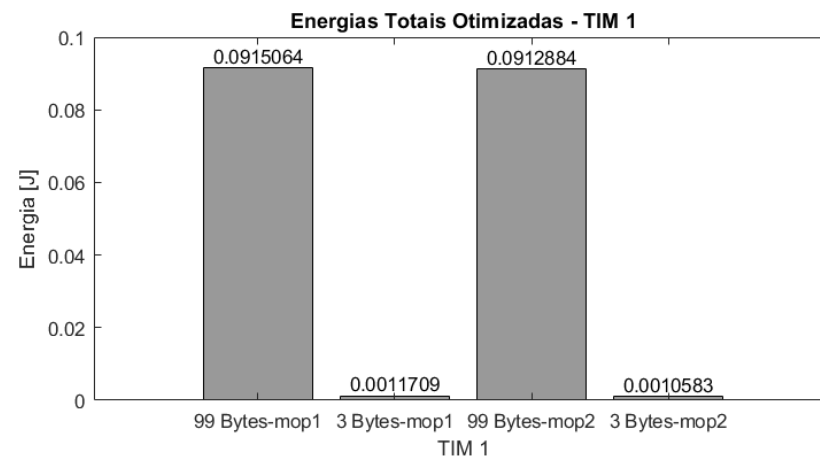
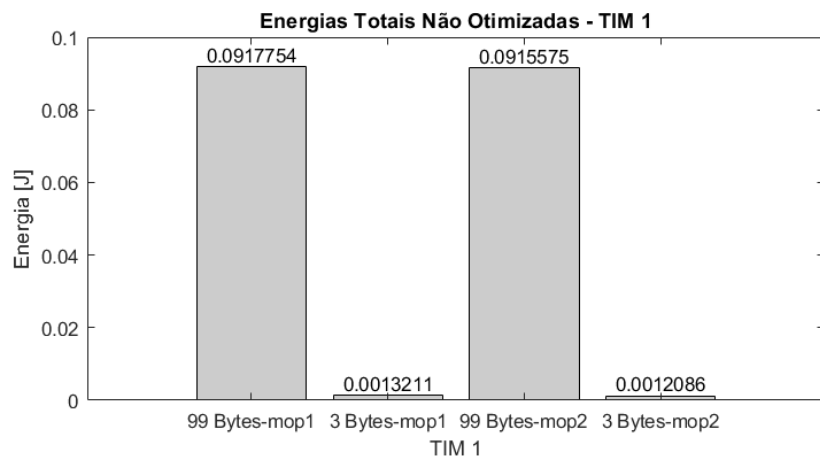


Figura 61 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B

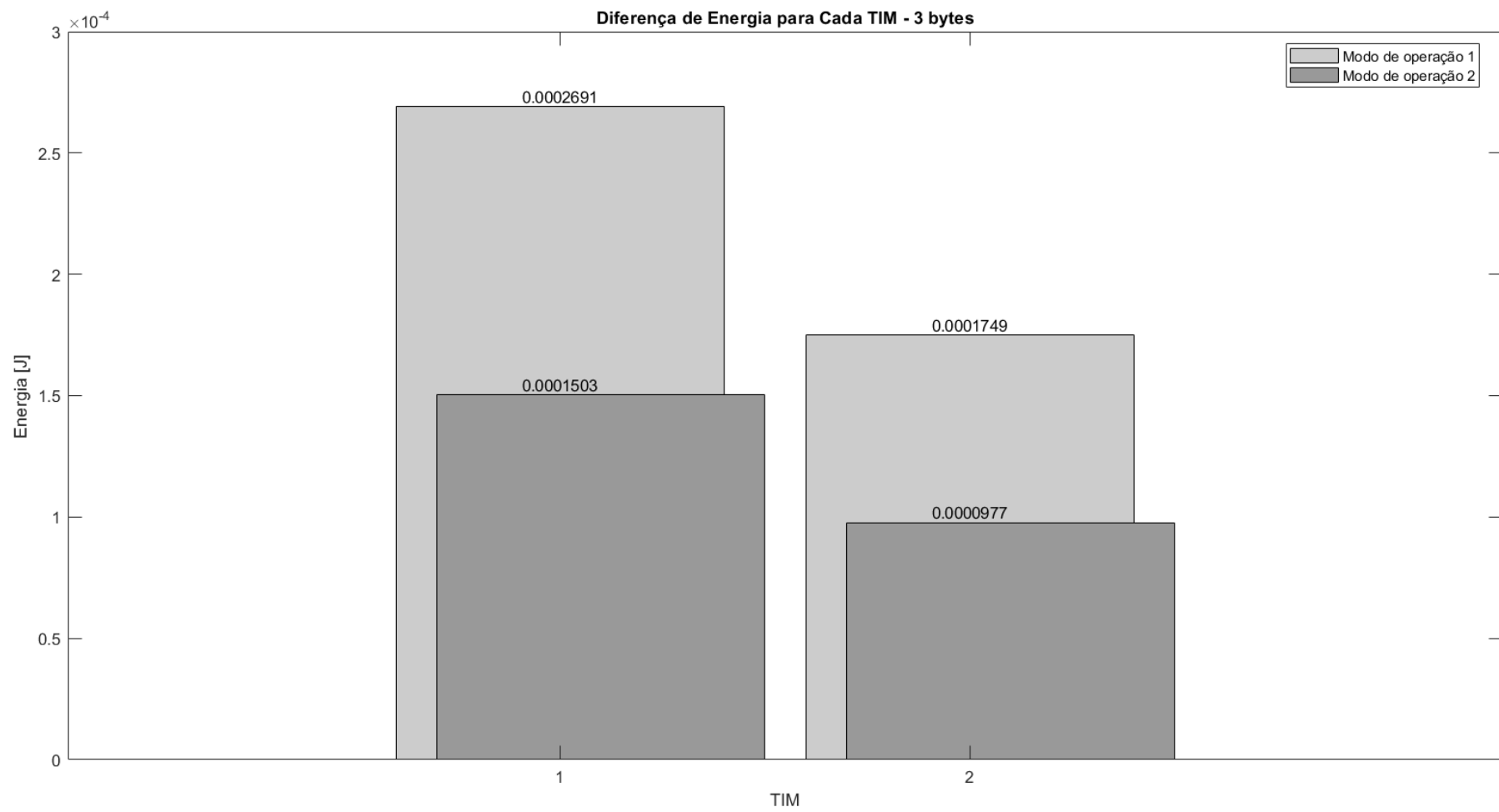


Figura 62 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM

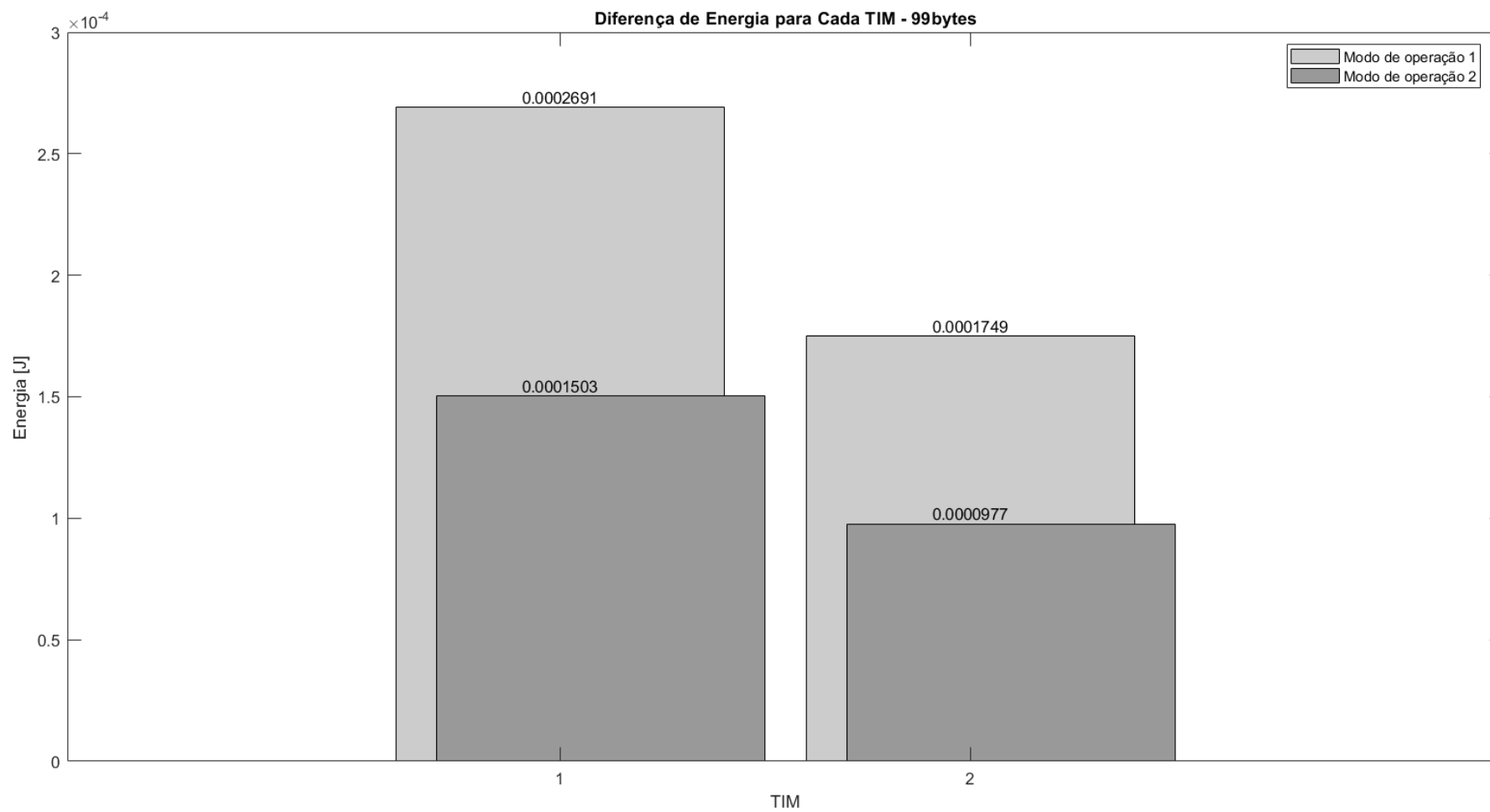


Figura 63 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM

#### 4.1.2.4. Simulação 3

- Valores de  $k * \Theta$  otimizado

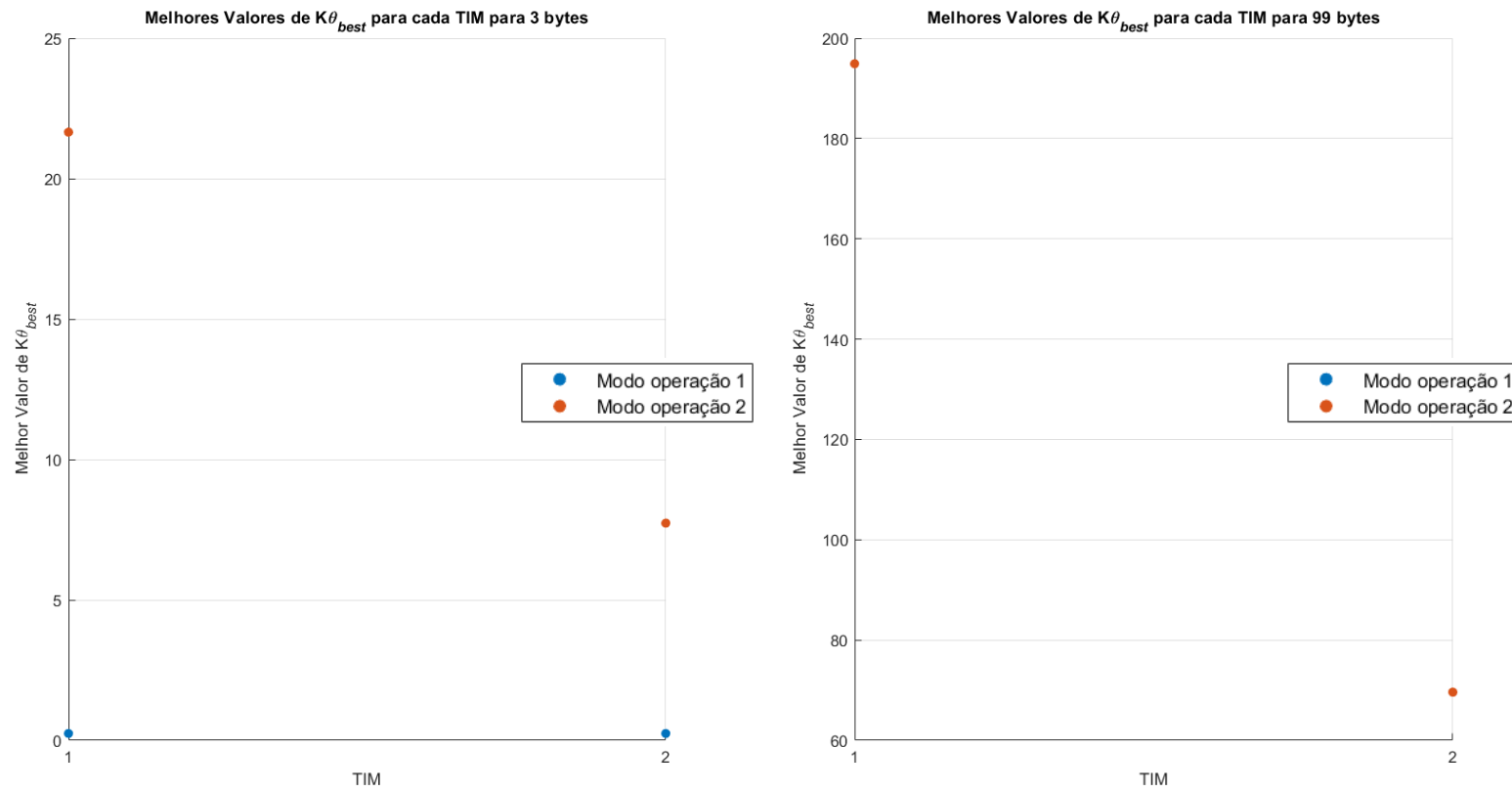


Figura 64 - Valores otimizados de  $k \cdot \Theta$ , para cada modo de operação e cada cenário A e B

- Comparação energética**

Comparação entre Energias Não Otimizadas e Otimizadas

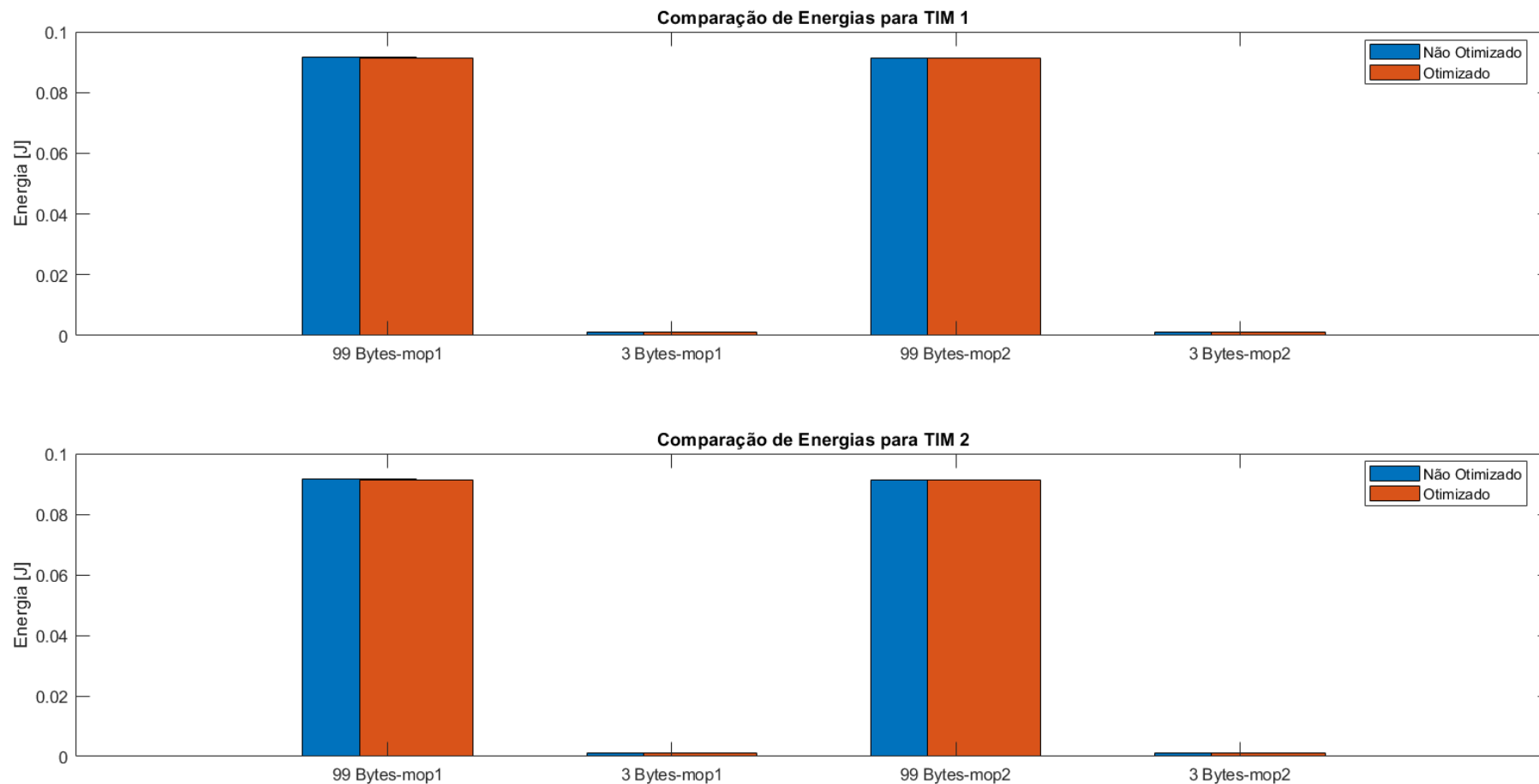


Figura 65 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação

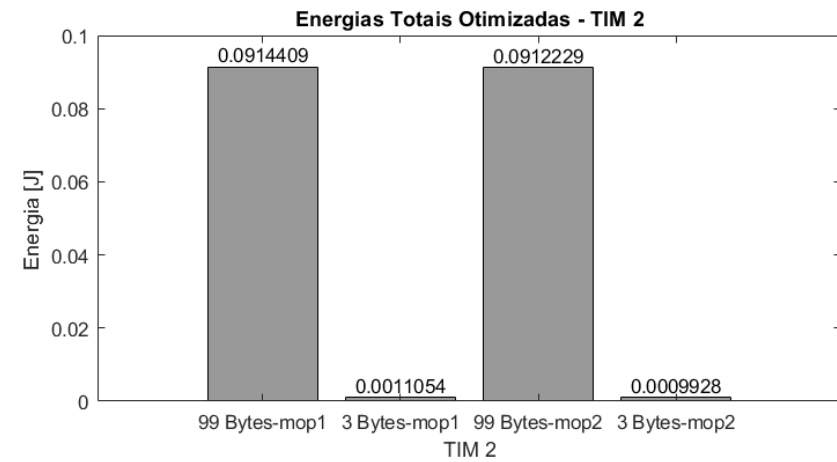
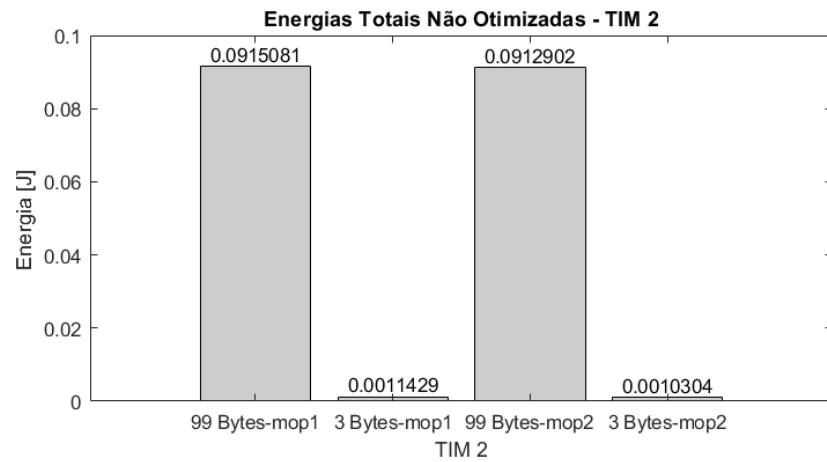
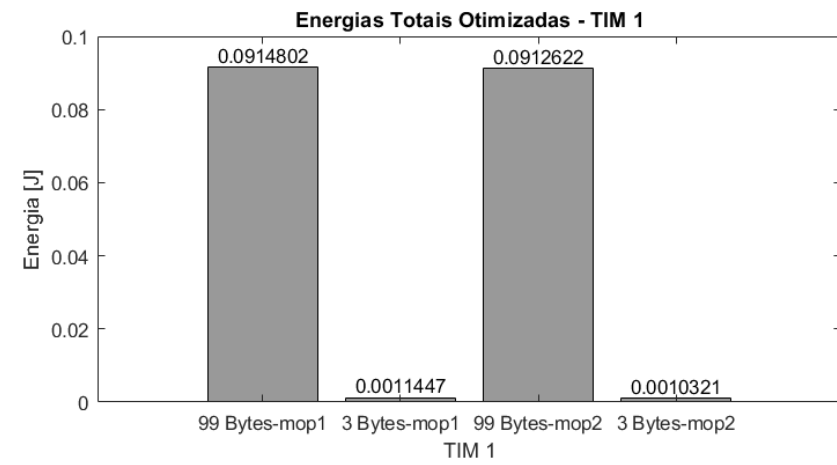
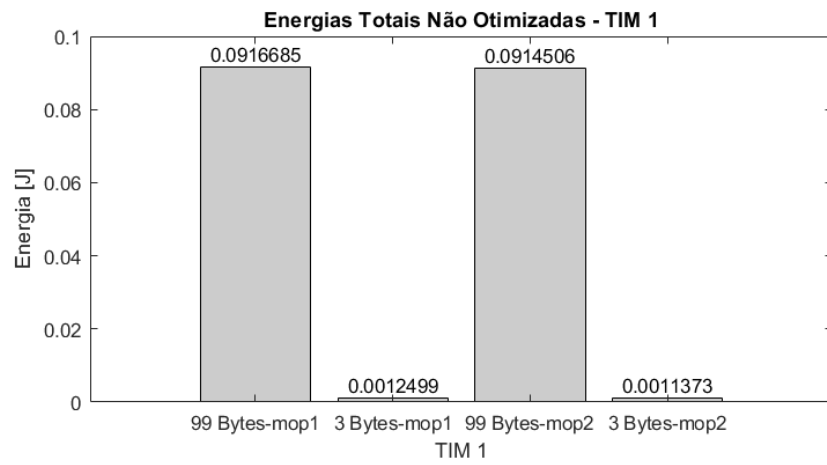


Figura 66 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B

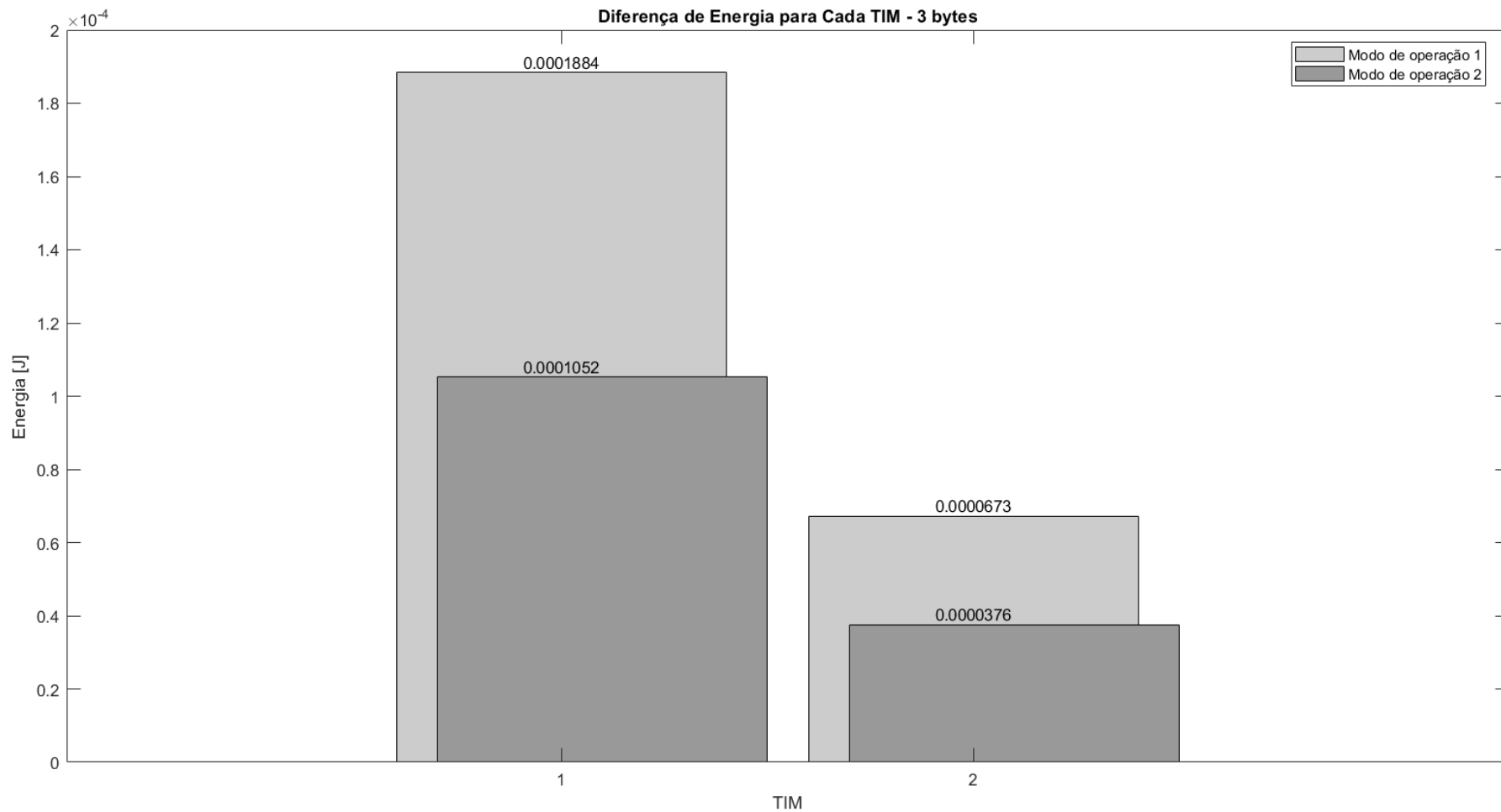


Figura 67 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM

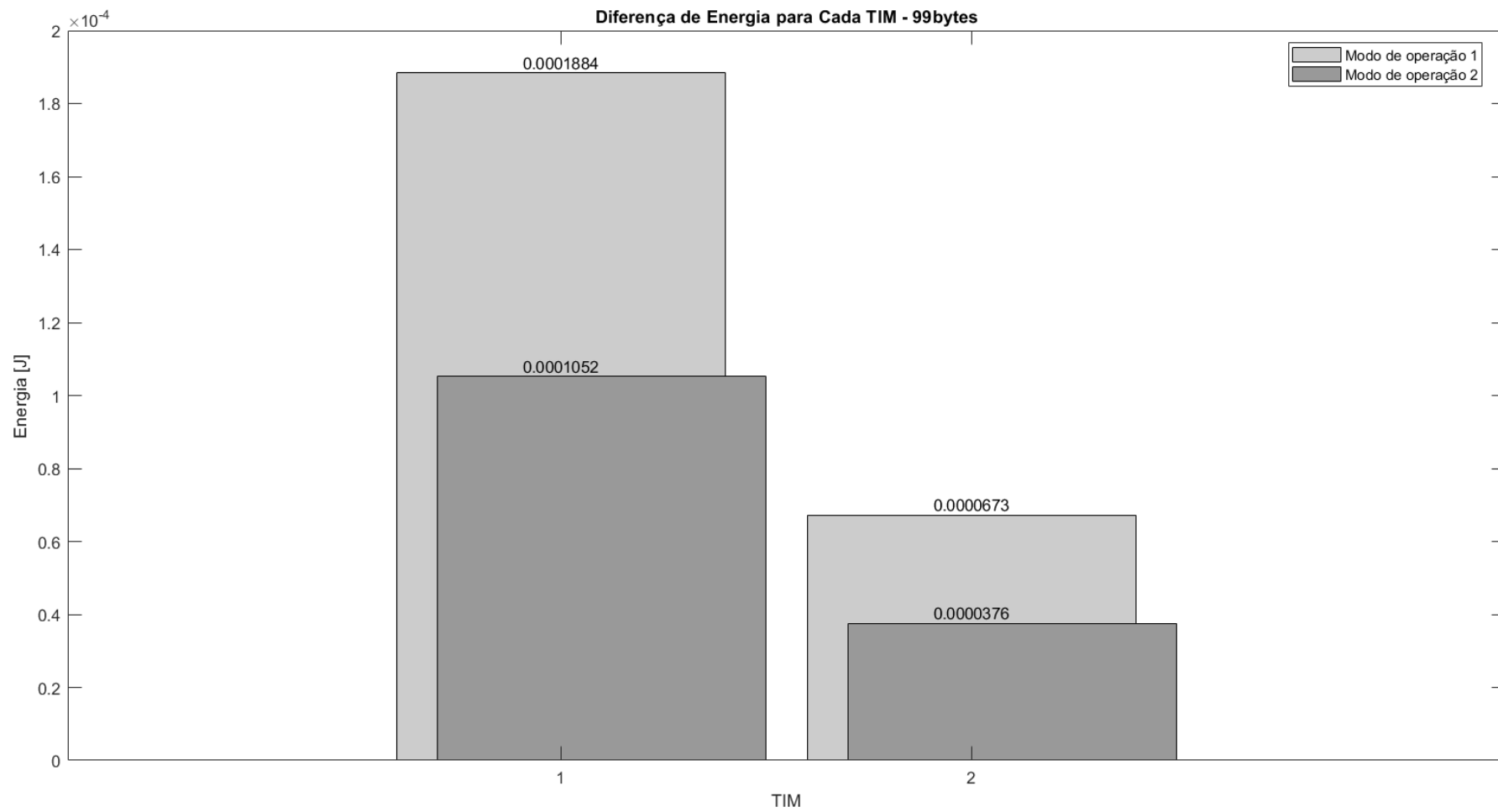


Figura 68 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM

### 4.1.2.5. Simulação 4

- Valores de  $k * \Theta$  otimizado

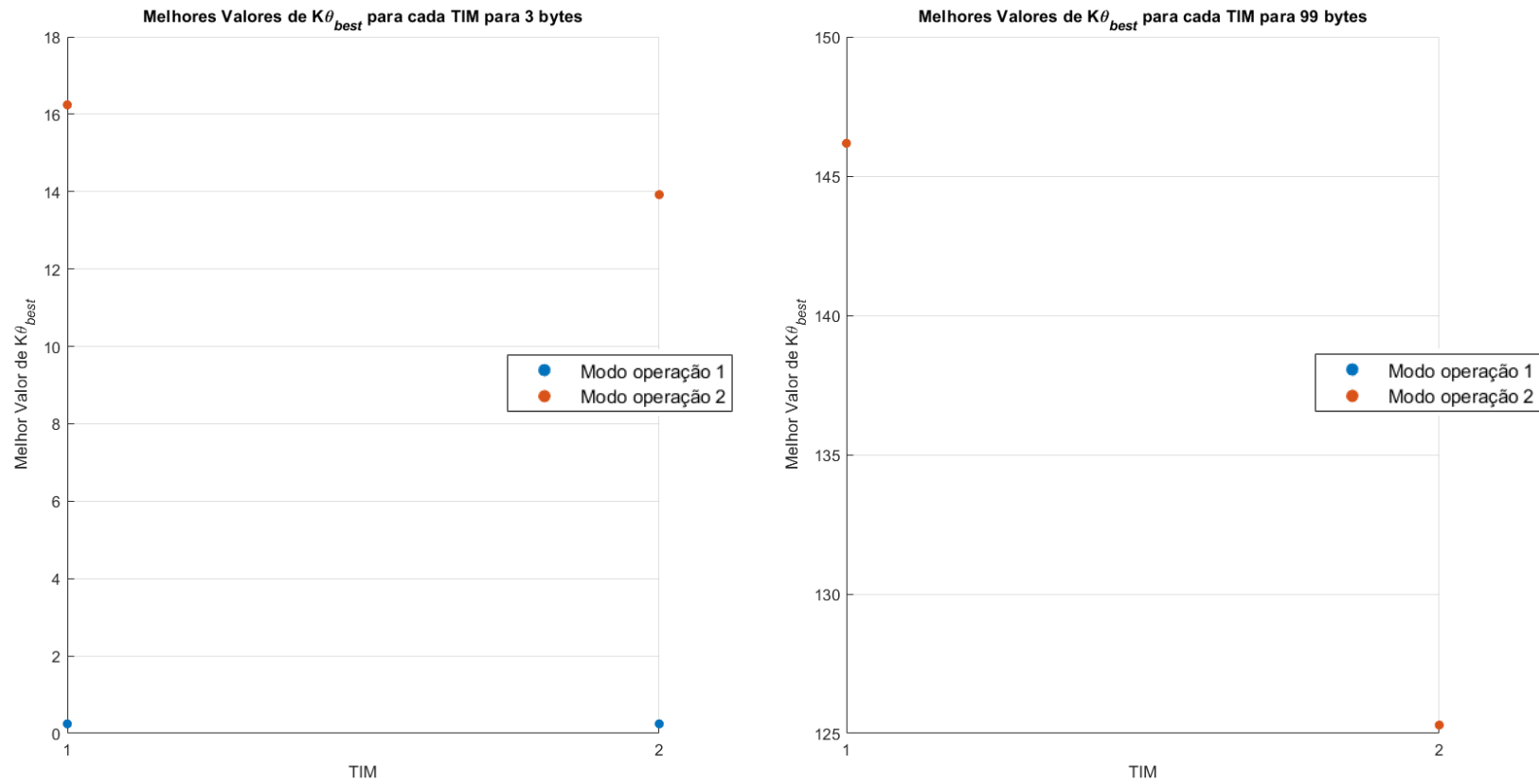


Figura 69 - Valores otimizados de  $k * \Theta$ , para cada modo de operação e cada cenário A e B

• **Comparação energética**

Comparação entre Energias Não Otimizadas e Otimizadas

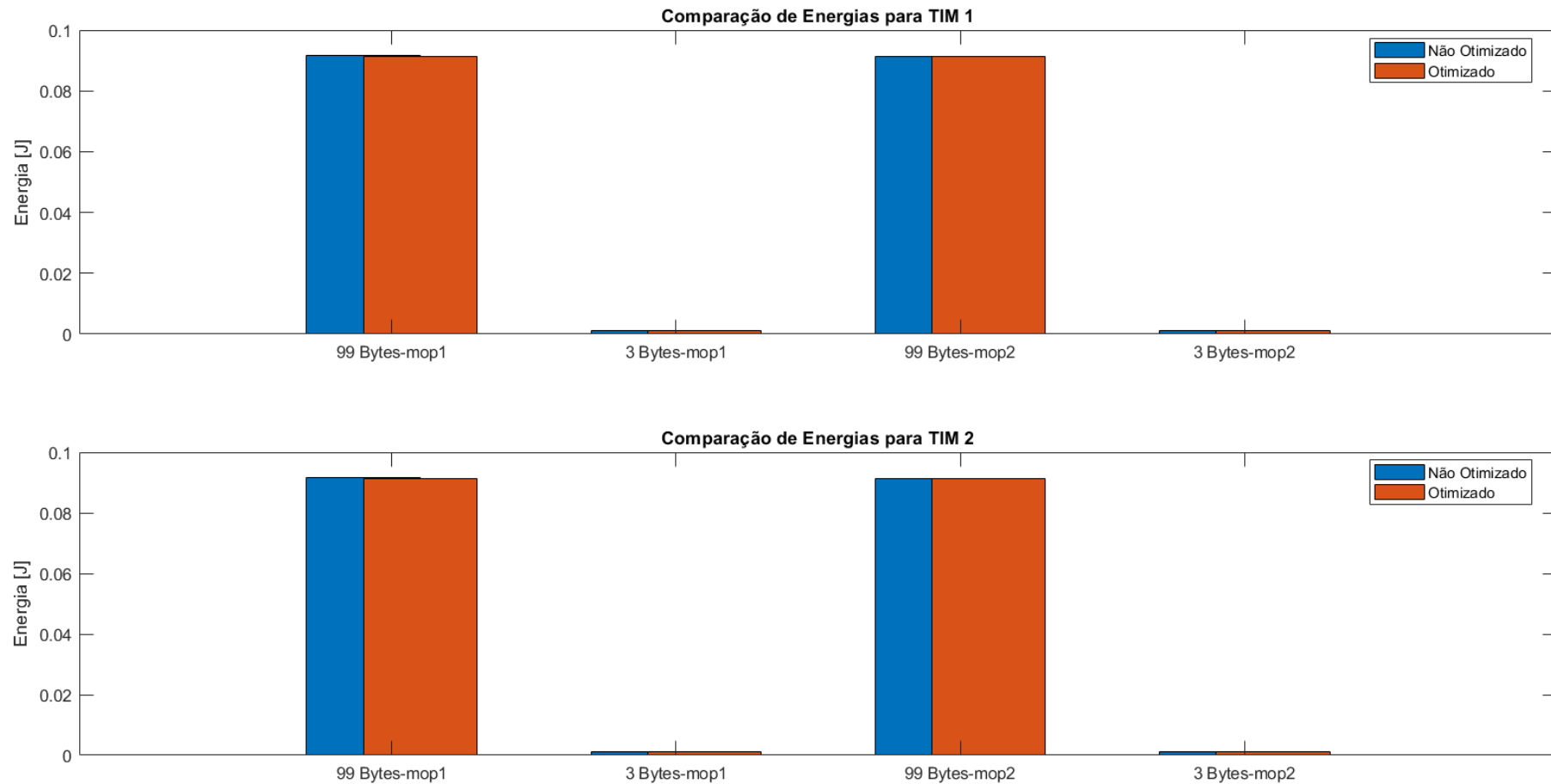


Figura 70 - Comparação entre as energias originais e as otimizadas de cada TIM, para cada cenário e cada modo de operação

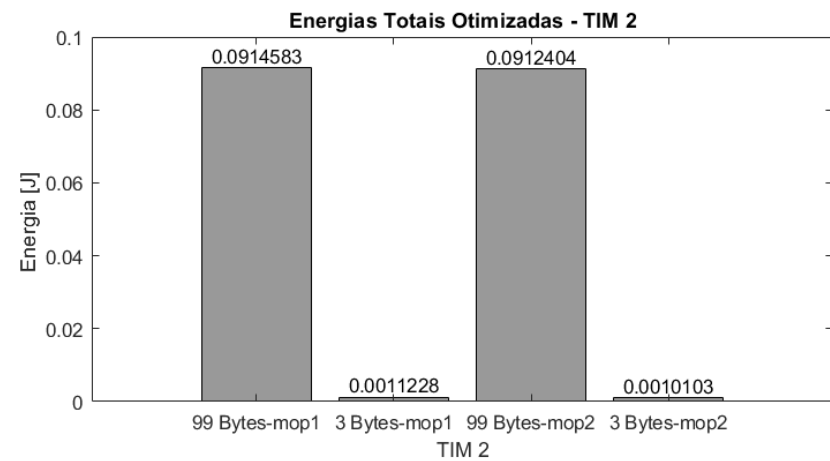
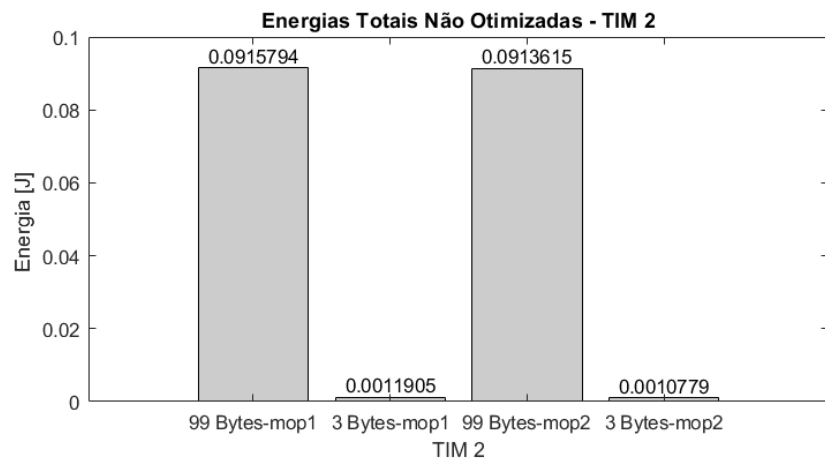
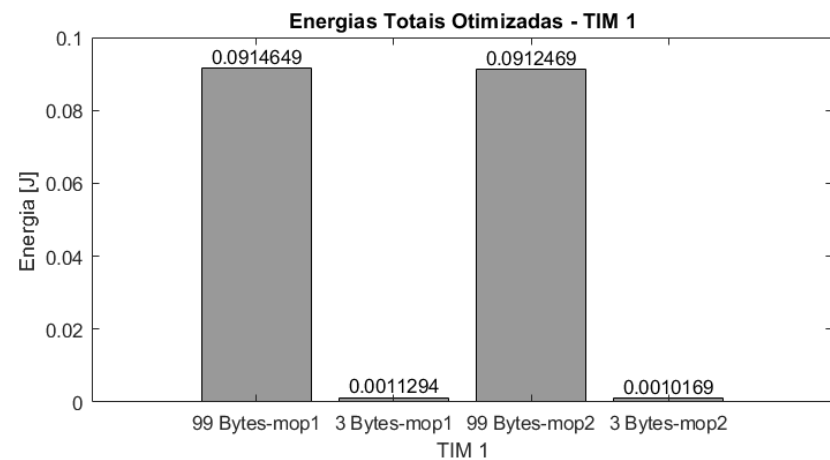
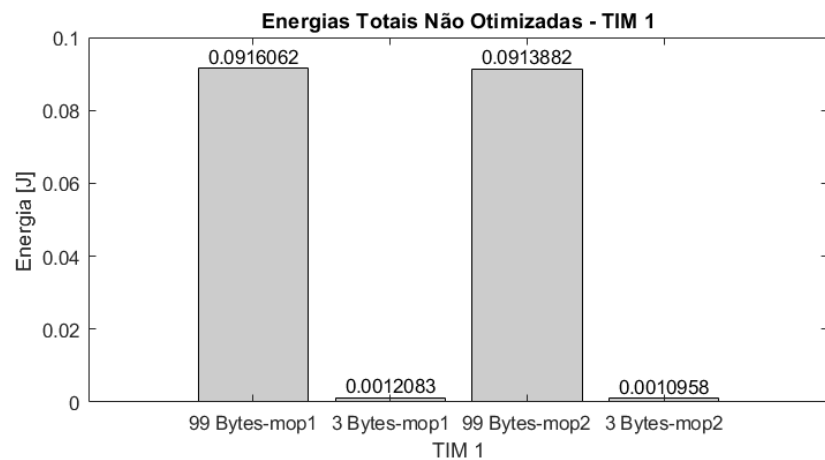


Figura 71 - Valores das energias totais para cada TIM, cada modo de operação e cada cenário A e B

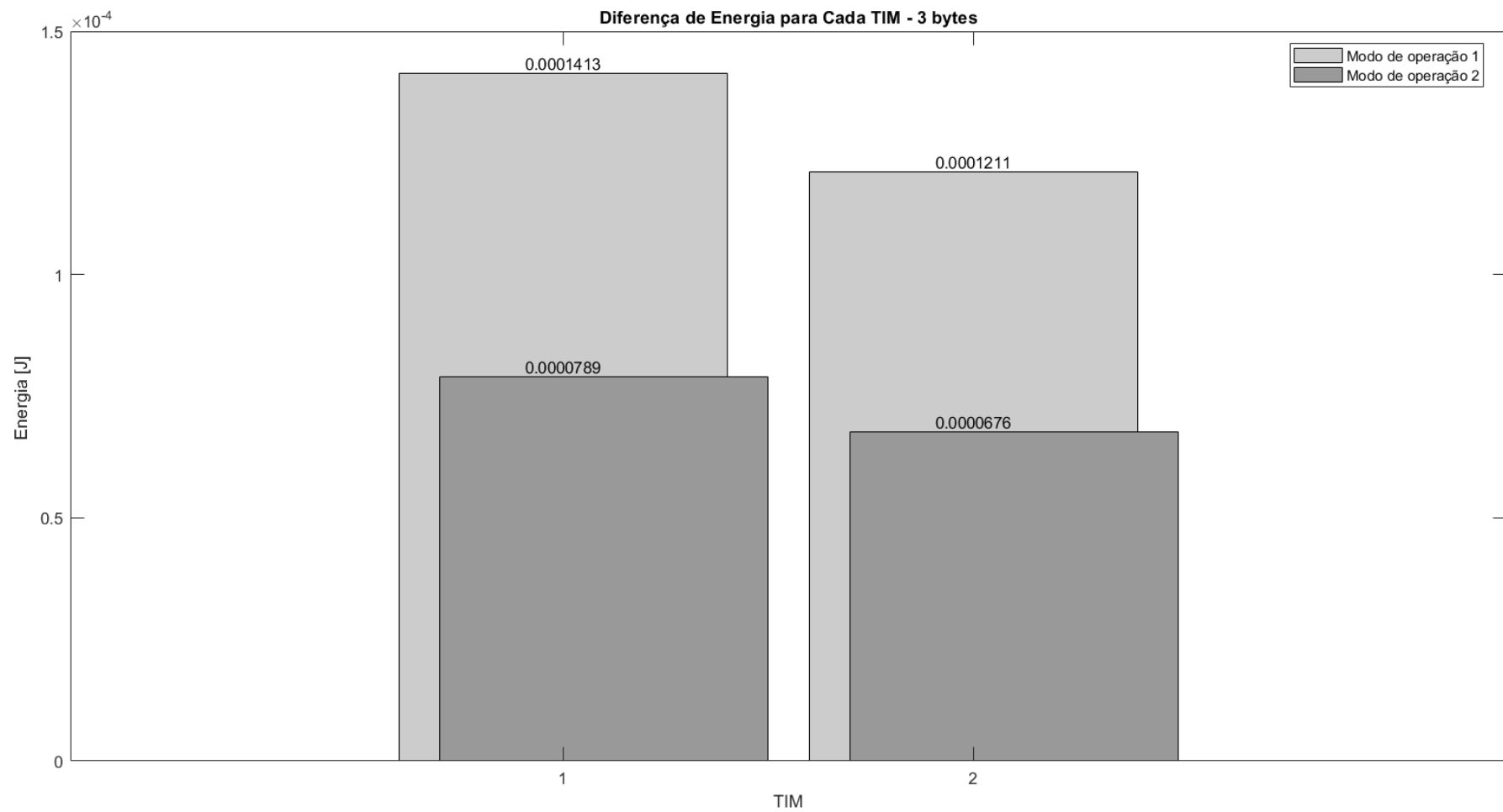


Figura 72 - Diferença entre as energias originais e as otimizadas, cenário A, para cada modo de operação e cada TIM

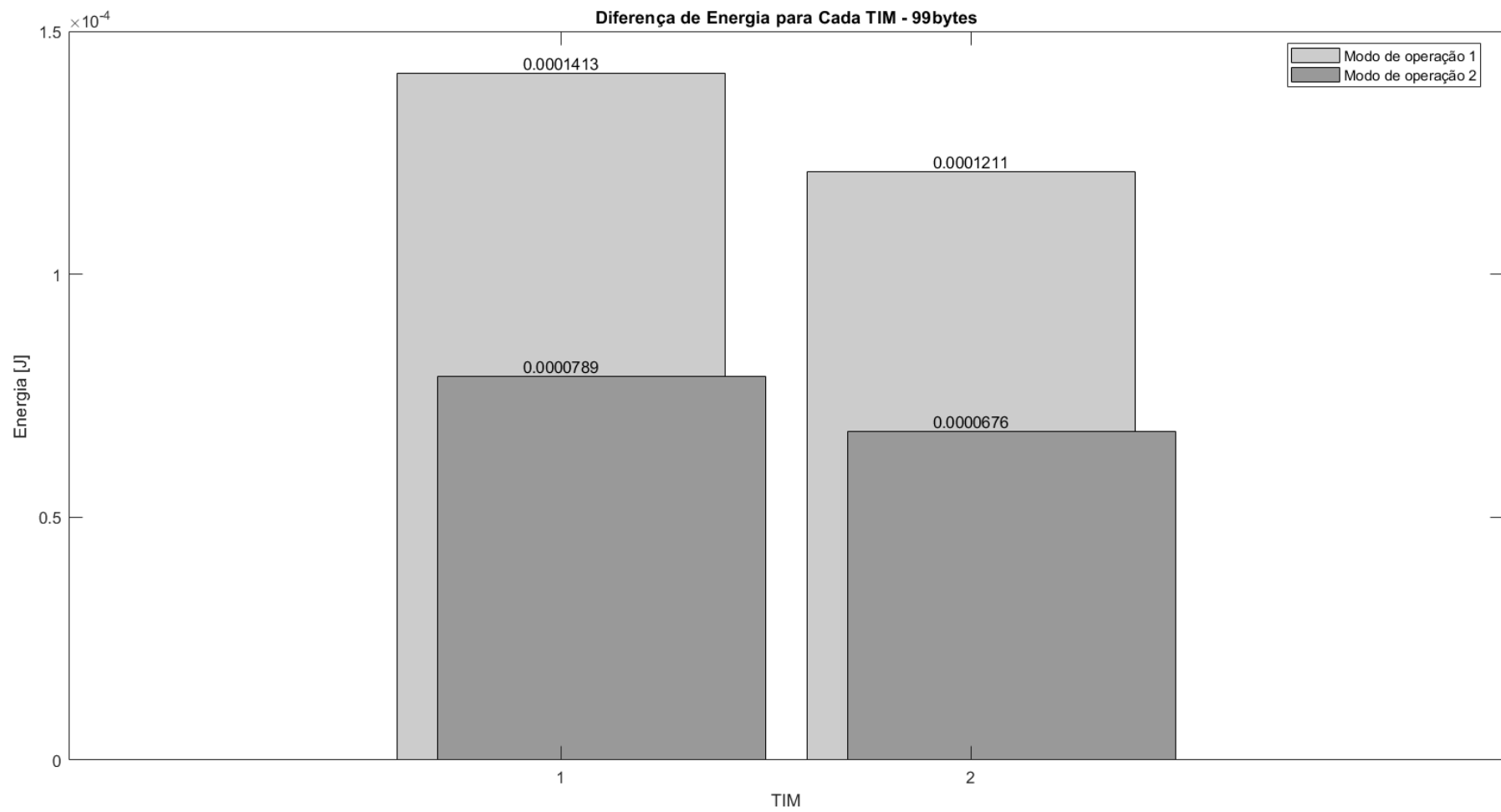


Figura 73 - Diferença entre as energias originais e as otimizadas, cenário B, para cada modo de operação e cada TIM

### 4.1.3. Análise dos resultados

Da análise dos gráficos anteriores observa-se e comprova-se que após o tempo de escuta ser otimizado é possível obter uma diminuição do gasto energético total para qualquer modo de operação, indiferentemente do número de bytes a serem enviados.

O algoritmo otimiza os valores de ' $k \cdot \Theta$ ' para cada modo de operação, como se observa nos gráficos da Figura 54, Figura 59, Figura 64 e Figura 69, sendo o modo de operação 1 referente ao de menor gasto energético, representado pelos pontos azuis e o modo de operação 2, maior gasto energético, representado pelos pontos vermelhos. É de salientar também que no caso do cenário 'B' os valores são iguais para ambos os modos de operação e daí advém que apenas vê-se os pontos vermelhos.

De seguida multiplica esses valores ótimos pela janela de deriva, obtida na primeira parte da simulação a partir dos erros dos TIM e da NCAP, originando assim o tempo de escuta otimizado. Esse tempo é usado para calcular a energia de escuta otimizada que é usada de seguida no cálculo da energia total. É assim que se alcançam os valores de energia total otimizada.

○ **Analizando os valores obtidos na simulação 1 para o TIM 1, Figura 56, tem-se:**

Para a energia não otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((11.96E-4/91.59E-3)*100)$  aproximadamente 1.3% para o modo de operação 1 e  $((10.84E-4/91.37E-3)*100)$  aproximadamente 1.18% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação obtém-se uma relação de aproximadamente 99.8%  $((91.37E-3/91.59E-3)*100)$  e de cerca de 90.6%  $((10.84E-4/11.96E-4)*100)$  para o cenário A.

Para a energia otimizada, a relação entre a quantidade de bytes nos cenários A e B é de aproximadamente 3.03%. Entre os cenários A e B temos uma relação energética de  $((11.25E-4/91.46E-3)*100)$  aproximadamente 1.2% para o modo de operação 1 e  $((10.13E-4/91.2E-3)*100)$  e cerca de 1.1% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação temos  $((91.2E-3/91.46E-3)*100)$  aproximadamente 99.8% e para o cenário A  $((10.13E-4/11.25E-4)*100)$  aproximadamente 90%.

Quando comparadas as energias não otimizadas com as otimizadas, entre elas correspondentemente tem-se:

Tabela 12 - Relação entre as energias original e otimizada, simulação 1 - Tim 1

	Cenário	Energia sem otimização	/	Energia otimizada	=	Relação
Modo de operação 1	A	11.96E-4	/	11.25E-4	=	1.06
	B	91.59E-3	/	91.46E-3	=	1.00
Modo de operação 2	A	10.84E-4	/	10.13E-4	=	1.07
	B	91.37E-3	/	91.2E-3	=	1.00

Observa-se assim que a relação entre gasto energético, otimizado ou não, é aproximadamente igual para ambos os modos de operação, num mesmo cenário.

o **Analisando os valores obtidos na simulação 2 para o TIM 1, tem-se**

**Figura 61:**

Para a energia não otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((13.21E-4/91.78E-3)*100)$  aproximadamente 1.44% para o modo de operação 1 e  $((12.09E-4/91.56E-3)*100)$  aproximadamente 1.30% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.56E-3/91.78E-3)*100)$  aproximadamente 99.8% e para cenário A  $((12.09E-4/13.21E-4)*100)$  aproximadamente 91.5%.

Para a energia otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((11.71E-4/91.51E-3)*100)$  aproximadamente 1.28% para o modo de operação 1 e  $((10.58E-4/91.29E-3)*100)$  aproximadamente 1.16% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.29E / 91.51E-3)*100)$  aproximadamente 99.8% e para cenário A  $((10.58E-4/11.71E-4)*100)$  aproximadamente 99.8%.

Quando comparadas as energias não otimizadas com as otimizadas, entre elas correspondentemente tem-se:

Tabela 13 - Relação entre as energias original e otimizada, simulação 2 - Tim 1

	Cenário	Energia sem otimização	/	Energia otimizada	=	Relação
Modo de operação 1	A	13.21E-4	/	11.71E-4	=	1.09
	B	91.78E-3	/	91.51E-3	=	1.00
Modo de operação 2	A	12.09E-4	/	10.58E-4	=	1.11
	B	91.56E-3	/	91.29E	=	1.00

Observa-se assim que a relação entre gasto energético, otimizado ou não, é aproximadamente igual para ambos os modos de operação, num mesmo cenário.

- **Comparando as relações percentuais das duas simulações acima tem-se, aproximadamente, que:**

Tabela 14 – Comparação das relações de energia das simulações 1 e 2, entre o cenário A e B, para TIM 1 e 2 nos dois modos de operação

Simulação	Modo de operação	Tipo de energia	Relação	%
1	1	Não otimizada	$((11.96E-4/91.59E-3)*100)$	1.30
	2		$((10.84E-4/91.37E-3)*100)$	1.19
1	1	Otimizada	$((11.25E-4/91.46E-3)*100)$	1.23
	2		$((10.13E-4/91.2E-3)*100)$	1.11
2	1	Não otimizada	$((13.21E-4 / 91.78E-3)*100)$	1.44
	2		$((12.09E-4/91.56E-3)*100)$	1.32
2	1	Otimizada	$((11.71E-4/91.51E-3)*100)$	1.28
	2		$((10.58E-4/0.91.29E)*100)$	1.16

Daqui observa-se que a relação do gasto energético entre os cenários A e B varia entre 1.1% e 1.8%, ou seja, entre aproximadamente 1 e 2%.

- **Analisando os valores comparativos das energias otimizada e não para a simulação 1 e 2 tem-se:**

Pode-se observar pela Figura 57 comparada com a Figura 58 e pela Figura 62 comparada com a Figura 63 que os valores de diferença energética são iguais para ambos os cenários, como é possível ver na tabela abaixo.

Tabela 15 – Relação das diferenças energéticas entre os modos de operação para ambos os TIM, simulação 1 e 2

Simulação	TIM	Modo op.	Diferença energética (cenários A e B)	Cenário	Energia original	Relação	%
1	1	1	12.78E-5	A	11.96E-4	$(12.78E-5/11.96E-4)*100$	10.68
				B	91.59E-3	$(12.78E-5/91.59E-3)*100$	0.14
		2	71.4E-5	A	10.84E-4	$(71.4E-5/10.84E-4)*100$	<b>6.59</b>
				B	91.37E-3	$(71.4E-5/91.37E-3)*100$	0.08
	2	1	18.84E-5	A	12.50 E-4	$(18.84E-5/12.50 E-4)*100$	15.07
				B	91.67E-3	$(18.84E-5/91.67E-3)*100$	0.20
		2	10.52E-5	A	11.37 E-4	$(10.52E-5/11.37 E-4)*100$	9.25
				B	91.45E-3	$(10.52E-5/91.45E-3)*100$	0.11
2	1	1	26.91E-5	A	13.21E-4	$(26.91E-5/13.21E-4)*100$	<b>20.37</b>
				B	91.78E-3	$(26.91E-5/91.78E-3)*100$	<b>0.29</b>
		2	15.03E-5	A	12.09E-4	$(15.03E-5/12.09E-4)*100$	12.44
				B	91.56E-3	$(15.03E-5/91.56E-3)*100$	0.16
	2	1	17.49E-5	A	12.38E-4	$(17.49E-5/12.38E-4)*100$	14.13
				B	91.65E-3	$(17.49E-5/91.65E-3)*100$	0.20
		2	9.77E-5	A	11.26E-4	$(9.77E-5/11.26E-4)*100$	8.68
				B	91.43E-3	$(9.77E-5/91.43E-3)*100$	<b>0.01</b>

Daqui vê-se que para o cenário A (3 bytes) a percentagem da redução energética anda na casa das unidades, entre 6.58% e 20.37%, enquanto, para o cenário B (99 bytes), situa-se na casa das dezenas, entre 0.011% e 0.293%. Olhando a tabela anterior para apenas a um TIM:

Tabela 16 - Comparação percentual para um mesmo cenário em modos de operação diferentes

Simulação	TIM	Modo op.	Diferença energética a (Cenário A e B)	Cenário	Energia original	Relação	%	Comparação para o mesmo
1	1	1	12.78E-5	A	11.964E-4	$(12.78E-5/11.964E-4)*100$	<b>10.68</b>	>
				B	91.5883E-3	$(12.78E-5/91.5883E-3)*100$	<b>0.14</b>	>

	2	71.4E-5	A	10.839E-4	$(71.4E-5/10.839E-4)*100$	6.5 9	<
			B	91.370E-3	$(71.4E-5/91.370E-3)*100$	0.0 8	<
2	1	26.91E-5	A	13.211E-4	$(26.91E-5/13.211E-4)*100$	<b>20.</b> <b>37</b>	>
			B	91.775E-3	$(26.91E-5/91.775E-3)*100$	<b>0.2</b> <b>9</b>	>
	2	15.03E-5	A	12.086E-4	$(15.03E-5/12.086E-4)*100$	12. 44	<
			B	91.557E-3	$(15.03E-5/91.557E-3)*100$	0.1 6	<

\*continuação da Tabela 16.

Por fim da análise das figuras do ponto anterior é possível afirmar que houve uma redução do gasto energético, por otimização de Kteta, constante independentemente do cenário usado. Ou seja, o valor nominal de otimização é constante para ambos os cenários sendo que obviamente tem mais influência no cenário A pois é o que já inicia com a vantagem de possuir menos quantidade de dados a enviar.

Tendo como base a Figura 46, partindo do conhecimento adquirido, tem-se que o melhor cenário, dar-se-ia para o cenário A (3 bytes) conjugado com o modo de operação de menor gasto, portanto o modo de operação 2, e que, o pior dar-se-ia para o cenário B (99 bytes) usando o modo de operação de maior gasto energético. Portanto conjugando isso com a tabela anterior percebe-se que entre os cenários A, de cada simulação, o que têm uma maior percentagem de otimização energética é o envio com o modo de operação 1, e entre os cenários B o que tem a maior percentagem de otimização é o que usa também o modo de operação 1. Resumindo, o modo de operação 1, que constitui o maior gasto energético é o que sofre uma maior otimização. O mesmo se verifica na simulação 2.

Sendo então obvio que o contrário também se observa, para o modo de operação 2 as percentagens de diminuição do gasto energético são inferiores quando comparadas entre um mesmo cenário de quantidade de bytes. Não significa isto que estes são as combinações mais vantajosas pois, da observação das energias pós otimização presentes nas Figura 56 e Figura 61, vê-se que continua a ser válido que, mesmo após otimização, apesar de sofrerem de uma menor percentagem desta, o modo de operação mais vantajoso é o 2 pois além de iniciar com menor gasto ainda sofre uma posterior otimização, mantendo o seu gasto inferior ao modo de operação 1. Portanto confirma-se que a Figura 46 descreve bem a categorização energética dos cenários.

○ **Analisando os valores obtidos na simulação 3 para o TIM 2, Figura 66, tem-se:**

Para a energia não otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((11.43E-4/91.51E-3)*100)$  aproximadamente 1.25% para o modo de operação 1 e  $((10.30E-4/91.29E-3)*100)$  aproximadamente 1.13% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.29E-3/91.51E-3)*100)$  aproximadamente 99.76% e para cenário A  $((10.30E-4/11.43E-4)*100)$  aproximadamente 90.16%.

Para a energia otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((11.05E-4/91.44E-3)*100)$  aproximadamente 1.21% para o modo de operação 1 e  $((9.93E-4/91.22E-3)*100)$  aproximadamente 1.08% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.22E-3/91.44E-3)*100)$  aproximadamente 99.75% e para o cenário A  $((9.93E-4/11.05E-4)*100)$  aproximadamente 89.86%.

Observa-se assim que a relação entre gasto energético, otimizado ou não, é aproximadamente igual para ambos os modos de operação, num mesmo cenário.

○ **Analisando os valores obtidos na simulação 4 para o TIM 2, Figura 71, tem-se:**

Para a energia não otimizada, a relação entre a quantidade de bytes do cenário A e B é de aproximadamente 3.03%, entre o cenário A e B tem-se uma relação energética de  $((11.90E-4/91.58E-3)*100)$  aproximadamente 1.29% para o modo de operação 1 e  $((10.78E-4/91.36E-3)*100)$  aproximadamente 1.17% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.36E-3/91.58E-3)*100)$  aproximadamente 99.76% e para cenário A  $((10.78E-4/11.90E-4)*100)$  aproximadamente 90.5%.

Para a energia otimizada, a relação entre a quantidade de bytes do cenário B e A é de aproximadamente 3.03%, entre o cenário B e A tem-se uma relação energética de  $((11.23E-4/91.46E-3)*100)$  aproximadamente 1.22% para o modo de operação 1 e  $((10.10E-4/91.24E-3)*100)$  aproximadamente 1.11% para o modo de operação 2. Comparando o cenário B para os diferentes modos de operação tem-se  $((91.24E-3/91.46E-3)*100)$  aproximadamente 99.8% e para cenário A  $((10.10E-4/11.23E-4)*100)$  aproximadamente 89.9%.

Observa-se assim que a relação entre gasto energético, otimizado ou não, é aproximadamente igual para ambos os modos de operação, num mesmo cenário e que a relação do gasto energético entre os cenários A e B varia entre 1.11% e 1.29%, ou seja, entre aproximadamente 1 e 1.3%.

○ **Analisando os valores comparativos das energias otimizada e não para a simulação 3 e 4 tem-se:**

Pode-se observar pela Figura 67 comparada com a Figura 68 e pela Figura 72 comparada com a Figura 73 que os valores de diferença energética são iguais para ambos os cenários, como é possível ver na tabela abaixo.

Tabela 17 – Relação das diferenças energéticas entre os modos de operação para ambos os TIM

Simulação	TIM	Modo de op.	Diferença energética para ambos os cenários	Cenário	Energia original	Relação	%
3	1	1	18.84E-5	A	12.50E-4	$(18.84E-5/12.499E-4)*100$	15.07
				B	91.668E-3	$(18.84E-5/91.668E-3)*100$	0.20
	2	2	10.52E-5	A	11.373E-4	$(10.52E-5/11.373E-4)*100$	9.25
				B	91.450E-3	$(10.52E-5/91.450E-3)*100$	0.11
	2	1	6.73E-5	A	11.429E-4	$(6.73E-5/11.429E-4)*100$	5.89
				B	91.508E-3	$(6.73E-5/91.508E-3)*100$	0.07
2	2	3.76E-5	A	10.304E-4	$(3.76E-5/10.304E-4)*100$	3.65	
			B	91.290E-3	$(3.76E-5/91.290E-3)*100$	0.04	
4	1	1	14.13E-5	A	12.083E-4	$(14.13E-5/12.083E-4)*100$	11.69
				B	91.606E-3	$(14.13E-5/91.606E-3)*100$	0.15
	2	2	7.89E-5	A	10.958E-4	$(7.89E-5/10.958E-4)*100$	7.20
				B	91.388E-3	$(7.89E-5/91.388E-3)*100$	0.09
	2	1	12.11E-5	A	11.905E-4	$(12.11E-5/11.905E-4)*100$	10.17
				B	91.579E-3	$(12.11E-5/91.579E-3)*100$	0.13
2	2	6.76E-5	A	10.779E-4	$(6.76E-5/10.779E-4)*100$	6.27	
			B	91.361E-3	$(6.76E-5/91.361E-3)*100$	0.07	

Olhando a tabela anterior para apenas a um TIM:

Tabela 18 - Comparação percentual para um mesmo cenário em modos de operação diferentes para TIM

Simulação	TIM	Modo de	Diferença energética (ambos os cenários)	Cenário	Energia original	Relação	%	Comparação para o mesmo
3	2	1	6.73E-5	A	11.43E-4	$(6.73E-5/11.43E-4)*100$	<b>5.89</b>	>

4		2	3.76E-5	B	91.51E-3	$(6.73E-5/91.51E-3)*100$	0.07	<
				A	10.30E-4	$(3.76E-5/10.30E-4)*100$	3.65	<
		1	12.11E-5	B	91.29E-3	$(3.76E-5/91.29E-3)*100$	<b>0.04</b>	>
				A	11.90E-4	$(12.11E-5/11.90E-4)*100$	<b>10.17</b>	>
		2	6.76E-5	B	91.58E-3	$(12.11E-5/91.58E-3)*100$	0.13	<
				A	10.78E-4	$(6.76E-5/10.78E-4)*100$	6.27	<
B	91.36E-3	$(6.76E-5/91.36E-3)*100$	<b>0.07</b>	>				

\*continuação da Tabela 18.

Por fim da análise das figuras do ponto anterior é possível afirmar que houve uma redução do gasto energético, por otimização de Kteta, constante independentemente do cenário usado. Ou seja, o valor nominal de otimização é constante para ambos os cenários sendo que obviamente tem mais influência no cenário A pois é o que já inicia com a vantagem de possuir menos quantidade de dados a enviar.

Tendo como base a Figura 46, partindo do conhecimento adquirido, tem-se que o melhor cenário, dar-se-ia para o cenário A (3 bytes) conjugado com o modo de operação de menor gasto, portanto o modo de operação 2, e que, o pior dar-se-ia para o cenário B (99 bytes) usando o modo de operação de maior gasto energético. Portanto conjugando isso com a tabela anterior percebe-se que entre os cenários A, de cada simulação, o que têm uma maior percentagem de otimização energética é o envio com o modo de operação 1, e entre os cenários B o que tem a maior percentagem de otimização é o que usa também o modo de operação 1. Resumindo, o modo de operação 1, que constitui o maior gasto energético é o que sofre uma maior otimização. O mesmo se verifica na simulação 2.

Sendo então obvio que o contrário também se observa, para o modo de operação 2 as percentagens de diminuição do gasto energético são inferiores quando comparadas entre um mesmo cenário de quantidade de bytes. Não significa isto que estes são as combinações mais vantajosas pois, da observação das energias pós otimização presentes nas Figura 67 e Figura 72, vê-se que continua a ser válido que, mesmo após otimização, apesar de sofrerem de uma menor percentagem desta, o modo de operação mais vantajoso é o 2 pois além de iniciar com menor gasto ainda sofre uma posterior otimização, mantendo o seu gasto inferior ao modo de operação 1. Portanto, novamente, confirma-se que a Figura 46 descreve bem a categorização energética dos cenários.

**Folha em branco**

## Capítulo 5 – Conclusão

Pode-se observar pela análise de resultados que o algoritmo de facto otimizou o parâmetro pretendido alcançando assim uma redução da energia gasta nas comunicações. Quando aliado a uma boa estratégia de agendamento (*scheduling*) das sincronizações de relógio, pode ser uma boa forma de diminuir os gastos dos sensores *low-power* e assim aumentar a sua vida útil e do seu método de armazenamento, diminuindo assim a necessidade de manutenção decorrente das suas baterias limitadas.

Consegue-se também perceber que a escolha do modo de operação de um aparelho também tem influência nos seus gastos e que uma boa gestão dessa escolha pode diminuí-los. Pela análise dos valores obtidos percebe-se que apesar da otimização ser maior, em percentagem, quando o modo de operação é o que usa maior gasto energético, o mais vantajoso e com valores mais baixos é o modo com menor gasto. Assim pode-se confirmar que o melhor cenário geral possível é usando a otimização para o modo de menor gasto (modo de operação 2) para o cenário de 3 bytes (cenário A) e o pior cenário geral é usando o modo de operação pior (modo de operação 1) para envio de 99 bytes (cenário B). entres estes dois modos tem-se ainda o envio de 3 bytes com o modo de operação 1 que é menos vantajoso que o melhor cenário geral e tem-se o envio de 99 bytes com o modo de operação 2 também este mais vantajoso que o uso do modo de operação 1, todos isto se aplica tanto antes como depois da otimização.

Portanto, se falarmos da melhor forma de gerir o envio de 3 e 99 bytes, seria usando o modo de operação 2 otimizado para ambos, obtendo assim um gasto energético mais reduzido, não podendo, por algum motivo, fazer desta forma, continua a ser útil usar outro modo otimizado de forma a reduzir o máximo possível os gastos.

Conclui-se assim que o uso de um algoritmo de otimização é na maioria dos casos, vantajoso pois permite gerir de uma melhor forma a energia e evitar que esta seja gasta em situações desnecessárias e ainda que os modos de operação que usufruem de um maior gasto energético sofrem uma otimização maior, mas os que possuem menor gasto energético antes e após otimização são os casos que usam o modo de operação com menor gasto.

### 5.1. Perspetivas futuras

Por fim aponto como possíveis trabalhos futuros a aplicação prática deste modelo bem como o aumento do seu grau de complexidade, adicionando por exemplo parâmetros variáveis de tarefas, modos de operação, ou até, a aplicação do modelo a sistema exemplo “completo” com um esquema de comunicações/tarefas para vários TIM, com várias características diferentes, e aí, avaliar os gastos energéticos no cenário todo, desde a sincronização, passando pela recolha energética até à otimização. Ou seja, a investigação e aperfeiçoamento contínuo do algoritmo, de

forma a torná-lo ainda mais eficiente e adaptável, aliando a isto uma adaptação a vários/diferentes modos de operação para fazer observações comparativas. Integração do algoritmo otimizado com tecnologias atualizadas, como protocolos de comunicação de próxima geração ou novos padrões de hardware, para garantir a compatibilidade e eficácia contínuas na sua utilização. Pode ainda ser avaliada a sua aplicabilidade a redes de maior escala, com diferentes topologias, e a adaptação de estratégias de sincronização, explorando outros métodos ou outras adaptações a contextos específicos. E por fim o possível desenvolvimento de ferramentas facilitadas e interfaces/ aplicações para utilizadores comuns, de forma a tornar a implementação e configuração do algoritmo em diferentes situações.

## Bibliografia

- [1] L. T. Bruscato, “PROPOSTA DE MÉTODOS DE SINCRONIZAÇÃO DE REDE DE SENSORES SEM FIO,” Porto Alegre, 2017.
- [2] I. Teixeira, “ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA PARA REDES DE SENSORES SEM FIO,” Rio de Janeiro, Apr. 2005.
- [3] M. D. Depexe, “CONCEPÇÃO DE UM CIRCUITO ENERGY HARVESTING APLICADO A REDES DE SENSORES SEM FIO PARA SISTEMAS DE ILUMINAÇÃO,” 2014.
- [4] D. da C. Pinho, “Bio-inspired energy harvesting and optimization Otimização e obtenção de energia de forma bio inspirada,” 2021.
- [5] L. Liu, Y. Song, H. Zhang, H. Ma, and A. V. Vasilakos, “Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks,” *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 819–832, 2015, doi: 10.1109/TC.2013.229.
- [6] H. A. Illias, N. S. Ishak, H. Mokhlis, and M. Z. Hossain, “IoT-based hybrid renewable energy harvesting system from water flow,” *PECon 2020 - 2020 IEEE International Conference on Power and Energy*, pp. 204–208, 2020, doi: 10.1109/PECon48942.2020.9314412.
- [7] B. V. FERREIRA, E. J. P. PEDRO, P. H. N. ALVES, T. A. de SOUZA, W. R. VARGAS, and L. G. da S. COSTA, “Conversão de energia através de Energy Harvesting,” *Centro de Ensino Superior de Juiz de Fora – CES/JF*, no. Centro de Ensino Superior de Juiz de Fora – CES/JF Curso de Engenharia Elétrica Trabalho de iniciação científica – Artigo.
- [8] Y. Da and A. Khaligh, “Hybrid offshore wind and tidal turbine energy harvesting system with independently controlled rectifiers,” *IECON Proceedings (Industrial Electronics Conference)*, pp. 4577–4582, 2009, doi: 10.1109/IECON.2009.5414866.
- [9] W. S. dos Santos, “DESENVOLVIMENTO DE DISPOSITIVOS FOTOELETROQUÍMICOS À BASE DE  $\text{BiVO}_4/\text{Bi}_4\text{V}_2\text{O}_{11}$  PARA CONVERSÃO DE ENERGIA SOLAR EM ENERGIA ELÉTRICA OU ENERGIA QUÍMICA A PARTIR DE RESÍDUOS LÍQUIDOS INDUSTRIAIS,” 2017.
- [10] U. Lucia, “Overview on fuel cells,” *Renewable and Sustainable Energy Reviews*, vol. 30. Elsevier Ltd, pp. 164–169, 2014. doi: 10.1016/j.rser.2013.09.025.

- [11] J. D. B. de Araújo, “Protótipo de Rastreador Solar de um eixo baseado em microcontrolador,” Dec. 2015.
- [12] V. M. R. Viegas, “PLATAFORMAS DE INTEROPERABILIDADE PARA SISTEMAS DE INSTRUMENTAÇÃO, MEDIDA E CONTROLO,” Lisboa, 2012. [Online]. Available: <https://www.researchgate.net/publication/260982564>
- [13] S. R. Rossi, “IMPLEMENTAÇÃO DE UM NÓ IEEE 1451, BASEADO EM FERRAMENTAS ABERTAS E PADRONIZADAS, PARA APLICAÇÕES EM AMBIENTES DE INSTRUMENTAÇÃO DISTRIBUÍDA,” São Paulo, Jan. 2005.
- [14] A. K. Kar, “Bio inspired computing - A review of algorithms and scope of applications,” *Expert Syst Appl*, vol. 59, pp. 20–32, 2016, doi: 10.1016/j.eswa.2016.04.018.
- [15] A. Kozłowski and J. Sosnowski, “Energy Efficiency Trade-Off Between Duty-Cycling and Wake-Up Radio Techniques in IoT Networks,” *Wirel Pers Commun*, vol. 107, no. 4, pp. 1951–1971, Aug. 2019, doi: 10.1007/s11277-019-06368-0.
- [16] D. da C. Pinho, “Elaboração de um algoritmo híbrido para aplicação de MPPT,” 2022.
- [17] “CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver Applications.”
- [18] H. Da Rocha, J. L. D. Pereira, T. A. G. N. Rodrigues, J. A. Salvado, and A. Espirito-Santo, “An Energy-Efficient Process for Optimal Communication Synchronization in Low Power Wireless Smart Sensors,” in *2022 IEEE International Symposium on Measurements and Networking, M and N 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/MN55117.2022.9887722.

## Anexos

- **Código MATLAB:**

```

%% cálculo do erro associado a cada relógio de cada tim -> resulta a
janela de deriva temporal e a mesma multiplicada pelo coeficiente k
%% Início
clear all
clc
close all

%% Parâmetros
format longE; % Formato para números longos (engenharia)
nTIM = 2; % Número de TIM'S
freqTIM = randi([1,9],1,nTIM); % Frequência de relógio das tim's (MHz)
fclkTIM = freqTIM*1e6; % Converter em ciclos de relógio/s, MHz-
>Hz(/s)=ciclos de relógio é *1e6
freqNCAP = 10; % Frequência de relógio da NCAP (MHz)
fclkNCAP = freqNCAP*1e6; % Conversão para ciclos de relógio por
segundo

%% Valor do desvio TETA e K
desvio = randi([10,50],1,nTIM); % Entre 10 e 50 ppm, nº aleatório para cada
tim
Teta = desvio*1e-6; % Partes por milhão dividido por 1milhão
para posteriormente obtermos o desvio multiplicando pela frequência

%% Gerar valores comparativos
for i=1:nTIM %para todas as TIM's
    %considerando (i,j) sendo j=1
    PeriodoTIM(i) = 1./fclkTIM(i); % Período da tim com freq nominal,
divisão ponto a ponto (em segundos)
    PeriodoNCAP = 1/fclkNCAP; % Período da NCAP (em segundos)

    % Frequência de relógio para cada tim, valor absoluto
    fdesvio(i) = fclkTIM(i) * (1 - Teta(i)); % f para cada TIM com desvio

    % Periodos max e min para cada tim, com desvio, valor absoluto
    PeriodoTIMdesv(i) = 1./fdesvio(i); % Período frequência com desvio
end

%% Gerar tarefas
TempoTarefa=[600,900]; % Matriz tempo de tarefa em segundos, para
comparar o desfazamento entre o master e a tim a realizar a mesma tarefa

for i = 1:nTIM

    for j = 1:length(TempoTarefa) % Até ao tamanho da TempoTarefa para
calcular para todos os valores

        % Número de ciclos por tarefa
        NciclosNCAP(i,j) = TempoTarefa(j)/PeriodoNCAP; % Ciclos
de relógio que a NCAP realiza em x s da tarefa
        NciclosTIM(i,j) = TempoTarefa(j)/PeriodoTIM(i); % Ciclos
de relógio que a TIM realiza em x s da tarefa
    end
end

```

```

    NciclosTIMdesv(i,j) = TempoTarefa(j)/PeriodoTIMdesv(i); % Ciclos
de relógio das tim's com desvio , valor absoluto, para realização das tarefas

    % Discrepância de relógio para uma mesma tarefa em numero de ciclos
em relação à ftim original que por sua vez será igual à discrepância em
relação à NCAP
    difNciclos(i,j) = NciclosTIM(i,j) - NciclosTIMdesv(i,j); % Diferença
entre as TIM com e sem desvio (absoluto)

    % Discrepância de relógio para a mesma tarefa em TEMPO (segundos) em
relação à ftim original que por sua vez será igual à discrepância em relação
à NCAP
    difTempo(i,j) = PeriodoTIM(i)*abs(difNciclos(i,j));

    %% usar o k como coeficiente dos valores
    % o difTempo representa a variação temporal em valor absoluto logo a
janela
    % temporal será o dobro deste valor
    % para cada tim corresponde um valor de JanelacomK0 para cada tarefa

    for k=3:-1:1

        JanelaTemporal(i,j,k) = 2*difTempo(i,j);

        JanelacomK0(i,j,k) = k*JanelaTemporal(i,j,k); %% K*0 sem
otimização(drift do relógio) j corresponde à tarefa!!!! 600 e 900

        %% paara ficar na ordem pior, melhor para usar no algoritmo
        JanelaTemporalinvertida = flip(JanelaTemporal, 2);
        JanelacomK0invertida = flip(JanelacomK0, 2);
    end
end
end

fclkTIM
fclkNCAP
NciclosNCAP
NciclosTIM
difNciclos
difTempo
JanelaTemporal
JanelacomK0

%% otimização
%%parâmetros
% Tensao de alimentação
V=3.3; % (volt)

%% Algoritmo PSO de otimização
% dimensao
dim = 2;
% limites
% lb vai ser no minimo k min vezes erro min 3*(10*1e-6) 10ppm
lb = [0 0];
% ub vai ser no máximo o k maximo* erro max 3*(50*1e-6) 50ppm
ub = [0.245 0.245];

```

```

% %parametros necessários
Tcom = 300; %segundos
%para o [PIOR CENÁRIO de 99 bytes, MELHOR de 3 bytes] para RT 250kbaud
Tdados= [0.0031, 0.0009]; %segundos pior,melhor
k=[3 2 1];

%numero de iterações
nit=0;
nitMax=20;

%% iniciar população aleatoriamente e velocidades das partículas
%% posição inicial em vez de ser randomizada são a janela de deriva a
multiplicar k 3 2 e 1

for i = 1:nTIM
    for k = 3:-1:1

        % k_Teta(i,j,p)= k(i)*Teta(j); % substituir teta pela diferença
temporal
        k_Teta(i,k)=JanelacomK0invertida(i,1,k);

        Tesc(i,k) = k_Teta(i,k);
    end
end

%consumos de corrente [PIOR,MELHOR]
Isleep = [92e-6,900e-9];    %(amperes) modo sleep
Iesc = [1.5e-3,1.5e-3];    %(amperes) modo idle
Irx = [18.8e-3,13.3e-3];   %(amperes) modo recessao
Itx = [21.5e-3,11.1e-3];   %(amperes) modo transmissão

% tempos de comunicação
Tsleep = 300; % vai depender do tempo necessário para recolher a energia
necessária para a realização da proxima tarefa
TescN_otim = JanelacomK0invertida; %sem otimização tem de ser pior,melhor (por
isso foi invertido la em cima)
TescParaOtim = JanelaTemporalinvertida;
Ttx = [0.003456,0.00384]; %(segundos) pior, melhor
Trx = 0.00144;            %(segundos)

% latências
%Ritmo de transmissão 250000 bits/s então 1 bit demora 4*10-6s calculando
%o tempo para cada latencia: RX-9bit Tx-8bit
LatRx = 3.6e-5;%segundos 9bits * 4*10-6s
LatTx = 3.2e-5;%segundos 8 bits * 4*10-6s

% calculo potencias [pior,melhor]
for i=1:nTIM
    for mop=1:2
        Psleep(i,mop) = (Isleep(1,mop).*V); %(W)
        Pesc(i,mop) = (Iesc(1,mop).*V); %(W)
        Prx(i,mop) = (Irx(1,mop).*V); %(W)
        Ptx(i,mop) = (Itx(1,mop).*V); %(W)

        Ptotal(i,mop) = Psleep(i,mop) + Pesc(i,mop) + Prx(i,mop) +
Ptx(i,mop); %(W)
    end
end

```

```

for i=1:nTIM
    for mop=1:2
        for k=3:-1:1
            for t =1:length(Ttx)
                % Cálculo das energias [PIOR,MELHOR] [J]
                Esleep(i,mop) = Psleep(i,mop) .* Tsleep;
                EescN_otim(i,k,mop) = Pesc(i,mop) .* TescN_otim(i,mop); %
sem otimização
                Erx(i,mop) = Prx(i,mop) * (Trx + LatRx);
                Etx(i,mop,t) = Ptx(i,mop) * (Ttx(1,t) + LatTx); %
(pior,melhor)
            end
        end
    end
end
for i=1:nTIM
    for mop=1:2
        for k=3:-1:1
            for t =1:length(Ttx)
                Etot(i,k,mop,t) = Esleep(i,mop) + EescN_otim(i,k,mop) +
Erx(i,mop) + Etx(i,mop,t) ;
            end
        end
    end
end

%% posições melhores
%melhor global apenas informativo
Gbest= -inf;
xGbest= zeros(1,nTIM);

%melhor particular
Ktetabest=zeros(nTIM,3,2,2)+inf;
xKTetaBest= zeros(3,nTIM,2,2);

%% parametros plots %%
% Inicializar a matriz para armazenar a evolução de Pbest
evolucao_Pbest = zeros(nitMax, nTIM,2);

%% função objetivo

while nit<=nitMax

    nit=nit+1;

    for i=1:nTIM % quantidade de TIM
        for k = 3:-1:1 %k[-3,-2,-1,1,2,3]
            for mop=1:2 %modo operação
                for t = 1:length(Ttx) %tamanho dos dados 3 ou 99 bytes

                    %Pesc em ordem a kteta
                    fobj(i,k,mop,t) = (Tcom * Tesc(i,k) * Pesc(i,mop)) /
Ttx(1,t);

                    if fobj(i,k,mop,t) < Gbest

                        Gbest=fobj(i,k,mop,t);
                        xGbest=Tesc(i,k);

```

```

end

if fobj(i,k,mop,t) < Ktetabest(i,1,mop,t)

    Ktetabest(i,1,mop,t)=fobj(i,k,mop,t);
    xKTetaBest(i,1,mop,t)= Tesc(i,k);
end

%% verificação dos limites
%%verificar se Tesc está dentro dos limites

if Ktetabest(i,k) > ub(1,i)
    Ktetabest(i,k)=ub(1,i);
end
if Ktetabest(i,k)<lb(1,i)
    Ktetabest(i,k)=lb(1,i);
end

i
k
mop
t

end
end
end
end

end
%% calculo valores otimizados
for i =1:nTIM % quantidade de TIM
    for mop=1:2 %modo operação
        for t=1:length(Ttx) %tamanho dos dados 3 ou 99 bytes
            for k=3:-1:1
                TescOtim(i,k,mop) = Ktetabest(i,k)*TescParaOtim(i,mop);
                Eesc(i,k,mop) = Pesc(1,mop) * TescOtim(i,k); %otimizada
                Etotalotimizada(i,k,mop,t) = Esleep(i,mop) + Eesc(i,k,mop) +
                Erx(i,mop) + Etx(i,mop,t);
                % diferença entre energia "original" e a energia otimizada
                DifEnergetica(i,k,mop,t) = Etotal(i,k,mop,t) -
                Etotalotimizada(i,k,mop,t);
            end
        end
    end
end

end

%% PLOTS
%% fazer plot das energia otimizada e por otimizar num mesmo grafico para
comparar
figure('Name', 'Energia total e energia total otimizada');
% Energias totais não otimizadas para TIM 1
subplot(2,2,1);
dadosBarras = [Ettotal(1,1,1,1), Ettotal(1,1,2,1), Ettotal(1,1,1,2),
Ettotal(1,1,2,2)];
bar(dadosBarras, 'FaceColor', [0.8, 0.8, 0.8]);
xlabel('TIM 1');

```

```

ylabel('Energia [J]');
title('Energias Totais Não Otimizadas - TIM 1');
set(gca, 'XTickLabel', {'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3
Bytes-mop2'});

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:length(dadosBarras)
    text(i, dadosBarras(i), sprintf('%.7f', dadosBarras(i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end

% Energias totais otimizadas para TIM 1
subplot(2,2,2);
dadosBarras = [Etotalotimizada(1,1,1,1), Etotalotimizada(1,1,2,1),
Etotalotimizada(1,1,1,2), Etotalotimizada(1,1,2,2)];
bar(dadosBarras, 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('TIM 1');
ylabel('Energia [J]');
title('Energias Totais Otimizadas - TIM 1');
set(gca, 'XTickLabel', {'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3
Bytes-mop2'});

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:length(dadosBarras)
    text(i, dadosBarras(i), sprintf('%.7f', dadosBarras(i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end

% Energias totais não otimizadas para TIM 2
subplot(2,2,3);
dadosBarras = [Etotal(2,1,1,1), Etotal(2,1,2,1), Etotal(2,1,1,2),
Etotal(2,1,2,2)];
bar(dadosBarras, 'FaceColor', [0.8, 0.8, 0.8]);
xlabel('TIM 2');
ylabel('Energia [J]');
title('Energias Totais Não Otimizadas - TIM 2');
set(gca, 'XTickLabel', {'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3
Bytes-mop2'});

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:length(dadosBarras)
    text(i, dadosBarras(i), sprintf('%.7f', dadosBarras(i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end

% Energias totais otimizadas para TIM 2
subplot(2,2,4);
dadosBarras = [Etotalotimizada(2,1,1,1), Etotalotimizada(2,1,2,1),
Etotalotimizada(2,1,1,2), Etotalotimizada(2,1,2,2)];
bar(dadosBarras, 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('TIM 2');
ylabel('Energia [J]');
title('Energias Totais Otimizadas - TIM 2');
set(gca, 'XTickLabel', {'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3
Bytes-mop2'});

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:length(dadosBarras)

```

```

    text(i, dadosBarras(i), sprintf('%.7f', dadosBarras(i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end

%% comparação das energias
figure('Name', 'Comparação das energias');
% Energias
subplot(2,1,1);
bar([1,2 ,3,4], [Etotal(1,1,1,1), Etotal(1,1,2,1), Etotal(1,1,1,2),
Etotal(1,1,2,2)], 'BarWidth', 0.4);
hold on;
bar([1.2, 2.2, 3.2, 4.2], [Etotalotimizada(1,1,1,1),
Etotalotimizada(1,1,2,1),Etotalotimizada(1,1,1,2), Etotalotimizada(1,1,2,2)],
'BarWidth', 0.4);
ylabel('Energia [J]');
title('Comparação de Energias para TIM 1');
legend({'Não Otimizado', 'Otimizado'}, 'Location', 'Best');
xticks([1.1, 2.1, 3.1, 4.1]);
xticklabels({'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3 Bytes-
mop2'});

% Energias
subplot(2,1,2);
bar([1, 2 ,3 , 4], [Etotal(2,1,1,1), Etotal(2,1,2,1), Etotal(2,1,1,2),
Etotal(2,1,2,2)], 'BarWidth', 0.4);
hold on;
bar([1.2, 2.2, 3.2, 4.2], [Etotalotimizada(2,1,1,1),
Etotalotimizada(2,1,2,1), Etotalotimizada(2,1,1,2),
Etotalotimizada(2,1,2,2)], 'BarWidth', 0.4);
ylabel('Energia [J]');
title('Comparação de Energias para TIM 2');
legend({'Não Otimizado', 'Otimizado'}, 'Location', 'Best');
xticks([1.1, 2.1, 3.1, 4.1]);
xticklabels({'99 Bytes-mop1', '3 Bytes-mop1', '99 Bytes-mop2', '3 Bytes-
mop2'});
sgtitle('Comparação entre Energias Não Otimizadas e Otimizadas');

%% kteta best
figure('Name', 'Melhores valores de Tesc');
subplot(1,2,1)
hold on;
for t=1:nTIM
    scatter(1:nTIM, Ktetabest(:,1,t,1), 'filled', 'DisplayName', ['TIM'
num2str(t)]);
    xlabel('TIM');
    xticks([1 2 3 4 5]);
    xticklabels({'1', '2', '3', '4', '5'});
    ylabel('Melhor Valor de  $K_{\theta_{itbest}}$ ');
    title('Melhores Valores de  $K_{\theta_{itbest}}$  para cada TIM para 3
bytes');
end
grid on;

subplot(1,2,2)
hold on;
for t=1:nTIM
    scatter(1:nTIM, Ktetabest(:,1,t,2), 'filled', 'DisplayName', ['TIM'
num2str(t)]);

```

```

xlabel('TIM');
xticks([1 2 3 4 5]);
xticklabels({'1','2','3','4','5'});
ylabel('Melhor Valor de K{\theta}_{\itbest}');
title('Melhores Valores de K{\theta}_{\itbest} para cada TIM para 99
bytes');
end
grid on;

%% Gráfico de Potências
% Potências para cada TIM
figure('Name', 'Potências para cada TIM');
bar([1, 2], Ptotal(:,1), 'FaceColor', [0.8, 0.8, 0.8]);
hold on;
bar([1.1, 2.1], Ptotal(:,2), 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('TIM');
ylabel('Potência [W]');
title('Potências Totais para Cada TIM');
legend({'Cenário Pior', 'Cenário Melhor'}, 'Location', 'Best');
xticks(1:nTIM);

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:nTIM
    text(i, Ptotal(i,1), sprintf('%.7f', Ptotal(i,1)), 'HorizontalAlignment',
'center', 'VerticalAlignment', 'bottom');
    text(i, Ptotal(i,2), sprintf('%.7f', Ptotal(i,2)), 'HorizontalAlignment',
'center', 'VerticalAlignment', 'bottom');
end

%% Gráficos para Componentes de Potência
% Gráfico para TIM 1
figure('Name', 'Componentes da Potência para TIM 1');
dadosBarrasP1 = [Psleep(1,1), Pesc(1,1), Prx(1,1), Ptx(1,1)];
bar([1, 2, 3, 4], dadosBarrasP1, 'FaceColor', [0.8, 0.8, 0.8]);
hold on;
dadosBarrasP2 = [Psleep(1,2), Pesc(1,2), Prx(1,2), Ptx(1,2)];
bar([1.1, 2.1, 3.1, 4.1], dadosBarrasP2, 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('Componentes de Potência');
ylabel('Potência [W]');
title('Componentes de Potência para TIM 1');
legend({'Modo operação 1', 'Modo operação 2'}, 'Location', 'Best');
xticks(1:4);
xticklabels({'Sleep', 'Idle', 'Rx', 'Tx'});

for i = 1:length(dadosBarrasP1)
    for j = 1:length(dadosBarrasP2)
        text(i, dadosBarrasP1(1,i), sprintf('%.7f', dadosBarrasP1(1,i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
        text(i, dadosBarrasP1(1,i), sprintf('%.7f', dadosBarrasP1(1,i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

        text(j, dadosBarrasP2(1,j), sprintf('%.7f', dadosBarrasP2(1,j)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
        text(j, dadosBarrasP2(1,j), sprintf('%.7f', dadosBarrasP2(1,j)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
    end
end
end

```

```

% Gráfico para TIM 2
figure('Name', 'Componentes da Potência para TIM 2');
dadosBarrasP3 = [Psleep(2,1), Pesc(2,1), Prx(2,1), Ptx(2,1)];
bar([1, 2, 3, 4], dadosBarrasP3, 'FaceColor', [0.8, 0.8, 0.8]);
hold on;
dadosBarrasP3 = [Psleep(2,2), Pesc(2,2), Prx(2,2), Ptx(2,2)];
bar([1.1, 2.1, 3.1, 4.1], dadosBarrasP3, 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('Componentes de Potência');
ylabel('Potência [W]');
title('Componentes de Potência para TIM 2');
legend({'Modo operação 1', 'Modo operação 2'}, 'Location', 'Best');
xticks(1:4);
xticklabels({'Sleep', 'Idle', 'Rx', 'Tx'});
%adicionar valores nas barras
for i = 1:length(dadosBarrasP1)
    for j = 1:length(dadosBarrasP2)
        text(i, dadosBarrasP1(1,i), sprintf('%.7f', dadosBarrasP1(1,i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
        text(i, dadosBarrasP1(1,i), sprintf('%.7f', dadosBarrasP1(1,i)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

        text(j, dadosBarrasP2(1,j), sprintf('%.7f', dadosBarrasP2(1,j)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
        text(j, dadosBarrasP2(1,j), sprintf('%.7f', dadosBarrasP2(1,j)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
    end
end

% Gráfico de Diferença de Energia
% Diferença de Energia para cada TIM
figure;%('Name', 'Diferença de Energia para cada TIM');
bar([1, 2], DifEnergetica(:,1,1,1), 'FaceColor', [0.8, 0.8, 0.8]);
hold on;
bar([1.1, 2.1], DifEnergetica(:,1,2,1), 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('TIM');
ylabel('Energia [J]');
title('Diferença de Energia para Cada TIM - 99bytes');
legend({'Modo de operação 1', 'Modo de operação 2'}, 'Location', 'Best');
xticks(1:nTIM);

% Adicionar rótulos diretamente nas barras com 5 casas decimais
for i = 1:nTIM
    text(i, DifEnergetica(i,1,1,1), sprintf('%.7f', DifEnergetica(i,1,1,1)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
    text(i, DifEnergetica(i,1,2,1), sprintf('%.7f', DifEnergetica(i,1,2,1)),
'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end

% Diferença de Energia para cada TIM (mop1, t2)
figure;
bar([1, 2], DifEnergetica(:,1,1,2), 'FaceColor', [0.8, 0.8, 0.8]);
hold on;
bar([1.1, 2.1], DifEnergetica(:,1,2,2), 'FaceColor', [0.6, 0.6, 0.6]);
xlabel('TIM');
ylabel('Energia [J]');
title('Diferença de Energia para Cada TIM - 3 bytes');
legend({'Modo de operação 1', 'Modo de operação 2'}, 'Location', 'Best');
xticks(1:nTIM);

```

```
% Adicionar rótulos diretamente nas barras com 5 casas decimais

for i = 1:nTIM
    text(i, DifEnergetica(i,1,1,2), sprintf('%.7f', DifEnergetica(i,1,1,2)),
        'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
    text(i, DifEnergetica(i,1,2,2), sprintf('%.7f', DifEnergetica(i,1,2,2)),
        'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
end
```