

Jailson Pereira de Carvalho

**SISTEMA INTELIGENTE DE CONTROLO DA TEMPERATURA
DA ÁGUA DE PISCINA UTILIZANDO ENERGIAS RENOVÁVEIS**



UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia
2024

Jailson Pereira de Carvalho

**SISTEMA INTELIGENTE DE CONTROLO DA TEMPERATURA
DA ÁGUA DE PISCINA UTILIZANDO ENERGIAS RENOVÁVEIS**

**Mestrado em Engenharia Eletrotécnica e Computadores
(Especialidade em Tecnologias da Informação e Telecomunicações)**

Trabalho realizado sob a orientação de:

Professor Doutor Cristiano Cabrita

Professor Doutor Jânio Monteiro



UNIVERSIDADE DO ALGARVE

Instituto Superior de Engenharia

2024

SISTEMA INTELIGENTE DE CONTROLO DA TEMPERATURA DA ÁGUA DE PISCINA UTILIZANDO ENERGIAS RENOVÁVEIS

Declaração de autoria da obra

Declaro ser o autor desta obra, que é original e inédita. Os autores e obras consultados são devidamente citados no texto e constam da listagem de referências incluídas.

(Jailson Pereira de Carvalho)

©2024, Jailson Pereira de Carvalho

A Universidade do Algarve reserva para si o direito, em conformidade com o disposto no Código do Direito de Autor e dos Direitos Conexos, de arquivar, reproduzir e publicar a obra, independentemente do meio utilizado, bem como de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição para fins meramente educacionais ou de investigação e não comerciais, conquanto seja dado o devido crédito ao autor e editor respetivos.

Dedicado aos meus pais.

AGRADECIMENTOS

A Deus por me ter dado saúde e força para superar as dificuldades. A esta universidade, seu corpo docente, direção e administração que me abriram a janela que hoje vislumbro um horizonte superior, enviado pela centrada confiança no mérito e ética aqui presente.

Agradeço aos meus orientadores Professor Doutor Cristiano Lourenço Cabrita e Professor Doutor Jânio Miguel Evangelista Ferreira Monteiro, todo o apoio, partilha do saber, críticas, ideias, revisão deste relatório e a grande disponibilidade em me orientar ao longo desta dissertação.

Aos meus colegas do laboratório 160 por todas as trocas de ideias e suporte durante essa etapa da minha vida. Todos estes contributos foram essenciais para o enriquecimento e realização deste trabalho.

Aos meus pais, pelo amor, incentivo e apoio incondicional, e a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

Desde há alguns anos atrás e atualmente muito se fala das alterações climáticas como um fator que põe em risco a qualidade de vida das pessoas. Uma das causas que tem provocado essas mudanças é o uso da energia produzidas por fontes não renováveis provocando a emissões de gases do efeito de estufa.

Ao investir em geração proveniente de energias renováveis está-se em simultâneo a proteger a qualidade de vida das pessoas, a evitar as emissões de gases de efeito de estufa e a reduzir custos financeiros, por representarem fontes de energia mais baratas em comparação com os combustíveis fósseis.

O presente trabalho teve como finalidade através do uso de fontes de energia renováveis desenvolver um algoritmo com o objetivo de coordenar o funcionamento de um conjunto de equipamentos de geração de energia térmica e de armazenamento térmico, para ajustar a temperatura da água de piscinas exteriores, de acordo com as necessidades do utilizador. Para tal, o algoritmo utiliza informações recolhidas por um conjunto de sensores e previsões de variáveis meteorológicas, que são depois combinadas num contexto de otimização.

A metodologia adotada passou pelo desenvolvimento de um algoritmo baseado no princípio de *Model Predictive Control* no qual é fundamental possuir antecipadamente o comportamento isolado das diferentes fontes de energia quando aplicadas ao sistema, garantindo desta forma a capacidade de fazer uma previsão da saída do sistema nas mais diversas combinações das fontes de energia.

O trabalho desenvolvido permite ao utilizador monitorizar, alterar e ajustar a temperatura da água na piscina através da utilização de uma plataforma *web* e de uma Interface Homem-Máquina (HMI) que combinadas permitem a interação entre o utilizador e o algoritmo.

Palavras-Chave: *Model Predictive Control*, Energias Renováveis, Controlo, Otimização, Previsões.

ABSTRACT

Since a few years ago, a lot has been mentioned about climate change as a factor that puts people's quality of life at risk. One of the causes of this change is the use of energy produced by non-renewable sources, which causes greenhouse gas emissions.

Investing in renewable energy generation simultaneously protects people's quality of life, avoids greenhouse gas emissions and at the same time saves money because it is a cheaper source of energy when compared to fossil fuels.

The purpose of this work was to develop an algorithm using renewable energy sources, with the aim of coordinating the operation of a set of thermal energy and thermal storage sources to adjust the water temperature of outdoor swimming pools according to the user's needs. To do this, it uses information collected by a set of sensors together with weather forecasts, which are then combined in an optimization context.

The methodology adopted considered the development of an algorithm based on the principle of Model Predictive Control in which it is essential to know in advance the isolated behavior of the different energy sources when applied to the system, thus guaranteeing the ability to predict the system's output in the most diverse combinations of energy sources.

The work developed allows the user to monitor, change and adjust the water temperature in the pool through the use of a web platform and a Human Machine Interface (HMI) which, combined, allow interaction between the user and the algorithm.

Keywords: Model Predictive Control, Renewable Energy, Control, Optimization, Forecasting.

ÍNDICE

| | |
|--|----------|
| 1. Introdução | 1 |
| 1.1. Problema | 3 |
| 1.2. Objetivos | 4 |
| 1.3. Estrutura do trabalho | 4 |
| 2. Estado da Arte..... | 7 |
| 2.1. Introdução | 7 |
| 2.2. Soluções energéticas sustentáveis para piscinas | 9 |
| 2.2.1. Coletores solares | 9 |
| 2.2.2. Painéis solares fotovoltaicos | 9 |
| 2.2.3. Materiais de mudança de fase | 13 |
| 2.2.4. Energia Geotérmica | 17 |
| 2.2.5. Permutador de calor de terraço | 18 |
| 2.3. Controlador lógico programável | 19 |
| 2.3.1. Protocolos de comunicação modbus | 21 |
| 2.4. Métodos de controlo | 22 |
| 2.4.1. Controlo adaptativo por modelo de referência..... | 22 |
| 2.4.2. Controlo proporcional integrativo e derivativo..... | 24 |
| 2.4.3. Regulador quadrático linear | 24 |
| 2.4.4. Controlo preditivo baseado em modelo | 25 |
| 2.5. Identificação dos subsistemas | 27 |
| 2.6. Métodos de previsão | 29 |
| 2.6.1. Métodos convencionais para modelação de séries temporais..... | 29 |
| 2.6.2. Modelo autorregressivo com entrada exógena | 30 |

| | |
|--|-----------|
| 2.6.3. Média móvel autorregressiva com entrada exógena..... | 31 |
| 2.6.4. Média móvel integrada autorregressiva com entrada exógena | 32 |
| 2.6.5. Média móvel integrada autorregressiva sazonal com entrada exógena | 33 |
| 2.6.6. Redes neuronais | 34 |
| 2.7. Algoritmo genéticos..... | 39 |
| 2.7.1. Princípio básico do funcionamento do algoritmo genético..... | 40 |
| 2.8. Critérios e métricas | 43 |
| 2.8.1. Critérios de informação de akaike | 43 |
| 2.8.2. Critério de informação de bayes | 43 |
| 2.8.3. Critério de informação de Hannan-quinn | 44 |
| 2.8.4. Erro quadrático médio..... | 44 |
| 2.8.5. Soma dos quadrados dos erros | 45 |
| 2.8.6. Erro médio relativo | 45 |
| 2.8.7. Raiz do erro quadrático médio | 46 |
| 2.8.8. R-quadrado (R^2) | 46 |
| 3. Trabalho realizado | 47 |
| 3.1. Caracterização do sistema a controlar..... | 47 |
| 3.2. Modelação dos subsistemas | 50 |
| 3.3. Controlo on-off | 50 |
| 3.3.1. Controlo preditivo baseado em modelo | 53 |
| 3.3.2. Algoritmo genético para controlo do subsistema de caldeira | 55 |
| 3.4. Estratégia de controlo inteligente proposta..... | 56 |
| 3.5. Plataforma inteligente | 59 |
| 3.6. Desenvolvimento de API RESTful para gestão de piscinas com integração em flask, html e css | 62 |
| 4. Resultados..... | 67 |
| 4.1. Preparação e divisão de dados para treino e teste utilizando o matlab | 67 |

| | |
|--|------------|
| 4.1.1. Rede perceptron multi - camadas..... | 69 |
| 4.1.2. Rede RNN-Long-Short-Term Memory | 71 |
| 4.1.3. Média móvel autorregressiva com entrada exógena (ARMAX) | 73 |
| 4.1.4. Controle MPC generalizado utilizando Algoritmo Genético (Caso 1)..... | 78 |
| 4.1.5. Proposta de controle utilizando algoritmo genético (Caso 2)..... | 81 |
| 4.2. Preparação e divisão de dados para treino e teste utilizando o SARIMAX..... | 83 |
| 4.3. Plataforma web | 90 |
| 4.3.1. Monitorização em tempo real | 91 |
| 4.3.2. Alteração dos parâmetros dos subsistemas | 92 |
| 5. Conclusões e Trabalhos Futuros | 93 |
| 5.1. Análise do trabalho realizado..... | 93 |
| 5.2. Trabalhos futuros | 94 |
| 5.3. Publicações e contribuições relacionadas com o trabalho realizado | 94 |
| Referências | 97 |
| Apêndices..... | 107 |
| A Matlab. | 109 |
| A.1 Apêndices I | 109 |
| A.2 Apêndices II | 110 |
| A.3 Apêndices III..... | 112 |
| B Python..... | 119 |
| B.1 Apêndices I | 119 |
| B.2 Apêndices II | 126 |
| B.3 Apêndices III..... | 133 |
| B.4 Apêndices III..... | 134 |
| B.5 Apêndices IV | 136 |
| B.6 Apêndices V | 136 |
| B.7 Apêndices VI | 137 |

| | | |
|-----|---------------------|-----|
| B.8 | Apêndices VII | 138 |
| C | .Html..... | 141 |
| C.1 | Apêndices I | 141 |
| C.2 | Apêndices II | 142 |
| C.3 | Apêndices II | 143 |
| C.4 | Apêndices III..... | 144 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 2.1. Diferentes tipos de tecnologia solar fotovoltaica (adaptado de [20]): a) 60 mono cells; b) 120 half-cut mono-cells; c) 60 or 96 Interdigitated Back Contact (IBC) cells (Células IBC); Shingled cells (Células sobrepostas ou telhadas); 60 multi busbar cells (Células com múltiplos barramentos) | 10 |
| Figura 2.2. Esquema de um sistema convencional de aquecimento solar de água (adaptado de [23])..... | 12 |
| Figura 2.3. Vários tipos de coletores solares (obtido de [24]) | 13 |
| Figura 2.4. Tipos de materiais de mudança de fase (adaptado de [26])..... | 14 |
| Figura 2.5. Esquema do sistema de aquecimento da água da piscina ([37])..... | 16 |
| Figura 2.6. Gamas de temperatura dos materiais de mudança de fase e aplicações (adaptado de [38])..... | 16 |
| Figura 2.7. MicroLogix 1400 Controller..... | 20 |
| Figura 2.8. Protocolo de comunicação Modbus rtu | 22 |
| Figura 2.9. Estrutura genérica do MRAC (adaptado de [51])..... | 23 |
| Figura 2.10. Diagrama de blocos de um circuito de controlo de feedback clássico (adaptado de [54])..... | 24 |
| Figura 2.11. A estrutura do sistema de controlo LQR (adaptado de [56])..... | 25 |
| Figura 2.12. Diagrama de blocos simplificado de um circuito de controlo baseado em MPC (adaptado de [58])..... | 26 |
| Figura 2.13. Sistema com saída y , entrada(s) u , perturbação medida w e perturbação não medida v (adaptado de [62]) | 27 |
| Figura 2.14. Arquitetura de rede Neuronal Perceptron Multicamadas (MLP) ([75]) ... | 36 |
| Figura 2.15. Esquemático de fluxo de dados na camada escondida duma rede RNN-LSTM tradicional ([76]) | 38 |
| Figura 2.16. Algoritmo genético padrão (adaptado de [74])..... | 40 |
| Figura 2.17. Princípio de funcionamento de um algoritmo genético (adaptado de [74]) | 41 |
| Figura 3.1. Arquitetura da integração dos sistemas de aquecimento | 48 |

| | |
|--|----|
| Figura 3.2. O cenário considerado para a regulação da temperatura da água. Esquerda: a caldeira é utilizada como elemento de aquecimento. Direita: um coletor solar é utilizado como elemento de aquecimento..... | 51 |
| Figura 3.3. Princípio de controlo ON-OFF | 52 |
| Figura 3.4. Modelação numérica para os meses de maio e junho, comparando as situações com e sem controlo, considerando um valor desejado de temperatura de 25°C e $\Delta T=2^\circ\text{C}$ | 52 |
| Figura 3.5. Diagrama de fluxo para Controlo Preditivo de Modelos..... | 53 |
| Figura 3.6. Resposta do sistema baseada no controlo preditivo de modelos generalizado | 55 |
| Figura 3.7. Diagrama temporal da abordagem proposta | 58 |
| Figura 3.8. Arquitetura global do sistema desenvolvido..... | 59 |
| Figura 3.9. Diagrama relacional entre as entidades constituintes da plataforma inteligente..... | 60 |
| Figura 3.10. Fluxograma da interface homem máquina..... | 62 |
| Figura 3.11. Fluxograma da aplicação inteligente baseado na arquitetura REST..... | 62 |
| Figura 4.1. Divisão dos dados de temperatura exterior ambiente e humidade nos conjuntos de treino e teste..... | 68 |
| Figura 4.2. Divisão dos dados de temperatura real da água do tanque e velocidade do vento nos conjuntos de treino e teste | 68 |
| Figura 4.3. Estrutura do modelo Perceptron Multicamadas, onde n representa o número de neurónios, w representa o peso e b representa os biases..... | 69 |
| Figura 4.4. Comparação entre a temperatura prevista pelo modelo e a temperatura real do sistema, obtida através de simulação no TRNSYS utilizando rede MLP..... | 71 |
| Figura 4.5. Comparação entre a temperatura prevista pelo modelo e a temperatura real do sistema, obtida através de simulação no TRNSYS. A curva vermelha indica o MRE entre as duas temperaturas utilizando rede LSTM..... | 73 |
| Figura 4.6. Previsão utilizando janelas de 80 dados à esquerda e de 160 à direita..... | 77 |
| Figura 4.7. Previsão utilizando janelas de 320 dados à esquerda e de 1920 à direita... | 77 |
| Figura 4.8. Utilização de Caldeira para aquecimento da água do tanque da piscina | 78 |
| Figura 4.9. Utilização do algoritmo genético: topo: Tsetpoint de 22 °C; meio: temperatura da água à saída da caldeira; abaixo: Temperatura prevista para o tanque da piscina | 79 |

| | |
|--|----|
| Figura 4.10. Comparação entre o controlo baseado no esforço assegurado (Caldeira + Temperatura da piscina) com um valor de MRE% de 6,15%, o melhor esforço (Temperatura da água do coletor) com um valor de MRE% de 14,10% e nenhum controlo | 80 |
| Figura 4.11. Comparação entre: (*) os valores estimados da temperatura da água de saída da caldeira como resultado da aplicação do controlo baseado no esforço garantido e (+) a temperatura da água de saída dos coletores solares como resultado do controlo baseado no melhor esforço | 81 |
| Figura 4.12. Primeira e décima melhores configurações obtidas pelo algoritmo genético para o funcionamento do sistema durante 12 horas | 83 |
| Figura 4.13. Janelas de dados correspondente a dois dias (48 dados de 30 em 30 minutos), utilizadas para treinar a rede | 84 |
| Figura 4.14. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado | 86 |
| Figura 4.15. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado | 86 |
| Figura 4.16. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado | 87 |
| Figura 4.17. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado | 87 |
| Figura 4.18. Variações do fitness de acordo com o tamanho da população ao longo das gerações..... | 88 |
| Figura 4.19. Previsão da temperatura da piscina baseada na melhor coordenação dos subsistemas térmicos, para um período de 8 horas de previsão (das 16h até às 24h)..... | 90 |
| Figura 4.20. Aspeto da plataforma Web. Ilustração das telas mostrando os passos para o registo e login na plataforma..... | 91 |
| Figura 4.21. Aspeto da plataforma Web. Exemplo de telas com a monitorização através da plataforma | 91 |
| Figura 4.22. Aspeto da plataforma Web. Exemplo de telas com o modo de funcionamento e formulário para alterar os parâmetros desejados..... | 92 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 2.1. Vantagens e desvantagens de tipos de materiais de mudanças de fase (adaptado de [35][36]) | 15 |
| Tabela 2.2. Características técnicas da série Micrologix da Allen-Bradley | 20 |
| Tabela 2.3. Formato de trama de dados de <i>Modbus rtu</i> | 21 |
| Tabela 2.4. Comparação entre os métodos de controlo (adaptado de [59]) | 26 |
| Tabela 3.1. Resumo da caracterização dos subsistemas | 47 |
| Tabela 3.2. Configuração dos cenários de funcionamento, onde: Vi corresponde à válvula i, Bj corresponde à bomba j, A significa aberta, L significa ligada | 49 |
| Tabela 3.3. Entidades e resumo de funcionalidades | 61 |
| Tabela 4.1. Resultado de simulação variando o número de neurónios n entre 5 e 90, em intervalos de 5 | 70 |
| Tabela 4.2. Resultado da simulação encontrado para diferentes parâmetros da rede | 72 |
| Tabela 4.3. Subsistema predefinido e respetiva numeração | 73 |
| Tabela 4.4. Modelos encontrados em fevereiro para vários subsistemas | 75 |
| Tabela 4.5. Modelos encontrados em março para vários subsistemas | 75 |
| Tabela 4.6. Modelos encontrados em julho para vários subsistemas | 76 |
| Tabela 4.7. Métricas de desempenho dos modelos para diferentes tamanhos das janelas | 77 |
| Tabela 4.8. Graus das polinomiais e respetivo valor de R2 | 79 |
| Tabela 4.9. Resumo das ilustrações de alguma ordem encontrada para os quatros subsistemas | 85 |

LISTA DE ACRÓNIMOS

| ACRÓNIMO | Significado |
|-----------------|---|
| AG | Algoritmo Genético |
| API | Interface de Programação de Aplicação |
| ARIMA | Média Móvel Integrada Autorregressiva |
| ARIMAX | Média Móvel Autorregressiva com Entrada Exógena |
| ARX | Autorregressivo com Entrada Exógeno |
| ANN | Redes Neurais Artificiais |
| ANN-GA | Redes Neurais Artificiais otimizadas por Algoritmo Genético |
| ASCII | American Standard Code for Information Interchange |
| ASHP | Air-Source Heat Pump |
| BD | Base de Dados |
| CO ₂ | Dióxido de Carbono |
| CSS | Cascading Style Sheets |
| CSP | Concentrating Solar Power |
| EU | União Europeia |
| HE | Troca de Calor |
| HTML | Hyper Text Markup Language |
| HMI | Interface Homem-Máquina |
| LQR | Regulador Quadrático Linear |
| MLP | Perceptron Multicamadas |
| MPC | Controlo Preditivo de Modelo |
| MRAC | Controlo Adaptativo de Referência Baseado em Modelo |
| MSE | Erro Médio Quadrático |
| MRE | Erro Relativo Médio |
| PCM | Material de Mudança de Fase |
| PID | Controlo Proporcional Integrativo e Derivativo |
| PLC | Controlador Lógico Programável |
| RTU | Unidade de Terminal Remota |
| SGE | Sistema de Gestão de Energia |
| SSE | Soma dos Quadrados dos Erros. |

1. INTRODUÇÃO

A União Europeia e os seus governos nacionais estabeleceram objetivos claros para orientar a política ambiental europeia até 2020 e uma visão a mais longo prazo (para os próximos 30 anos). Programas específicos de investigação, legislação e financiamento pretendem proteger, conservar e melhorar o capital natural da União Europeia (UE), para transformá-lo numa economia verde, competitiva, de baixo carbono e eficiente em termos de recursos. Estas orientações visam proteger os cidadãos europeus das pressões e riscos para a saúde e promover o bem-estar relacionado com o ambiente.

Neste contexto, uma parte significativa da investigação dedica-se a explorar a utilização da energia eólica, da biomassa e da energia solar para substituir as fontes de energia convencionais, uma vez que apresentam muitas vantagens, como a redução dos impactos ambientais, e são concebidas como um meio de cumprir as normas ambientais europeias e mundiais [1].

De facto, há um interesse crescente na descarbonização da geração de eletricidade, com foco especial na introdução de Fontes de Energia Renovável (FER). Tendo em consideração os dados de consumo de energia elétrica, fornecidos pelo portal de estatísticas português Pordata [2], em Portugal e durante o ano de 2022, o consumo de energia correspondeu a um total de 21.315 mil tep (Tonelada Equivalente de Petróleo), entre as quais 6.557 mil de tep foram geradas a partir de fontes de energia renováveis. Estes últimos números identificam Portugal como um dos países europeus onde a penetração efetiva das energias renováveis é significativa (uma vez que é responsável por 30,0% do consumo total de energia) com um impacto superior quando comparado com o valor médio de 11,7% na UE. Portanto, a demanda por novas fontes de flexibilidade e o crescente reconhecimento da natureza multienergética das regiões estão a aumentar o interesse na interação entre os setores de energia, como eletricidade, aquecimento/refrigeração e gás e na quantidade significativa de flexibilidade suportada pelos sistemas de aquecimento [3][4].

De acordo com o *United States Census Bureau* [5], a natação é a quarta atividade esportiva mais popular nos Estados Unidos e uma boa maneira de obter atividade física aeróbica regular. Além dos benefícios desportivos e de saúde, as piscinas também estão intimamente relacionadas com as atividades de lazer, especialmente nas áreas turísticas [6]. Por estas razões, tanto as piscinas interiores como as exteriores podem ser encontradas em todo o mundo e, como referido em [7], as estatísticas indicam que cerca de 140 milhões de pessoas nadam na Europa (cerca de 20% da população) para um universo de 13 milhões de piscinas a nível

global. Aproximadamente 29% dessas piscinas encontram-se na Europa e 59% na América do Norte, com o resto do mundo a registar cerca de 12%.

A maioria das piscinas existentes é amplamente ineficiente devido às perdas de água e calor, tendo um impacto considerável na pegada ambiental associada. A este nível, refira-se que em [8] foi calculado o impacto ambiental médio de uma piscina residencial no estado do Arizona (EUA) e noutras regiões mais quentes, destacando-se uma pegada energética que varia entre 2400 e 2800 kWh/ano, uma pegada hídrica de até 185 m³/ano e 1400 kg ± 50 kg de CO₂e/ano.

Fatores como 1) evaporação, 2) radiação para o céu; 3) a convecção próxima à superfície da água e 4) e a convecção através das paredes e do piso até o solo, estão intimamente relacionadas à alta demanda de consumo de energia para aquecimento de piscinas externas. Por um lado, a água das piscinas é um excelente meio para a acumulação de calor, pelo que quando a radiação solar é elevada, particularmente nos meses de verão, as condições para a sua utilização melhoram. No entanto, fora desses períodos, as condições ambientais tendem a causar menores ganhos de calor livre e, simultaneamente, maiores perdas térmicas.

A maneira mais comum de usar a energia solar é converter a luz solar em energia térmica para produzir água quente, usando coletores solares térmicos. Nesse caso, o mecanismo básico usa a radiação solar incidente para gerar calor, pois converte a energia irradiada em energia térmica. Existem muitas aplicações diferentes em que a energia térmica solar pode ser usada, como sistemas domésticos de aquecimento de água, aquecedores de piscina e sistemas de aquecimento de ambientes [4].

Para aquecer as piscinas exteriores, as bombas de calor de fonte de ar são amplamente utilizadas. Por vezes estes sistemas são combinados com outras soluções, como por exemplo em [9] onde se propõe um novo sistema de aquecimento para piscinas exteriores que integra uma bomba de calor de fonte de ar (para fornecimento de calor) e um tanque de armazenamento de material de mudança de fase (PCM), para armazenamento de calor. Além disso, em [10] um sistema de aquecimento integrado PCM para natação ao ar livre é estudado. Os resultados mostram que, em comparação com o sistema de aquecimento tradicional, o custo operacional pode ser reduzido em quase 85%.

Para além destes trabalhos, outros como em [9][11], abordam a melhoria da eficiência utilizando um número pequeno de sistemas térmicos solares e a gás.

Neste contexto, o trabalho apresentado nesta Dissertação propõe um novo esquema para aumentar e controlar a temperatura da água de uma piscina utilizando sistemas térmicos sustentáveis suportados em dispositivos locais de geração renovável. O objetivo do sistema de controlo é coordenar o funcionamento de um conjunto de fontes de energia térmica e

armazenamento térmico, para ajustar a temperatura da água das piscinas externas de acordo com as necessidades do utilizador. Para tal o sistema baseia-se em informações obtidas a partir de um conjunto de sensores. A informação (variáveis ambientais e modelação da função de transferência interna dos subsistemas) é depois combinada com uma estrutura de otimização cujo objetivo é, em última análise, reduzir os requisitos de intervenção humana na manutenção da piscina e proporcionar poupança de recursos para o utilizador final em termos de recursos financeiros e naturais, contribuindo para a sustentabilidade ambiental.

O trabalho desenvolvido nesta dissertação está enquadrado no projeto ALG-01-0247-FEDER-047231 – “ECOPOOL+++”. A solução proposta é uma técnica nova para regulação da temperatura da água aproveitando o uso das energias provenientes de fontes renováveis, de enorme importância para o meio ambiente.

1.1. PROBLEMA

O aquecimento de piscinas exteriores com energias renováveis está em crescimento, impulsionado pela conscientização ambiental e pelos benefícios económicos e ecológicos. As principais vantagens incluem a redução da pegada de carbono, já que fontes como a solar e a geotérmica evitam a emissão de gases de efeito estufa, contribuindo para a mitigação das mudanças climáticas. Embora o investimento inicial em sistemas renováveis seja superior ao dos tradicionais, a economia a longo prazo nos custos energéticos é significativa. Esses sistemas também promovem a sustentabilidade ao garantir um consumo de energia mais responsável e a preservação dos recursos naturais, além de operar silenciosamente e requerer baixa manutenção, reduzindo os custos operacionais.

Em Portugal, o aquecimento solar é a solução mais popular devido à abundância de sol, enquanto as bombas de calor também ganham popularidade por utilizar a energia térmica do ar ou do solo. A escolha do sistema de aquecimento renovável ideal depende de fatores como a localização geográfica, o tamanho da piscina, o orçamento e as preferências do utilizador, sendo essencial a consulta com um profissional especializado. Com a inovação tecnológica e a redução de custos, espera-se que essas soluções se tornem mais acessíveis e eficientes, promovendo um futuro mais sustentável no aquecimento de piscinas exteriores.

Contudo, a integração de energias renováveis no aquecimento de piscinas exteriores apresenta desafios significativos. As fontes de energia renovável, como solar e eólica, são recursos intermitentes, o que significa que a disponibilidade de energia pode variar ao longo

do dia e das estações do ano. Além disso, existem diversos equipamentos a serem controlados, como bombas de calor, painéis solares e sistemas geotérmicos, e a coordenação eficaz desses sistemas é complexa.

1.2. OBJETIVOS

Os objetivos da dissertação passam por: (1) desenvolver um algoritmo que faça previsão e controlo de um conjunto de equipamentos térmicos de uma piscina exterior para alcançar uma determinada temperatura da água, procurando responder às preferências do utilizador e tendo em consideração uma série de variáveis lidas de sensores; (2) desenvolver uma plataforma (Web e HMI) que permitirá ao utilizador interagir com o algoritmo desenvolvido; (3) projetar a arquitetura do funcionamento do sistema de controlo e (4) criar um mecanismo capaz de monitorizar todos os equipamentos pertencentes ao sistema.

1.3. ESTRUTURA DO TRABALHO

Este relatório está dividido e organizado em 5 capítulos. No capítulo atual, é apresentado o enquadramento ao problema, os objetivos e organização do relatório.

No capítulo seguinte (Capítulo 2), é apresentada uma revisão bibliográfica de alguns trabalhos publicados e são abordados conceitos fundamentais relativos aos diversos sistemas habitualmente utilizados para o aquecimento de piscinas. Neste capítulo também se descreve o funcionamento dos autómatos programáveis (*Programmable Logic Controller*), o protocolo de comunicação Modbus, e apresenta-se um resumo dos métodos de controlo relacionados com a temática, e dos algoritmos de previsão e otimização (no caso, o algoritmo genético). É também neste capítulo que se definem os critérios e métricas utilizados.

No Capítulo 3 é apresentada a caracterização do sistema de controlo, modelação do subsistema, controlo on-off, controlo preditivo baseado em modelo, algoritmo genético para controlo do subsistema de caldeira, estratégia de controlo inteligente proposta, plataforma inteligente e por fim o desenvolvimento de API RESTful para gestão de piscinas com integração em Flask, *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS).

Com o Capítulo 4, os algoritmos e metodologias apresentadas são aplicadas em ambiente simulado e real, utilizando dados monitorizados coletados a partir do protótipo existente.

No Capítulo 5 são tecidas as conclusões sobre o trabalho realizado e descritos os passos e desenvolvimentos futuros com vista a melhorar o sistema de controlo proposto.

2. ESTADO DA ARTE

Neste capítulo começaremos por fazer uma revisão bibliográfica e descrever os conceitos associados aos subsistemas utilizados na piscina. Começa-se por abordar as soluções sustentáveis para geração de energia em piscinas, os equipamentos de controlo desses subsistemas. Finaliza-se com a descrição dos métodos de controlo e previsão que se podem utilizar para coordenar os equipamentos, assim como dos critérios e métricas utilizados para os avaliar.

2.1. INTRODUÇÃO

Ao falar sobre os métodos de aquecimento das piscinas, não se deve esquecer que existem várias formas de o fazer, incluindo sistemas tradicionais que eram muito utilizados antigamente e sistemas controlados que permitem um uso mais eficiente. Os métodos tradicionais de aquecimento incluem sistemas a gás, elétricos, híbridos (combinando gás e outras fontes), bombas de calor e aquecimento solar.

O que distingue os sistemas de aquecimento atuais das piscinas dos sistemas tradicionais são as tecnologias empregues, cuja finalidade permite ter total controlo sobre os sistemas instalados, permitindo uma gestão mais eficaz dos mesmos.

O trabalho publicado em [12] propõe um novo esquema de monitorização e controlo da qualidade da piscina (pH, cloro, nível de água e temperatura) através de um sistema de baixo custo baseado em redes de sensores sem fios. Apesar de ter benefícios económicos por consumir menos recursos naturais e materiais, não considera o conforto térmico e não inclui sistemas térmicos sustentáveis.

De forma semelhante, um sistema de informação de piscinas baseado na web foi apresentado por Marais et al. [13]. Numerosos elementos de manutenção da piscina, como o pH, os níveis de cloro e o nível da água, podem ser monitorizados remotamente e configurados de acordo com o programa definido pelo utilizador.

Uma das tecnologias de aquecimento mais populares para piscinas exteriores, que também é amiga do ambiente, baseia-se em tanques de armazenamento de PCM. Em [14], Y Li e G Huang discutiram a sua aplicação a piscinas exteriores e mostraram que podem trazer benefícios económicos simplesmente deslocando o consumo de eletricidade dos períodos de pico para os períodos fora de pico. A regulação da temperatura da água é feita através do controlo ON-OFF de uma bomba utilizando um modelo de piscina em que se assume que a

temperatura da água da piscina é igual à temperatura da água ao sair da piscina. A análise numérica e as simulações foram efetuadas numa plataforma que combina Matlab e TRNSYS.

Num trabalho mais recente [15], Y Li e G Huang mostraram que o desempenho do depósito de PCM pode ser melhorado propondo uma nova abordagem em que o conforto térmico é considerado em vez de se utilizar a temperatura da água à saída da piscina. Para tal, incorporaram registadores de irradiação solar, registador de dados, anemómetro ultrassónico, sensores de temperatura e outros sensores para recolher dados de campo.

Outros autores [16] analisaram a eficiência energética utilizando um sistema combinado de água quente composto por coletores solares térmicos e uma central térmica a gás natural. Concluíram que as fontes de energia térmica que utilizam gás natural e energia solar continuam a ser as melhores soluções em termos de eficiência energética, baixa poluição e custos operacionais.

Em [17], os autores usam o controlo preditivo para mostrar que a regulação da temperatura da água de piscinas interiores pode tornar-se mais eficiente em termos energéticos quando um sistema híbrido solar e caldeira (possivelmente alimentado por biomassa) é usado como fonte térmica. Os seus resultados indicam que a regulação baseada no controlo preditivo pode manter as condições térmicas da piscina, reduzindo a procura de energia. Além disso, mostram também que esta abordagem consome menos combustível quando comparada com o controlo tradicional Proporcional Integral Derivativo (PID). Este sistema ainda não foi aplicado a piscinas exteriores.

Em [18], Dong et al, propõem um sistema de controlo integrado composto por um algoritmo de otimização automática difusa e o compensador preditor Smith para ajustar a temperatura da água da piscina. As simulações mostram que este sistema consegue obter bons efeitos de controlo para sistemas de controlo da temperatura de piscinas com atrasos e inércias graves.

No entanto, apesar dos progressos significativos observados na literatura, ainda não existe um trabalho que integre ou coordene de forma eficaz várias fontes de energia para o aquecimento de piscinas. Essa lacuna é evidente, indicando uma oportunidade de pesquisa futura para o desenvolvimento de um sistema integrado que combine diferentes fontes de energia e tecnologias de controle, maximizando a eficiência e minimizando o impacto ambiental do aquecimento de piscinas.

Neste contexto, esta dissertação propõe justamente um sistema inovador que visa integrar e coordenar o funcionamento de múltiplas fontes de energia para o aquecimento de piscinas. A solução apresentada neste trabalho não apenas preenche a lacuna identificada na literatura, mas também avança no sentido de proporcionar uma gestão otimizada, sustentável e eficiente

do aquecimento de piscinas, combinando tecnologias avançadas de controle com diversas fontes energéticas.

2.2. SOLUÇÕES ENERGÉTICAS SUSTENTÁVEIS PARA PISCINAS

Neste subcapítulo faz-se a descrição de algumas tecnologias utilizadas em piscinas e que permitem reduzir os requisitos energéticos das mesmas. Em particular descreve-se o funcionamento e aplicações dos sistemas dos painéis solar fotovoltaicos, Coletores Solares, PCM, Sistemas de Energia Geotérmicas e por fim Permutador do Calor de Terraço que estão relacionados com os sistemas utilizados no trabalho.

2.2.1. COLETORES SOLARES

Um coletor solar é um dispositivo especializado que transforma a energia da radiação solar em energia térmica no fluido de trabalho para aplicações solares térmicas, ou em energia elétrica direta em aplicações fotovoltaicas. Em sistemas solares térmicos, o coletor solar absorve a radiação solar e converte-a em calor, que é transferido para o fluido de trabalho, como ar, água ou óleo. Este calor pode ser utilizado para aquecer água para uso doméstico ou para aquecimento de ambientes, ou ainda para carregar um tanque de armazenamento de energia térmica, de onde o calor pode ser retirado para uso posterior (durante a noite ou em dias nublados). Nas aplicações fotovoltaicas, um módulo fotovoltaico, também conhecido como painel fotovoltaico, converte diretamente a radiação solar em energia elétrica (normalmente com uma eficiência relativamente baixa), mas também gera calor residual, que pode ser recuperado para uso térmico ao ligar a placa fotovoltaica a tubos de recuperação preenchidos com fluidos transportadores [19].

2.2.2. PAINÉIS SOLARES FOTOVOLTAICOS

O painel solar fotovoltaico é um dispositivo que capta radiação solar e a transforma em energia elétrica. Passaram 185 anos desde 1839, quando Alexandre Edmond Becquerel observou o efeito fotovoltaico (PV) através de um eletrodo numa solução condutora exposta à luz [20].

Os painéis solares são divididos em três grandes famílias [21]: painéis solares de película fina, painéis solares monocristalinos e painéis solares policristalinos.

Os painéis solares de película fina consistem em filmes esticados que podem ser facilmente instalados em qualquer lugar apropriado. Este tipo de painel não é afetado pela poeira e pode funcionar em condições adversas. Em tempo nublado, a sua eficácia é reduzida para 20% [21].

Os painéis solares monocristalinos são compostos por um grande número de células individuais, que são preenchidas com silício. Destacam-se pela flexibilidade, leveza, fiabilidade e durabilidade. São fáceis de instalar e dependem da luz solar direta. Neste caso, mesmo uma leve cobertura de nuvens pode levar à interrupção da produção de energia [21].

Os painéis solares policristalinos contêm células compostas por cristais pontiagudos em diferentes direções. Isto torna possível capturar luz difusa e ser menos dependente de iluminação direta. São utilizados com sucesso para alimentar casas, prédios, escritórios e até ruas.

A figura 2.1 apresenta diversos tipos de tecnologias de células solares fotovoltaicas em 2020, cada uma com características distintas.

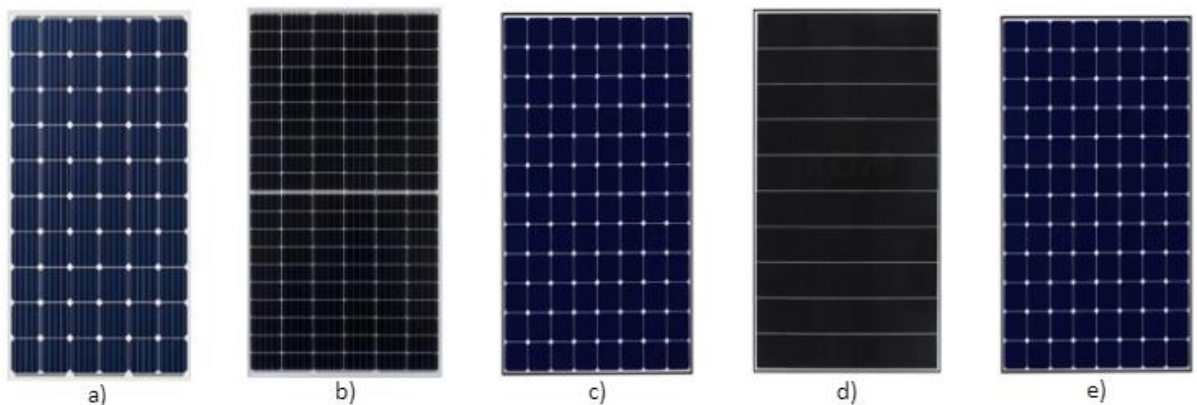


Figura 2.1. Diferentes tipos de tecnologia solar fotovoltaica (adaptado de [20]): a) 60 mono cells; b) 120 half-cut mono-cells; c) 60 or 96 Interdigitated Back Contact (IBC) cells (Células IBC); Shingled cells (Células sobrepostas ou telhadas); 60 multi busbar cells (Células com múltiplos barramentos)

De acordo com [22], as *mono cells* (Células monocristalinas) são células solares feitas de um único cristal de silício. As células monocristalinas são conhecidas pela sua elevada eficiência e longa durabilidade. Elas têm uma aparência uniforme, geralmente de cor preta, e são mais eficientes na conversão da luz solar em eletricidade.

Nas *half-cut* mono-cells (Células monocristalinas cortadas ao meio), as células monocristalinas tradicionais são cortadas ao meio, resultando em 120 células em vez das

tradicionais 60 células. Isso reduz as perdas resistivas e melhora a eficiência e o desempenho em sombra parcial.

As células *Interdigitated Back Contact* (IBC) cells (Células IBC) são uma tecnologia avançada em que os contatos elétricos estão localizados na parte traseira da célula. Isso permite uma área de superfície frontal limpa, maximizando a absorção de luz.

As células *Shingled cells* (Células sobrepostas ou telhadas) são cortadas em tiras finas e sobrepostas umas sobre as outras como telhas. Isso elimina a necessidade de barramentos entre as células, aumentando a área ativa e a eficiência.

As *multi busbar cells* (Células com múltiplos barramentos) utilizam mais barramentos (busbars) para recolher a corrente gerada nas células solares. Mais barramentos resultam em menores perdas de resistência e melhor desempenho geral.

Resumindo estas tecnologias representam avanços no desenvolvimento e na fabricação de painéis solares que permitem melhorar a eficiência, reduzir custos, aumentar a durabilidade e a confiabilidade dos sistemas fotovoltaicos

2.2.2.1. COLETORES SOLARES TÉRMICOS

Um sistema de aquecimento solar de água, como mostra a Figura 2.2, consiste num coletor discreto, concebido para maximizar a absorção solar e reduzir as perdas de calor. O coletor solar pode ser um absorvedor de placa plana de cor preta, ligado a uma tubagem de cobre e coberto com um vidro transparente [23].

Quando a radiação solar atravessa o vidro transparente ou os tubos evacuados (tubos de vácuo) e incide sobre a superfície de elevada absorção do coletor, uma grande parte da energia é absorvida pelo coletor e depois transferida para o fluido a ser transportado nos tubos.

O fluido de transferência de calor, geralmente uma mistura de água e fluido anticongelante, é bombeado, ou conduzido por convecção natural, através do coletor para um permutador de calor em serpentina no fundo de um tanque cilíndrico, onde o calor que transporta é transportado para a água de serviço para armazenamento ou utilização direta. O reservatório contém um aquecedor auxiliar, por exemplo, uma resistência elétrica de imersão ou uma caldeira convencional, para utilização no inverno.

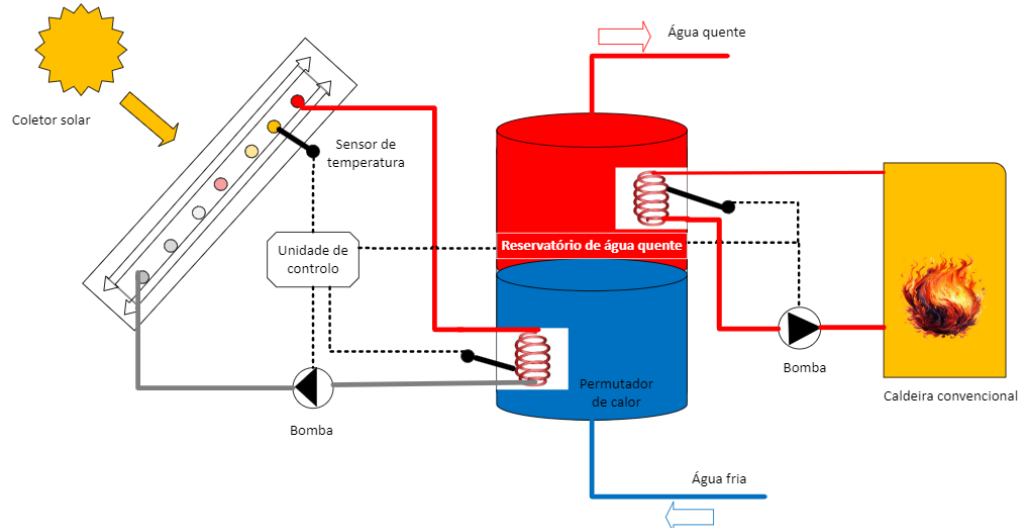


Figura 2.2. Esquema de um sistema convencional de aquecimento solar de água (adaptado de [23])

2.2.2.2. APLICAÇÕES DOS COLETORES SOLARES

O sistema de aquecimento que usam os coletores solares pode ser aplicado em vários casos dependendo se for a nível doméstico e ou industrial, não esquecendo que há vários tipos de coletores solares dependendo da necessidade desejada.

Mais concretamente podem-se aplicar os coletores solares:

- No aquecimento de banhos, para líquidos para lavagem, tinturaria ou processos químicos;
- No aquecimento de ar, para processos de secagem;
- Na geração de vapor de baixa pressão.

Na prática, existem diferentes tipos de coletores solares térmicos, como indicado na figura 2.3. Os mais simples são os unglazed panels, que são os mais adequados para o caso em estudo de aquecimento de piscinas, onde não é necessário um incremento de temperatura acima de alguns graus em relação à temperatura do ar ambiente, pelo que as perdas de calor não são relevantes.

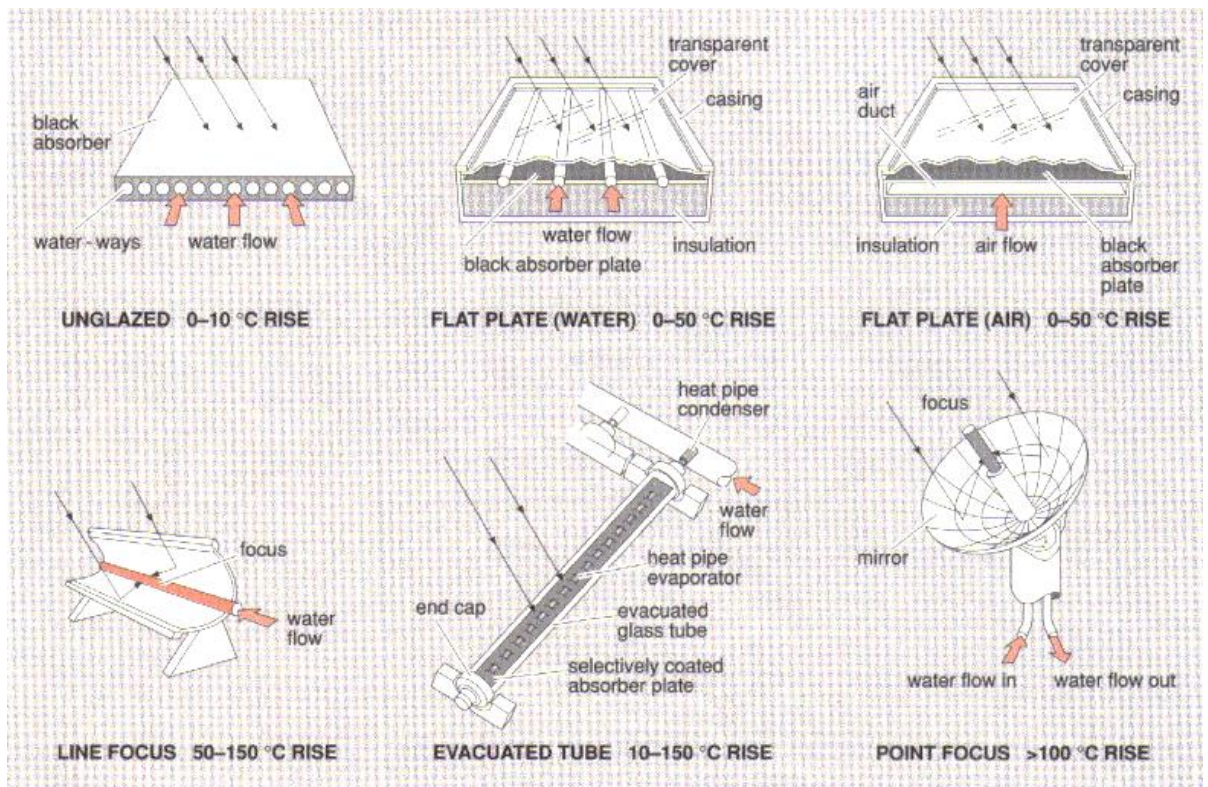


Figura 2.3. Vários tipos de coletores solares (obtido de [24])

2.2.3. MATERIAIS DE MUDANÇA DE FASE

Os PCM captam e armazenam grandes quantidades de energia térmica durante a transição de fase. Uma das propriedades críticas dos materiais de mudanças de fase é a conservação e o armazenamento de energia térmica com um estado de equilíbrio térmico que ocorre após vários ciclos de fase. Enquanto os materiais de mudança de fase trocam calor em ciclos constantes de mudança de fase, também fornecem uma temperatura ambiente mais estável [24][25]. Os materiais existem na natureza nas fases sólida, líquida ou gasosa. Quando ocorre uma mudança de fase numa substância, ela armazena ou liberta algum calor. Este calor armazenado ou libertado é designado por calor latente (LH).

Para ilustrar, quando a temperatura de uma substância sólida é aumentada até à sua temperatura de fusão, esta substância sólida funde-se captando uma grande quantidade de calor e a sua fase transforma-se de sólida para líquida. Os estados de fase dos materiais de mudança de fase podem mudar dentro de um determinado intervalo de temperatura [25].

Os PCM podem ser considerados como um equipamento usado para controlar a oscilação da temperatura ou armazenar energia dentro de um intervalo de tempo específico, podendo ser classificados em 3 tipos conforme se ilustra na figura 2.4 [26].

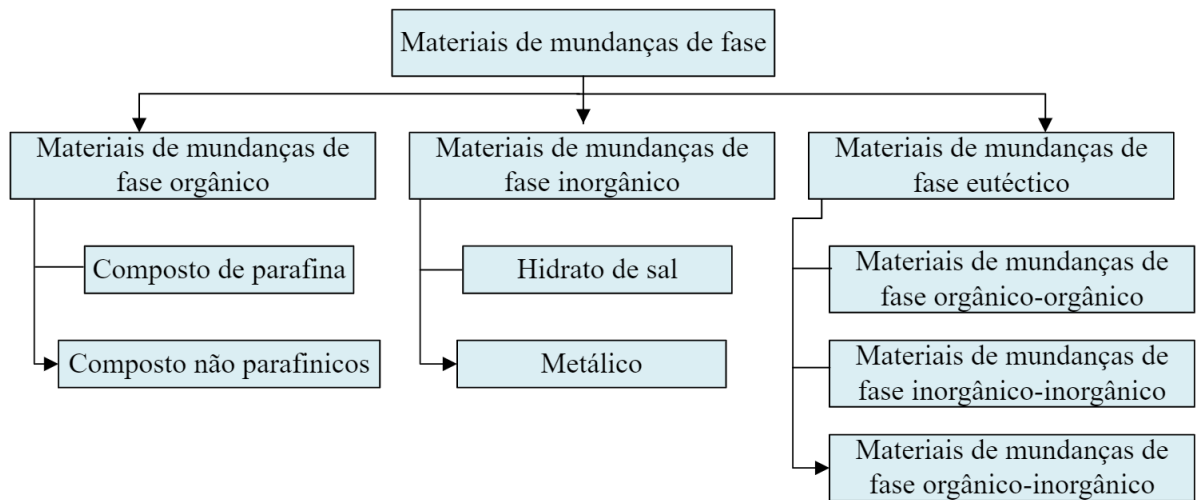


Figura 2.4. Tipos de materiais de mudança de fase (adaptado de [26])

Os PCM orgânicos são um dos tipos de PCM atualmente em uso que é mais preferido. Os mesmos são classificados em dois tipos: parafínicos e não parafínicos.

A parafina está entre os PCM mais utilizados para operações em sistemas de armazenamento de energia térmica [27][28]. As parafinas podem ser utilizadas separadamente ou em combinação umas com as outras a gamas de temperatura desejadas. Os materiais não parafínicos são fabricados a partir de uma variedade de matérias orgânicas, contendo ésteres, ácidos gordos, glicóis e álcoois [29][30].

Ao contrário dos orgânicos, os PCM inorgânicos não são inflamáveis e são muito mais baratos. Os PCM inorgânicos contêm metais, sais, hidratos de sal e ligas metálicas [31]. Os hidratos de sal são geralmente formulados como $AB \cdot nH_2O$; onde n representa o número de moléculas de água na mistura de sal [32]. O sal e os hidratos de sal são fontes abundantes em lagos salgados ou na água do mar, pelo que são competitivos em termos de custos, que ronda 1% do custo da parafina. O custo razoável e as características não inflamáveis dos materiais inorgânicos de mudança de fase incentivam muito mais a sua comercialização do que os materiais orgânicos [33].

Os PCM eutéticos obtêm-se através de uma mistura de compostos ou elementos químicos. Embora existam três tipos diferentes de eutéticos, dois tipos são amplamente utilizados atualmente: materiais de mudança de fase orgânico-orgânicos [34] e materiais de mudança de fase orgânico-inorgânicos. A tabela 2.1 apresenta uma comparação entre os distintos materiais de mudança de fase [35].

Tabela 2.1. Vantagens e desvantagens de tipos de materiais de mudanças de fase (adaptado de [35][36])

| PCM Orgânicos | | PCM Inorgânico | | PCM Eutéticos | |
|---|-----------------------------|-------------------------------|----------------------------|--|---------------------------|
| Vantagens | Desvantagens | Vantagens | Desvantagens | Vantagens | Desvantagens |
| Reciclável com outros materiais | Inflamabilidade | Disponibilidade a baixo custo | Corrosão | Armazenamento térmico volumétrico elevado | Dispendioso |
| Boa compatibilidade química | Baixa densidade | Alteração de baixo volume | Sobreaquecimento | Densidade | Baixo calor latente total |
| Quimicamente estável | Mais caro | Não inflamável | Suscetível de se degradar | Temperatura de fusão acentuada | Odor forte |
| Disponibilidade numa vasta gama de temperaturas | Baixa condutividade térmica | Elevado calor de fusão | Requer contentor e suporte | Sem segregação de fases e mudança de fase congruente | Falta de dados de ensaio |

2.2.3.1. FUNCIONAMENTO DOS MATERIAIS DE MUDANÇA DE FASE

Os PCM absorvem calor num processo endotérmico durante a fusão. O calor é novamente libertado num processo exotérmico quando o PCM solidifica.

O PCM pode ser dividido em sólido-líquido, sólido-sólido e sólido-gás de acordo com as mudanças morfológicas antes e depois da mudança de fase. O volume do PCM sólido-gás é demasiado grande após a mudança de fase, e a placa de molibdénio não pode proporcionar um espaço tão grande, pelo que não é adequada para este estudo. O calor latente do PCM sólido-sólido é pequeno. O calor latente do sólido-líquido é maior do que o do sólido-líquido, e a mudança de volume após a mudança de fase é menor.

As propriedades termofísicas, cinéticas e químicas, bem como as considerações económicas e ambientais são os principais fatores na seleção dos materiais de mudança de fase [26].

A figura 2.5 [37] ilustra um esquema de funcionamento dos materiais de mudanças de fase para aquecimento da água na piscina, que inclui um tanque de armazenamento de PCM (*PCM storage tank* - PST), bomba de calor (*air-source heat pump* - ASHP), cobertura de isolamento térmico (TIC), permutadores de calor (HE), válvulas e bombas. O PST é adotado para armazenar o calor fornecido pelas ASHP durante o período de tarifário em vazio e libertá-lo para a piscina durante o período de tarifário de ponta. Assim, a eletricidade consumida é transferida dos períodos de ponta para os períodos fora de ponta, o que reduz eficazmente o

custo de funcionamento. As ASHP foram adotadas para não só carregar o PST, mas também para pré-aquecer a piscina. A cobertura foi utilizada para reduzir as perdas de calor da piscina durante o período de fecho.

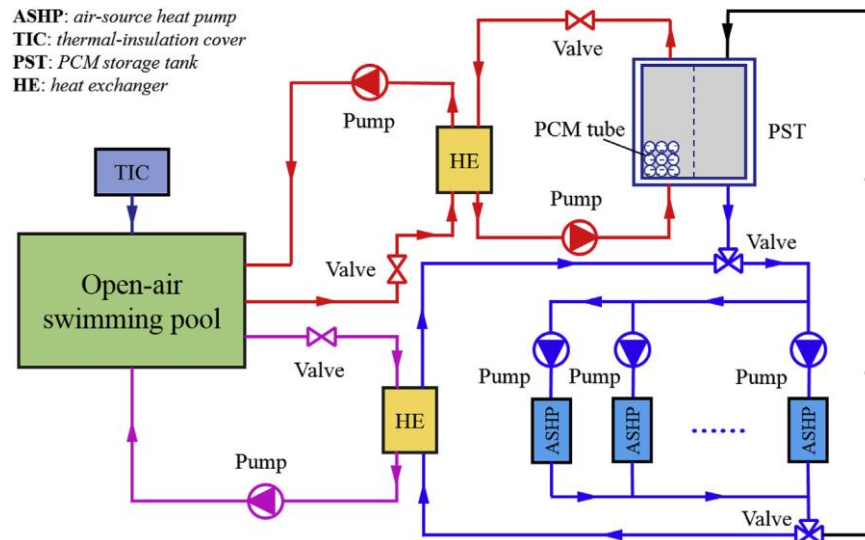


Figura 2.5. Esquema do sistema de aquecimento da água da piscina ([37])

Em termos de controlo, no sistema proposto na figura 2.5, foi utilizado o controlo baseado no tempo onde executam ações com base em um cronograma fixo, sem considerar as condições atuais, sendo mais simples, mas potencialmente menos eficiente enquanto controlo baseado na temperatura ajusta a operação com base em medições contínuas da temperatura atual, proporcionando precisão e resposta às condições reais.

Refira-se que a energia usada para aquecimento/arrefecimento através das bombas de calor é de fonte energia elétrica não necessariamente renovável.

2.2.3.2. APLICAÇÕES DAS MATÉRIAS DE MUDANÇA DE FASE

Os PCM têm sido amplamente utilizados como dispositivos de arrefecimento passivo em várias aplicações, tais como eletrónica de potência, equipamento de comunicações, naves espaciais e aviónica, onde a dissipação de calor pode ser variável no tempo ou periódica. Tem-se registado um aumento na utilização de PCM, especialmente para fins de aquecimento e arrefecimento. Algumas aplicações são descritas na Figura 2.6 [38].



Figura 2.6. Gamas de temperatura dos materiais de mudança de fase e aplicações (adaptado de [38])

Foi referido nas secções anteriores que nem todos os PCM podem ser utilizados em todas as aplicações. Os PCM têm uma determinada gama de temperaturas de fusão e são utilizados em aplicações de acordo com estes valores de temperatura. Os PCM podem ser agrupados em 4 categorias principais, de acordo com a sua gama de temperaturas de funcionamento (Figura 2.6). A primeira gama de temperaturas é a gama de temperaturas baixas (-20 °C a +5 °C) onde os materiais de mudança de fase são utilizados em sistemas de refrigeração/arrefecimento [39]. A segunda faixa é a faixa de temperatura média-baixa (+5 °C a +40 °C), onde os PCM são utilizados em processos de aquecimento/arrefecimento e ar condicionado em estruturas [40]. A terceira gama de temperaturas é a gama de temperaturas médias (+40 °C a +80 °C), em que os PCM são utilizados em sistemas de aquecimento solar/produção de água quente e operações de transferência de calor em eletrónica [41][42]. A última gama de temperaturas é a gama de temperaturas elevadas (+80°C a +1200 °C), em que os PCM são utilizados em aplicações Concentrating Solar Power (CSP) onde se usam espelhos para refletir e concentrar a luz solar em um recetor, sistemas de recuperação de calor residual e projetos de arrefecimento por absorção [43].

2.2.4. ENERGIA GEOTÉRMICA

A energia geotérmica é um tipo de energia renovável extraída do núcleo da terra. Provém do calor gerado durante a formação original do planeta e da decomposição radioativa dos materiais. Esta energia térmica é armazenada em rochas e fluidos no centro da Terra.

O primeiro aquecimento residencial do mundo com água geotérmica foi realizado em *Chaude Aigues* (França), no século XIV. Por outro lado, o primeiro sistema municipal de aquecimento urbano com água geotérmica foi criado em Reykjavik, na Islândia, em 1930. Atualmente, 90% da população total da Islândia vive em casas aquecidas por água geotérmica. Em muitos países, como a França, Rússia, Geórgia, China, Itália e EUA, foram construídos sistemas de aquecimento urbano em grande escala que utilizam água geotérmica [44].

A energia geotérmica, sob a forma de vapor natural e água quente, tem sido explorada desde há décadas para gerar eletricidade, bem como no aquecimento de espaços e em processos industriais [44].

2.2.4.1. APLICABILIDADE DA ENERGIA GEOTÉRMICA

A energia geotérmica pode ser utilizada de três formas: Energia geotérmica direta; Bomba de calor geotérmica e Central geotérmica.

A Energia geotérmica direta utiliza-se em áreas onde existem fontes termais ou reservatórios geotérmicos perto da superfície da terra. A água quente pode ser canalizada diretamente para aquecer casas ou edifícios de escritórios. A água geotérmica é bombeada através de um permutador de calor, que transfere o calor da água para o sistema de aquecimento do edifício. A água usada é injetada de novo num poço, no reservatório, para ser reaquecida e utilizada novamente.

No caso da bomba de calor geotérmica, alguns metros abaixo do solo, o solo ou a água permanecem a uma temperatura constante de 50 a 60 *graus Fahrenheit* (10-15 graus celsius) durante todo o ano. Esta temperatura pode ser utilizada para aquecer ou arrefecer casas e escritórios. O fluido circula através de uma série de tubos (designados por circuitos) sob o solo ou sob a água de uma lagoa ou de um lago e para dentro de um edifício. Um compressor elétrico e um permutador de calor retiram o calor dos tubos e enviam-no através de um sistema de condutas para todo o edifício. No verão, o processo é invertido. Os tubos retiram o calor da casa e transportam-no para o solo ou para a água no exterior, onde é absorvido.

Na central geotérmica a água quente e o vapor das profundezas do subsolo podem ser canalizados através de poços subterrâneos e utilizados para gerar eletricidade numa central elétrica [45].

Nas últimas duas décadas, o calor geotérmico tem sido utilizado numa escala cada vez maior na criação de animais, na piscicultura, na secagem de culturas, aquecimento do solo e também para aquecimento de água [44].

2.2.5. PERMUTADOR DE CALOR DE TERRAÇO

Um permutador de calor é um dispositivo térmico capaz de transferir eficientemente calor de um meio para outro e é uma solução eficiente para fornecer energia térmica num curto espaço de tempo [46].

O principal objetivo de um permutador de calor é transferir calor entre fluidos de diferentes temperaturas. Trata-se, portanto, de um dispositivo térmico utilizado para arrefecer ou aquecer um meio. No nosso quotidiano, deparamo-nos com vários permutadores de calor. Por

exemplo, o radiador do motor de um automóvel, cuja água transfere calor para o ar. Para não falar do ar condicionado, cuja serpentina separa o ar ambiente do gás refrigerante.

2.2.5.1. FUNCIONAMENTO DO PERMUTADOR DE CALOR DE TERRAÇO

Embora o princípio de funcionamento dos permutadores de calor seja o mesmo, estes podem ser diferenciados em termos dos seus mecanismos de troca de calor. Neste sentido, os principais tipos de permutadores de calor são:

- Permutadores de calor de placas: são constituídos por placas paralelas, colocadas umas sobre as outras, que permitem a formação de canais de fluido. Têm duas passagens: uma para o fluido frio e outra para o fluido quente, que circulam em direções opostas.
- Casco e tubo: é o mais utilizado. É constituído por um invólucro e por tubos no seu interior. Um dos fluidos passa através do interior dos tubos e o outro através do exterior. A troca de calor ocorre, portanto, na superfície das condutas.
- Tubulares: variam na sua geometria. Por este motivo, existem várias subcategorias, tais como bitubulares lineares, bitubulares com pinças, multitubulares e com alhetas.

Assim como os outros sistemas mencionados acima (PCM e Energia Geotérmica) o permutador de calor pode ser aplicado nas residências e indústrias com finalidade de aquecer ou arrefecer conforme a necessidade e ou finalidade onde se aplica.

2.3. CONTROLADOR LÓGICO PROGRAMÁVEL

O controlo dos diversos sistemas que compreendem uma piscina é feito através de um Controlador Lógico Programável (PLC). Este é um equipamento robusto usado para automação industrial que pode automatizar um processo específico, uma máquina ou mesmo uma linha de produção inteira. Na Figura 2.7, é mostrado o controlador *MicroLogix 1400* que foi utilizado nesta dissertação.



Figura 2.7. MicroLogix 1400 Controller

A tabela a seguir (Tabela 2.2) mostra uma comparação em termos de características técnicas como número de entradas/saídas, tipos de entradas e saídas, expansibilidade, entre outros parâmetros que ajudam a entender as diferenças entre os modelos para uma aplicação específica entre os PLCs *MicroLogix* 1100, 1200 e 1400 da *Allen-Bradley* [47]:

Tabela 2.2. Características técnicas da série Micrologix da Alen-Bradley

| Característica | <i>Micrologix</i> 1100 | <i>Micrologix</i> 1200 | <i>Micrologix</i> 1400 |
|---------------------------|---|---|---|
| Número de Entradas/Saídas | 10 entradas, 6 saídas (14 no total) | Modelos com 16, 24 ou 40 I/O | 20 entradas, 12 saídas (32 no total) |
| Tipo de Entrada | Digital | Digital/Analógica (dependendo do modelo) | Digital e Analógica |
| Tipo de Saída | Relé ou Transistor | Relé ou Transistor | Relé ou Transistor |
| Expansibilidade | Até 4 módulos de expansão | Até 6 módulos de expansão | Até 7 módulos de expansão |
| Velocidade da CPU | 1K instruções/ms | 0.9K instruções/ms | 1K instruções/ms |
| Memória do Utilizador | 4 KB | 6 KB | 10 KB |
| Portas de Comunicação | 1x <i>Ethernet</i> , 1x RS232/RS485 (DF1) | 1x RS232 | 1x <i>Ethernet</i> , 1x RS232/RS485 (DF1), 1x RS232 (Isolada) |
| Contadores Rápidos | 3 contadores de 20 kHz | 6 contadores de 20 kHz (dependendo do modelo) | 6 contadores de 100 kHz |
| Relógio em Tempo Real | Sim | Sim | Sim |
| <i>Display</i> LCD | Sim, embutido | Não | Sim, embutido |

Ao comparar os PLCs *MicroLogix* 1100, 1200 e 1400 conforme ilustrada na tabela 2.2, fica evidente que cada modelo foi projetado para atender diferentes níveis de complexidade e requisitos de controle onde:

- *MicroLogix* 1100: Este modelo é ideal para aplicações de controle pequenas a médias. Com sua conectividade *Ethernet* integrada e recursos básicos, é uma opção acessível para

quem busca um PLC capaz de realizar tarefas de automação simples, com a vantagem de conectividade para redes industriais.

- *MicroLogix 1200*: Posicionado como um PLC intermediário, o *MicroLogix 1200* oferece mais flexibilidade e capacidade de expansão em comparação ao 1100. Ele é adequado para aplicações que necessitam de um maior número de entradas e saídas, sem comprometer o desempenho, tornando-o uma escolha sólida para sistemas de controle que requerem um pouco mais de complexidade.

- *MicroLogix 1400*: O mais avançado da linha, o *MicroLogix 1400*, combina a maior capacidade de expansão com um conjunto robusto de recursos, como contadores rápidos de 100 kHz e múltiplas portas de comunicação. Este PLC é ideal para aplicações mais complexas que demandam alta velocidade de processamento e um número elevado de pontos de I/O, além de opções avançadas de conectividade e controle.

Assim, a escolha entre esses três modelos dependerá principalmente do tamanho da aplicação, do número de entradas e saídas necessárias, e da complexidade dos requisitos de comunicação e controle. No caso do projeto ECOPOOL+++, a seleção do modelo *MicroLogix 1400* resulta da necessidade de interligação do mesmo com uma série de sensores e atuadores analógicos e digitais. A capacidade de comunicação com dispositivos externos através de protocolos Modbus é neste caso muito relevante.

2.3.1. PROTOCOLOS DE COMUNICAÇÃO MODBUS

O controlador lógico programável comunica com os dispositivos a controlar através de vários protocolos industriais. Um dos protocolos mais comuns é o *Modbus*, definido pela empresa *Modicon* em 1979. Trata-se de um tipo de protocolo de comunicação série. Atualmente *Modbus* é o protocolo de comunicação padrão do sector industrial, sendo também amplamente utilizado para ligar dispositivos industriais. O *Modbus* permite que os dispositivos comuniquem num mesmo barramento (*bus*). Num barramento só pode haver um dispositivo mestre e podem existir um máximo de 247 dispositivos escravos [48].

Tabela 2.3. Formato de trama de dados de *Modbus* rtu

| Código de endereço | Código de Função | Dados | Verificação CRC |
|--------------------|------------------|-----------|-----------------|
| 8bits | 8bits | n x 8bits | 16bits |

Com *Modbus*, o mestre solicita e o escravo responde. Os dois tipos de trama *Modbus* são a ASCII e *Modbus rtu (Remote Terminal Unit)*. Habitualmente o *Modbus* corre sobre barramentos série RS-485, mas pode também correr sobre RS-232, tramas Ethernet ou sobre TCP/IP.

A Figura 2.8 apresenta a arquitetura típica de uma comunicação em *Modbus rtu* [48].

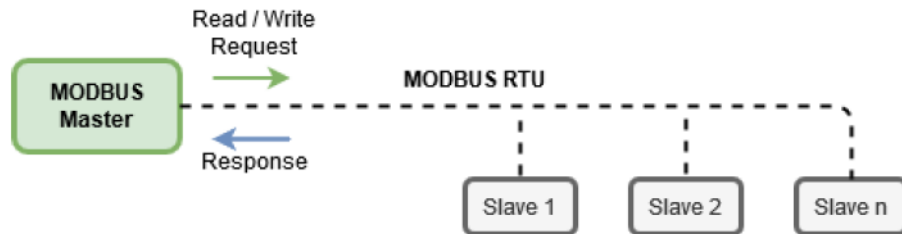


Figura 2.8. Protocolo de comunicação *Modbus rtu*

Para que possa haver comunicação entre o mestre e o escravo deve confirmar-se que os dispositivos no mesmo barramento têm os mesmos parâmetros de comunicação.

2.4. MÉTODOS DE CONTROLO

Neste subcapítulo são descritos os princípios e a formulação matemática de alguns dos métodos de controlo que têm por natureza o funcionamento por antecipação. Em particular descreveremos o controlo adaptativo baseado em modelo de referência, o controlo proporcional integrativo e derivativo, o regulador quadrático linear e por fim o controlo preditivo baseado em modelo.

2.4.1. CONTROLO ADAPTATIVO POR MODELO DE REFERÊNCIA

A estrutura genérica do Controlo Adaptativo Baseado em Modelo de Referência (*Model Reference Adaptive Control, MRAC*) é apresentada na figura 2.9 e é composta pelos seguintes módulos: (i) Modelo de Referência, (ii) Modelo do Processo (sistema), (iii) Controlador e (iv) Algoritmo da Adaptação. A estrutura básica do controlador é especificada por uma construção linear com os correspondentes algoritmos de adaptação em tempo real por uma regra do tipo gradiente ou baseada na teoria da estabilidade [49][50].

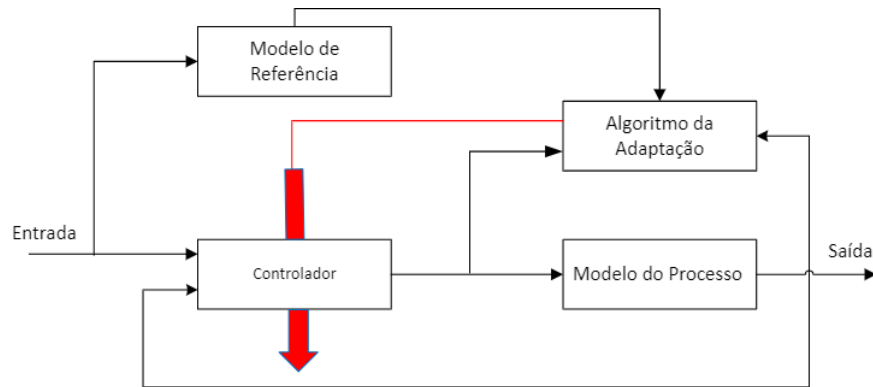


Figura 2.9. Estrutura genérica do MRAC (adaptado de [51])

A principal distinção entre o MRAC e outros métodos reside no facto de as leis adaptativas serem incorporadas e analisadas como parte de todo o sistema de controlo, em vez de se analisar o critério em si. De seguida, discutimos os componentes importantes do sistema MRAC e motivamos a sua aplicação direta [52].

- **Modelo de Referência**

O modelo de referência especifica o comportamento desejado do processo, está normalmente disponível numa representação paramétrica (por exemplo, modelos de função de transferência/espaco de estados) e implementado no computador de controlo. A sua resposta tem de ser exactamente igualada e, portanto, tem de ser estável, de fase mínima e ser razoavelmente representativa do processo em estudo.

- **Controlador**

O controlador deve satisfazer algumas restrições para poder ser parte integrante de um sistema MRAC. Deve: (1) Satisfazer a condição de "correspondência perfeita com o modelo", de modo que os parâmetros de controlo permitam fornecer uma resposta em malha fechada que corresponda à do modelo de referência; e (2) a adaptação direta implica uma função linear dos parâmetros de controlo ou seja, uma lei de controlo linear.

- **Algoritmo da Adaptação**

As regras ou leis de adaptação evoluíram a partir de três abordagens básicas: (1) modelos de sensibilidade; (2) estabilidade de *Lyapunov*; e (3) *Hiper* estabilidade de *Popov* - todos tentando garantir um sistema estável em malha fechada.

- **Modelo do Processo**

Este bloco simula o comportamento do sistema real que está sendo controlado. A saída desse modelo é comparada à saída do "Modelo de Referência", e a diferença entre as duas saídas é usada pelo "Algoritmo da Adaptação" para ajustar o controlador.

2.4.2. CONTROLO PROPORCIONAL INTEGRATIVO E DERIVATIVO

O PID é um controlador fiável com uma estrutura simples, tendo sido utilizado em muitas aplicações industriais durante décadas. É também um instrumento utilizado em aplicações de controlo industrial para regular a temperatura, o fluxo, a pressão, a velocidade e outras variáveis do processo. Os controladores PID utilizam um mecanismo de *feedback* do circuito de controlo para controlar as variáveis do processo e são os controladores mais precisos e estáveis [53]. O controlador PID discreto é representado pela equação 1 a seguir.

A figura 2.10 mostra um diagrama de blocos de um sistema de controle em malha fechada. O controlador (Controlo) recebe um erro e , que é a diferença entre o valor de referência w e a saída real y , e gera um sinal de controle u para o sistema. O sistema é influenciado por uma perturbação z , e o objetivo é ajustar u para que a saída y siga o valor de referência w o mais próximo possível.

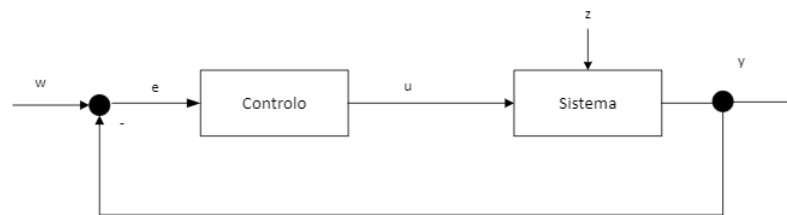


Figura 2.10. Diagrama de blocos de um circuito de controlo de feedback clássico (adaptado de [54])

$$u(k) = K_p(e(k) - e(k-1)) + K_i t_s e(k) + \frac{K_D}{t_s}(e(k) - 2e(k-1) + e(k-2)) + u(k-1) \quad (1)$$

em que $e(k)$, $e(k-1)$, $e(k-2)$ são os erros em diferentes instantes de amostragem, K_p é o ganho proporcional, K_i é o ganho integral, K_D é o ganho derivativo e t_s é o tempo de amostragem.

2.4.3. REGULADOR QUADRÁTICO LINEAR

O Regulador Linear Quadrático (*Linear Quadratic Regulator*, LQR) é uma estratégia de controlo ótimo muito eficaz. Tem sido amplamente desenvolvida e utilizada em diferentes aplicações de controlo. Também é um tipo de controlo ótimo que se baseia na representação do espaço de estados onde o LQR é um controlador com realimentação de estados completo.

A estrutura do LQR retroalimenta o vetor de estados completamente, multiplica-o por uma matriz de ganho e subtrai-o da referência de escala [55].

A figura 2.11 ilustra um sistema de controle em malha fechada utilizando realimentação de estados. O vetor de estados X é realimentado através do ganho K , influenciando a entrada u do sistema dinâmico descrito pelas equações de estado $\dot{X} = Ax + Bu$ e $y = Cx + Du$. Esse tipo de controle é usado para estabilizar e otimizar o desempenho do sistema.

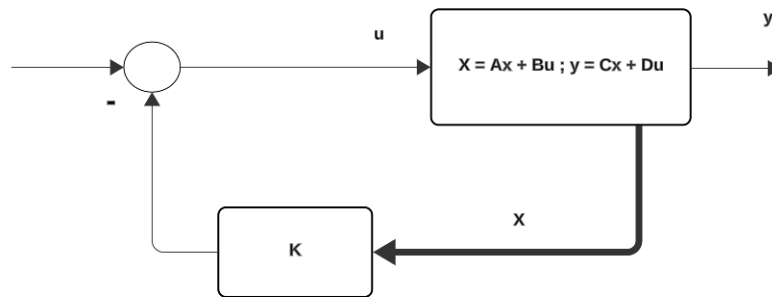


Figura 2.11. A estrutura do sistema de controle LQR (adaptado de [56])

Na concepção LQR, o vetor de ganhos K ótimo é obtido escolhendo as especificações em malha fechada que são importantes para o problema. Especificamente, qual o desempenho do sistema e quanto esforço é necessário para obter esse desempenho [55].

2.4.4. CONTROLO PREDITIVO BASEADO EM MODELO

O Controle Preditivo Baseado em Modelo (*Model Based Predictive Control*, MPC) é uma estratégia de controle avançada e moderna baseado na otimização numérica em tempo real. A saída do processo é prevista para o horizonte temporal usando um modelo estimado do sistema. A entrada do sistema é otimizada em cada instante de tempo de acordo com uma função de custo e restrições. Três conceitos fundamentais na concepção do MPC são a previsão de estados e saídas futuras, a forma de obter a informação atual da instalação e a abordagem da implementação de atividades futuras (realização do controle) [51].

Importa realçar que o MPC é um dos controladores mais adequados para aplicações industriais, especialmente para sistemas com restrições. No entanto, exige uma elevada carga computacional, o que é considerado como o principal inconveniente [57].

A figura 2.12 mostra um diagrama de um sistema MPC. O bloco de "Otimização" recebe o valor de referência w e utiliza o "Modelo" para prever o comportamento do sistema. A otimização calcula a entrada u que minimiza um custo predefinido, levando em conta as previsões e restrições do sistema. Esse sinal de controle u é aplicado ao "Sistema", que também

é afetado por uma perturbação z , resultando na saída y . O sinal de saída y é realimentado ao controlador MPC, permitindo ajustes contínuos com base na diferença entre w e y , aprimorando a performance do sistema em tempo real.

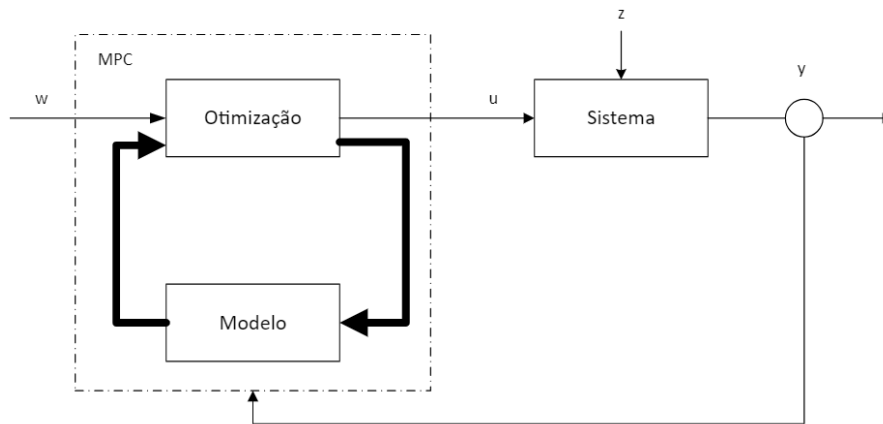


Figura 2.12. Diagrama de blocos simplificado de um circuito de controlo baseado em MPC (adaptado de [58])

Tabela 2.4. Comparação entre os métodos de controlo (adaptado de [59])

| Características | PID | MRAC | LQR | MPC |
|-------------------------------------|----------------|------------|--------------------------|----------------|
| Simplicidade | Simple | Moderado | Complexo | Complexo |
| Implementação | Fácil | Moderado | Difícil | Difícil |
| Modelo do Sistema | Não necessário | Necessário | Necessário | Não necessário |
| Otimização | Local | Global | Global | Local |
| Robustez a Distúrbios | Boa | Moderada | Boa | Variável |
| Desempenho em Sistemas Não Lineares | Limitado | Bom | Excelente | Bom |
| Consideração de Restrições | Não | Sim | Sim | Não |
| Horizonte de Tempo | Curto | Infinito | Definido pelo utilizador | Curto |
| Cálculo Computacional | Baixo | Moderado | Alto | Moderado |
| Convergência | Rápida | Rápida | Variável | Lenta |
| Sensibilidade a Ruído | Baixa | Baixa | Moderada | Alta |

A tabela 2.4 apresenta uma visão geral comparativa dos métodos. As características específicas podem variar de acordo com a implementação e aplicação sendo que é importante analisar os requisitos específicos da sua aplicação e considerar os prós e contras de cada método antes de fazer sua escolha.

Ao comparar os métodos, acabou-se por escolher o MPC por ser o único que permite ao utilizador definir um horizonte de tempo que vai ao encontro do que é necessário nesta dissertação, para que o utilizador consiga definir um dos parâmetros para o funcionamento do algoritmo.

As metodologias de controlo anteriores baseiam-se na identificação de um modelo que represente o sistema a controlar. Em seguida, são apresentados os conceitos genéricos relacionados com a identificação de um modelo que seja representativo do comportamento do processo.

2.5. IDENTIFICAÇÃO DOS SUBSISTEMAS

A identificação dos subsistemas consiste em encontrar o modelo adequado para um subsistema "real". Isto inclui encontrar as entradas adequadas e estimar os valores dos parâmetros adequados que definem o modelo. Se for visto como uma caixa negra, a aplicação do modelo torna-se simples e fornece técnicas de controlo alternativas para identificar o melhor sinal de controlo a aplicar na entrada do subsistema.

O sistema dinâmico apresenta as entradas, que estão normalmente associadas aos sinais externos e podem ser manipuladas pelo observador e, as perturbações que correspondem a sinais que não podem ser medidos. Depois, há as saídas de interesse particular para o observador. A Figura 2.13 ilustra as relações entre as entradas, as perturbações e as saídas de um subsistema dinâmico típico.

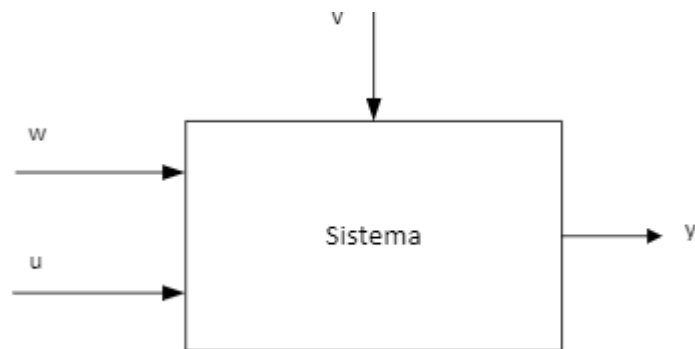


Figura 2.13. Sistema com saída y , entrada(s) u , perturbação medida w e perturbação não medida v (adaptado de [62])

De um modo geral, um subsistema dinâmico é aquele em que o valor de saída futuro $y(t)$, está relacionado com as entradas e perturbações passadas de acordo com uma função não linear $f(y, u)$:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n), u(t-1), u(t-2), \dots, u(t-m), w, v) \quad (2)$$

Onde u , n , m , w e v representam:

- u : Representa as entradas ou sinais de controle aplicados ao sistema. Essas entradas podem ser comandos ou influências externas que afetam diretamente o comportamento do sistema.
- n : É o número de saídas passadas do sistema que influenciam a saída atual $y(t)$. Ou seja, n é a ordem do sistema em relação às suas saídas passadas.
- m : É o número de entradas passadas do sistema que influenciam a saída atual $y(t)$. Ou seja, m é a ordem do sistema em relação às suas entradas passadas.
- w : Representa distúrbios externos ou ruídos que podem afetar o sistema. Esses distúrbios são geralmente imprevisíveis e podem introduzir incerteza ou variabilidade no comportamento do sistema.
- v : Representa parâmetros adicionais ou fatores que influenciam o sistema. Esses parâmetros podem incluir coeficientes específicos, constantes de tempo, ganhos, ou outras características que definem a dinâmica do sistema.

A identificação dos subsistemas pode ser realizada através da análise de registos de séries temporais para cada um dos cenários ou configurações anteriormente mencionadas. Algoritmos adequados ajudam a definir um modelo *Auto-Regressive-Moving-Average* (ARMA), *Auto-Regressive-Integrated - Moving-Average* (ARIMA) ou *Seasonal ARIMA with exogenous inputs* (SARIMA).

A escolha de um modelo autorregressivo depende do compromisso entre a simplicidade do modelo e as propriedades de uma série cronológica. Se uma série temporal for considerada estacionária (ou seja, se as suas propriedades não forem afetadas por uma alteração na fonte temporal), os modelos escolhidos são normalmente o Autorregressivo com Entrada Exógena (ARX) e/ou o Autorregressivo de Média Móvel com Entrada Exógena (ARMAX).

O ARMAX é mais complexo do que o ARX devido ao facto de ter a capacidade de lidar com séries temporais estacionárias cuja regressão de erros é uma combinação linear.

Se uma série temporal for não-estacionária, pode e deve ser utilizado o modelo *Autoregressive Integrated Moving Average with exogenous input* (ARIMAX) e/ou o modelo *Seasonal Autoregressive Integrated Moving Average with exogenous input* (SARIMAX). Ambos os modelos, ARIMAX e SARIMAX, são capazes de lidar com séries estacionárias e

não estacionárias. No entanto, se a série temporal tiver elementos sazonais, a melhor opção seria o SARIMAX.

Em [54], foi feita uma análise para a previsão da procura de carga no contexto das redes inteligentes, utilizando ARX, Redes Neurais Artificiais (RNA) e Redes Neurais Artificiais otimizadas por Algoritmo Genético (RNA-GA). Nesta mesma análise, o ARX apresentou um erro percentual absoluto médio maior, mas um tempo de execução menor, quando comparado com as soluções ANN e ANN-GA.

Em [56], foi desenvolvido um modelo híbrido para prever a procura de eletricidade em função da temperatura exterior. Em seguida, foi comparado com o modelo ARMAX. Apesar do bom desempenho de ambos, o ARMAX apresentou erros de previsão mais elevados do que o modelo híbrido.

No presente trabalho, verificou-se que a modelação do sistema utilizando o ARMAX produziu resultados aceitáveis.

2.6. MÉTODOS DE PREVISÃO

Neste capítulo são descritos os conceitos as equações que representam os diferentes métodos de previsão para séries temporais e por fim algum trabalho realizado com aplicação desses métodos fazendo comparação entre elas.

2.6.1. MÉTODOS CONVENCIONAIS PARA MODELAÇÃO DE SÉRIES TEMPORAIS

Os métodos convencionais para séries temporais referem-se aos modelos autorregressivos que são amplamente utilizados para lidar com as séries temporais [60]. Em termos de métodos convencionais existem ARX, ARMAX, ARIMAX e SARIMAX e a escolha de um dos métodos depende do compromisso entre a simplicidade do método e as propriedades de uma série temporal.

Em [54], foi feito um trabalho com finalidade de analisar a previsão da demanda de carga em *smart grids* utilizando o método ARX, o *Artificial Neural Networks* (ANN) e *Artificial Neural Networks optimized by Genetic Algorithm* (ANN-GA). A conclusão tirada foi que o ARX apresentou menor tempo de execução e valor de *mean absolute percentage error* superior quando comparado com ANN e ANN-GA.

Em [61], foi feita uma análise onde compararam o ARMAX com um modelo híbrido desenvolvido para prever a demanda de eletricidade em função da temperatura exterior e a conclusão tirada foi que o ARMAX apresentou erros de previsão superior ao modelo híbrido.

Em [58], no *Bengal Ocidente* foram utilizados o ARIMA e ARIMAX para prever a demanda de carga. Foram utilizadas com variáveis exógenas as variações climáticas no método ARIMAX e no mesmo trabalho, o ARMAX apresentou valores de Mean Absolute Error inferiores em todos os cenários quando comparado com o ARIMA.

Em [59], foi feita uma comparação entre o SARIMAX, o *gated recurrent neural network* (GRNN) e o *gated convolutional neural network* (GCNN) relativo ao desempenho na previsão do nível de carga nos edifícios. Neste mesmo trabalho, o SARIMAX apresentou menor tempo de execução, mas com menor precisão quando comparado com os outros métodos.

2.6.2. MODELO AUTORREGRESSIVO COM ENTRADA EXÓGENA

O modelo ARX [56] [62] é um modelo de regressão cuja saída atual depende de saídas passadas (variáveis dependentes) e de entradas exógenas (variáveis independentes). O modelo ARX é capaz de lidar com séries temporais estacionárias com características regressivas, ou seja, a saída depende das entradas passadas. Este modelo é representado pelas equações:

$$\boldsymbol{\phi}(L)y(t) = \sum_{i=1}^n \boldsymbol{\beta}_i(L)\mathbf{u}_i(t) + e(t) \quad (3)$$

$$\boldsymbol{\phi}(L) = 1 + \phi_1 L^{-1} + \dots + \phi_p L^{-p} \quad (4)$$

$$\boldsymbol{\beta}_i(L) = \beta_{i1} + \beta_{i2} L^{-1} + \dots + \beta_{ip} L^{-p+1} \quad (5)$$

Onde

- $y(t)$ – Saídas do sistema (por exemplo previsão de consumo);
- $\mathbf{u}_i(t)$ – Variáveis exógenas;
- $e(t)$ – Variável de distribuição do erro;
- $\boldsymbol{\phi}(L)$ – Polinómio associado aos termos de auto-regressão;
- $\boldsymbol{\beta}_i(L)$ – Polinómio associado à entrada exógena i ;
- n – Número de variáveis independentes;
- L – Representa as regressões, i.e., assumindo $p = 1$, significa que $\boldsymbol{\phi}(L)y(t) = \boldsymbol{\phi}(L^{-1})y(t) = \phi_1 y(t - 1)$;
- p – Grau do polinómio da componente auto-regressiva.

Na forma compacta, $\boldsymbol{\phi}(L)$ e $\boldsymbol{\beta}_i(L)$ são representados por:

$$\boldsymbol{\phi}_t = [\phi_1, \phi_2, \dots, \phi_p]^T \quad (6)$$

$$\mathbf{y}_t = [y(t-1), y(t-2), \dots, y(t-p)] \quad (7)$$

$$\boldsymbol{\beta}_t = [\beta_{11}, \beta_{12}, \dots, \beta_{1p}, \beta_{21}, \beta_{22}, \dots, \beta_{2p}, \beta_{n1}, \beta_{n2}, \dots, \beta_{np}]^T \quad (8)$$

$$\mathbf{u}_t = [u_{11}(t), u_{12}(t-1), \dots, u_{1p}(t-p+1), u_{21}(t), u_{22}(t-1), \dots, u_{2p}(t-p+1), u_{n1}(t), u_{n2}(t-1), \dots, u_{np}(t-p+1)]. \quad (9)$$

Em [63], os vetores de coeficientes $\boldsymbol{\phi}$ e $\boldsymbol{\beta}$ são estimados a partir de uma solução onde é utilizado o método *Conditional Sum of Squares* e *Maximum Likelihood Estimation* (CSS-MLE).

2.6.3. MÉDIA MÓVEL AUTORREGRESSIVA COM ENTRADA EXÓGENA

À semelhança do modelo ARX, o ARMAX também inclui a componente de média móvel para além das componentes de auto-regressão com variáveis exógenas. O ARMAX é utilizado quando uma série temporal tem características de regressão e a regressão do erro é uma combinação linear [64][65]. A estrutura do ARMAX é representada pela equação:

$$\boldsymbol{\phi}(L)y(t) = \sum_{i=1}^n \boldsymbol{\beta}_i(L)\mathbf{u}_i(t) + \boldsymbol{\psi}(L)e(t) \quad (10)$$

Com

$$\boldsymbol{\phi}(L) = 1 + \phi_1 L^{-1} + \dots + \phi_p L^{-p} \quad (11)$$

$$\boldsymbol{\beta}_i(L) = \beta_{i1} + \beta_{i2} L^{-1} + \dots + \beta_{ip} L^{-p+1} \quad (12)$$

$$\boldsymbol{\psi}(L) = 1 + \psi_1 L^{-1} + \dots + \psi_q L^{-q} \quad (13)$$

Onde

- $\boldsymbol{\psi}(L)$ – Polinómio de *moving average*;
- q – Grau do polinómio de *moving average*;

Com as seguintes representações vetoriais:

$$\boldsymbol{\phi}_t = [\phi_1, \phi_2, \dots, \phi_p]^T \quad (14)$$

$$\mathbf{y}_t = [y(t-1), y(t-2), \dots, y(t-p)], \quad (15)$$

$$\boldsymbol{\beta}_t = [\beta_{11}, \beta_{12}, \dots, \beta_{1p}, \beta_{21}, \beta_{22}, \dots, \beta_{2p}, \beta_{n1}, \beta_{n2}, \dots, \beta_{np}]^T \quad (16)$$

$$\mathbf{u}_t = [u_{11}(t), u_{12}(t-1), \dots, u_{1p}(t-p+1), u_{21}(t), u_{22}(t-1), \dots, \quad (17)$$

$$\dots, u_{2p}(t-p+1), u_{n1}(t), u_{n2}(t-1), \dots, u_{np}(t-p+1)],$$

$$\boldsymbol{\psi} = [1, \psi_1, \psi_2, \dots, \psi_q], \quad (18)$$

$$\mathbf{e}_t = [e(t), e(t-1), e(t-2), \dots, e(t-q)]. \quad (19)$$

Os coeficientes de $\boldsymbol{\phi}$, $\boldsymbol{\beta}$ e $\boldsymbol{\psi}$ são definidos nas Equações (11), (12) e (13) e estimados a partir do método CSS-MLE [63].

2.6.4. MÉDIA MÓVEL INTEGRADA AUTORREGRESSIVA COM ENTRADA EXÓGENA

O ARIMAX é um método similar ao modelo ARMAX. No entanto, além das características do modelo ARMAX, possui a capacidade de lidar com séries temporárias não estacionárias. Assim sendo, o método é representado pela equação:

$$\boldsymbol{\phi}(L)y(t) = \sum_{i=1}^n \boldsymbol{\beta}_i(L)\mathbf{u}_i(t) + \boldsymbol{\psi}(L)e(t) \quad (20)$$

Com coeficientes de $\boldsymbol{\phi}$, $\boldsymbol{\beta}$ e $\boldsymbol{\psi}$ definidos nas equações

$$\boldsymbol{\phi}(L) = 1 + \phi_1 L^{-1} + \dots + \phi_p L^{-p} \quad (21)$$

$$\boldsymbol{\beta}_i(L) = \beta_{i1} + \beta_{i2} L^{-1} + \dots + \beta_{ip} L^{-p+1} \quad (22)$$

$$\boldsymbol{\psi}(L) = 1 + \psi_1 L^{-1} + \dots + \psi_q L^{-q} \quad (23)$$

e com as seguintes representações vetoriais:

$$\boldsymbol{\phi}_t = [\phi_1, \phi_2, \dots, \phi_p]^T \quad (24)$$

$$\mathbf{y}_t = [y(t-1), y(t-2), \dots, y(t-p)], \quad (25)$$

$$\boldsymbol{\beta}_t = [\beta_{11}, \beta_{12}, \dots, \beta_{1p}, \beta_{21}, \beta_{22}, \dots, \beta_{2p}, \beta_{n1}, \beta_{n2}, \dots, \beta_{np}]^T \quad (26)$$

$$\mathbf{u}_t = [u_{11}(t), u_{12}(t-1), \dots, u_{1p}(t-p+1), u_{21}(t), u_{22}(t-1), \dots, u_{2p}(t-p+1), u_{n1}(t), u_{n2}(t-1), \dots, u_{np}(t-p+1)], \quad (27)$$

$$\boldsymbol{\psi} = [1, \psi_1, \psi_2, \dots, \psi_q], \quad (28)$$

$$\mathbf{e}_t = [e(t), e(t-1), e(t-2), \dots, e(t-q)]. \quad (29)$$

onde

- Δ – Diferença entre um sinal no instante t e $t-1$, i.e, para o caso de $y(t)$: $y(t) - y(t-1) = (1-L)y(t)$;
- d – Ordem do componente da diferenciação.

Os coeficientes são estimados por intermédio de uma solução resultante da combinação do CSS-MLE [63].

2.6.5. MÉDIA MÓVEL INTEGRADA AUTORREGRESSIVA SAZONAL COM ENTRADA EXÓGENA

O SARIMAX é um método que estende o ARIMAX. Além disso, o SARIMAX, ao contrário dos outros métodos de auto-regressão referidos anteriores, possui a capacidade de lidar não somente com séries temporais de carácter estacionário, mas também com séries não estacionárias com elementos sazonais. O método SARIMAX é representado pela equação:

$$\boldsymbol{\phi}(q)\mathbf{M}(L^s)y(t) = \sum_{i=1}^n \boldsymbol{\beta}_i(L)\mathbf{u}_i(t) + \boldsymbol{\psi}(L)\mathbf{O}(L^s)e(t), \quad (30)$$

com

$$\boldsymbol{\phi}(L) = \Delta^d \Delta_s^D 1 + \Delta^d \Delta_s^D \phi_1 L^{-1} + \dots + \Delta^d \Delta_s^D \phi_p L^{-p} \quad (31)$$

$$\boldsymbol{\beta}_i(L) = \beta_{i1} + \beta_{i2} L^{-1} + \dots + \beta_{ip} L^{-p+1} \quad (32)$$

$$\boldsymbol{\psi}(L) = 1 + \psi_1 L^{-1} + \dots + \psi_q L^{-q} \quad (33)$$

$$\mathbf{M}(L^s) = 1 + m_1 L^{-s} + \dots + m_p L^{-Ps} \quad (34)$$

$$\mathbf{O}(L^s) = 1 + o_1 L^{-1} + \dots + o_q L^{-Qs} \quad (35)$$

e com as seguintes representações vetoriais:

$$\boldsymbol{\phi}_t = [\Delta^d \Delta_s^D \phi_1, \Delta^d \Delta_s^D \phi_2, \dots, \Delta^d \Delta_s^D \phi_p]^T \quad (36)$$

$$\mathbf{y}_t = [y(t-1), y(t-2), \dots, y(t-p)], \quad (37)$$

$$\boldsymbol{\beta}_t = [\beta_{11}, \beta_{12}, \dots, \beta_{1p}, \beta_{21}, \beta_{22}, \dots, \beta_{2p}, \beta_{n1}, \beta_{n2}, \dots, \beta_{np}]^T \quad (38)$$

$$\mathbf{u}_t = [u_{11}(t), u_{12}(t-1), \dots, u_{1p}(t-p+1), u_{21}(t), u_{22}(t-1), \dots, u_{2p}(t-p+1), u_{n1}(t), u_{n2}(t-1), \dots, u_{np}(t-p+1)], \quad (39)$$

$$\boldsymbol{\psi} = [1, \psi_1, \psi_2, \dots, \psi_q], \quad (40)$$

$$\mathbf{e}_t = [e(t), e(t-1), e(t-2), \dots, e(t-q)], \quad (41)$$

onde:

- $\mathbf{M}(L^s)$ – Coeficiente sazonal de auto-regressão;
- $\mathbf{O}(L^s)$ – Coeficiente sazonal de média móvel
- Δ_s – Diferença sazonal entre um sinal no instante t e $t-s$, i.e., $y(t) - y(t-s) = (1 - L^s)$;
- s – Duração de sazonalidade, i.e., trimestral $s = 4$ e mensal $s = 12$;
- D – Ordem do integrador sazonal.

No modelo SARIMAX a ordem (p, d, q) obtém-se de: (1) p - representa a ordem dos auto-regressores (AR), (2) d - dos termos de diferença (MA) e (3) q - dos termos de média móvel (ARMA). Os coeficientes de $\boldsymbol{\phi}$, $\boldsymbol{\beta}$, $\boldsymbol{\psi}$, \mathbf{M} e \mathbf{O} são estimados a partir do método CSS-MLE [63].

2.6.6. REDES NEURONAIS

As redes neurais têm sido aplicadas em situações onde um modelo de um sistema é obtido numa representação simplificada do modelo, e a metodologia de treino tipicamente usada

enquadra-se na aprendizagem supervisionada *offline* e *online* [66]. A metodologia de treino supervisionada *offline* envolve a utilização de todo o conjunto de dados disponível de uma só vez. Este método é dividido em várias etapas, em que cada etapa consiste em passar por todo o conjunto de dados para atualizar os pesos da rede neuronal. Por outro lado, no treino supervisionado *online*, o modelo é treinado utilizando uma pequena quantidade de dados de cada vez, com os pesos da rede a ser atualizados após cada etapa [66].

A diferença fundamental entre os dois métodos é que o treino *offline* é geralmente utilizado quando o conjunto de dados é fixo e está disponível na totalidade desde o início. Em contraste, o treino *online* é apropriado para situações em que os dados são recebidos em tempo real ou mudam frequentemente [66].

Aprendizagem é o processo no qual tanto a forma (ou estrutura) como os parâmetros do modelo são estimados. Tipicamente, definir os modelos deste género obriga à recolha de um alargado conjunto de dados (de entrada vs. saída) representativos do comportamento do sistema que se pretende modelar.

Estes dados são então divididos em 3 conjuntos distintos: dados de treino, dados de validação e dados de teste.

Os dados de treino são usados para estimar o modelo num processo que minimiza o erro quadrático absoluto. Quando se considera a validação, os dados de treino são repartidos em 2 grupos, em que uma parcela se refere por dados de validação por representarem valores não são usados para estimar o modelo, e assim evitar *overfitting* do modelo. Aumentando a capacidade em generalizar. É usual recorrer a uma distribuição entre dados para treino e validação na ordem dos (70%-30%) ou então executar validação *k-fold*, onde os dados de validação são selecionados aleatoriamente e o desempenho médio resultante das *k* vezes é tomado em consideração.

A forma do modelo é um problema de otimização global (onde se define a complexidade do modelo) mas a estimação dos respetivos parâmetros é um problema de otimização local e algoritmos *state-of-the-art* (*Backpropagation*, *Levenberg-Marquardt*, etc.) são muito frequentemente aplicados.

A metodologia anterior é identificada por *off-line* pois o conjunto de dados para encontrar o modelo é definido previamente na fase de aprendizagem e a escolha destes é determinante para o desempenho do modelo encontrado [67].

2.6.6.1. REDE PERCEPTRÃO MULTI-CAMADAS

A rede perceptrão multi-camadas, caracteriza-se por possuir diversas camadas, constituídas por elementos – os neurónios. Os neurónios da camada de entrada estão ligados aos neurónios da camada escondida (camada oculta) e, os neurónios das camadas escondidas vão estar ligadas aos correspondentes, na camada de saída conforme ilustrada na figura 2.14. Este género de rede neuronal não possui realimentação porque todas as ligações têm a mesma direção partindo de entrada até chegar na camada de saída. Geralmente, estão completamente ligados entre si (no tipo *feedforward*) e consoante a camada que estão presentes, exercem distintas funções. A primeira camada é designada de camada de entrada, e é esta camada que alimenta a rede neuronal com os sinais exteriores ou sinais de entrada. Os dados de entrada podem ser previamente alvo de pré-processamento. É uma fase que pode ser identificada por reescalonamento e/ou normalização dos sinais exteriores para melhorar a capacidade de aprendizagem da rede e torná-la menos sensível a variações bruscas nos sinais de entrada. A camada escondida corresponde à camada onde quer as ligações entre os neurónios quer as funções que os caracterizam têm por objetivo definir a relação entre os dados de entrada e a saída da rede. Esta é a camada que pode conter uma ou mais subcamadas. O desempenho da rede está muito relacionado com a forma e tamanho desta camada, pois quer a quantidade de neurónios em cada subcamada quer o número de camadas vão determinar a qualidade de aproximação entre a saída da rede e a saída desejada [68].

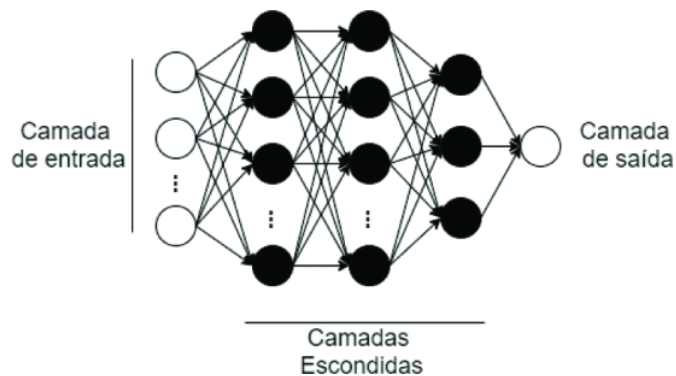


Figura 2.14. Arquitetura de rede Neuronal Perceptron Multicamadas (MLP) ([75])

2.6.6.2. REDE NEURONAL RECORRENTE – MEMÓRIA DE LONGO E CURTO PRAZO

Uma Rede Neural Recorrente com Memória de Longo e Curto Prazo (RNN-LSTM) é um tipo especializado de Rede Neural Recorrente (RNN), projetada para superar as limitações das RNN tradicionais, especialmente no que diz respeito ao problema dos gradientes de fuga, que afeta o aprendizado em sequências longas. As RNN-LSTM são especialmente úteis porque incorporam mecanismos que permitem o armazenamento e a manipulação de informações por períodos prolongados.

Essas redes diferenciam-se das redes neurais MLP-*feedforward* (Perceptron Multicamadas) porque possuem realimentação, o que significa que elas utilizam informações de estados anteriores da rede para influenciar previsões futuras. Isso as torna especialmente adequadas para tarefas como séries temporais, reconhecimento de fala e processamento de linguagem natural, onde a ordem e a dependência temporal dos dados são cruciais.

As RNN-LSTM são constituídas por unidades recorrentes, que por sua vez são compostas por quatro componentes principais:

- ***Input Gate*** (Porta de Entrada): Controla quais informações da entrada atual serão utilizadas para atualizar o estado da célula.
- ***Forget Gate*** (Porta de Esquecimento): Determina quais informações do estado anterior da célula devem ser esquecidas ou mantidas.
- ***Cell Candidate*** (Candidato à Célula): Cria uma nova informação candidata, que pode ser utilizada para atualizar o estado da célula, dependendo da decisão das outras portas.
- ***Output Gate*** (Porta de Saída): Decide quais informações do estado atual da célula serão usadas para produzir a saída da LSTM.

Esses componentes trabalham em conjunto para permitir que a LSTM mantenha e manipule informações de maneira eficiente ao longo do tempo. A estrutura e o funcionamento desses componentes são ilustrados na Figura 2.15, que descreve detalhadamente o fluxo de informações dentro de uma unidade LSTM [69].

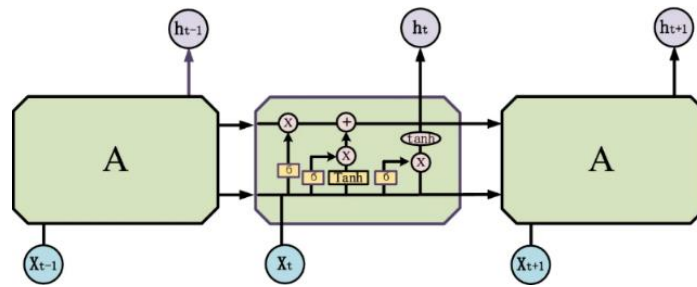


Figura 2.15. Esquemático de fluxo de dados na camada escondida numa rede RNN-LSTM tradicional ([76])

O processo geral de treino envolve ajustar os parâmetros de uma RNN-LSTM para minimizar um erro em relação aos dados de treino, através das seguintes etapas [69]: (1) Recolha e Pré-processamento de Dados, (2) Elaboração da Estrutura da Rede, (3) Seleção da Função de Custo, (4) Optimizador, (5) Treino da Rede e (6) Avaliação e Melhoria. Em seguida descrevemos cada um deles:

(1) Recolha e Pré-processamento de Dados

A recolha e pré-processamento de dados visa reunir um conjunto abrangente e adequado de dados de treino para a tarefa específica que se deseja que a rede execute. Preparar os dados de forma adequada para garantir que estejam formatados corretamente para a rede. Isso pode envolver etapas como normalização, codificação de variáveis categóricas e segmentação de sequências, conforme necessário.

(2) Elaboração da Estrutura da Rede

Esta fase visa decidir quantas camadas LSTM serão usadas na rede. Um maior número de camadas pode resultar em um desempenho superior, mas também aumenta a complexidade do modelo e o tempo necessário para o treino.

Definir o tamanho das camadas LSTM, que se refere ao número de unidades LSTM em cada camada da rede.

Especificar as funções de ativação a serem utilizadas nas camadas LSTM. Algumas funções de ativação comuns para LSTMs incluem Tangente Hiperbólica (*tanh*) e Unidade Linear Retificada (*ReLU*).

(3) Seleção da Função de Custo

Esta fase visa escolher uma métrica da função de custo que quantifica a discrepância entre as previsões feitas pela rede neural e os valores reais esperados. Funções de custo frequentemente usadas em problemas de classificação incluem a entropia cruzada, enquanto para problemas de regressão, é comum utilizar o erro quadrático médio.

(4) Optimizador

Esta fase visa selecionar um método de otimização para ajustar os parâmetros da rede durante o processo de treino. Algoritmos de otimização amplamente utilizados incluem estimativa de momento adaptativo e gradiente descendente estocástico.

(5) Treino da Rede

Esta fase compreende passar repetidamente pelo conjunto de dados de treino, alimentando cada sequência de dados na rede e calculando a função custo (é uma métrica que quantifica o quão bem ou mal um modelo está se saindo em termos de prever os resultados desejados e ela calcula a diferença entre as previsões feitas pela rede neural e os valores reais (ou esperados) para os dados de treinamento correspondente.

Empregar o algoritmo de otimização para ajustar os parâmetros da rede na direção que reduza a função custo observada.

Avaliar o desempenho da rede usando um conjunto de dados de validação separado para prevenir o sobreajuste.

Encerrar o processo de treino quando a função custo começar a estabilizar (convergir) ou quando alcançar um número máximo de iterações pré-definido.

(6) Avaliação e Melhoria

Esta fase compreende o testar do desempenho da rede já treinada usando um conjunto de dados de teste independente. Examinar os resultados obtidos e identificar possíveis áreas de melhoria no modelo, como ajustar a estrutura da rede, a função de custo ou o algoritmo de otimização.

2.7. ALGORITMO GENÉTICOS

Os algoritmos genéticos (AG) são algoritmos de pesquisa baseados nos conceitos de seleção natural e de genética natural proposta por Charles Darwin e desenvolvido por John Holland na década de 1970 [70]. O AG adota o conceito de seleção natural para selecionar potenciais pais de uma população para produzir descendentes de melhor qualidade nas gerações futuras [71] e ela melhoram a população de soluções de forma iterativa, aproximando-se gradualmente de uma solução ótima. É particularmente eficaz quando a função objetivo é estocástica, não diferenciável, descontínua ou altamente não linear. Os AG incluem três operadores fundamentais, a seleção, o cruzamento e a mutação, que determinam a forma como a solução ótima é gerada após cada iteração

[72].

O AG foi desenvolvido para simular alguns dos processos observados na evolução natural, um processo que atua sobre os cromossomas. O AG difere de outros métodos de pesquisa na medida em que procura entre uma população de pontos e trabalha com uma codificação do conjunto de parâmetros, em vez de trabalhar com os valores dos próprios parâmetros. Utiliza também a informação da função objetivo sem qualquer informação sobre o gradiente. O esquema de transição do AG é probabilístico, enquanto os métodos tradicionais utilizam informação sobre o gradiente. Devido a estas características, os algoritmos são utilizados como algoritmos de otimização de uso geral. Também são capazes de pesquisar espaços irregulares e, por conseguinte, são aplicados a uma variedade de aplicações de otimização de funções, estimativa de parâmetros e aprendizagem automática [72].

O AG foi utilizado no trabalho com o objetivo de servir como um otimizador com a finalidade de determinar quais os melhores parâmetros a ser utilizado a nível de controlo num determinado instante futuro.

2.7.1. PRINCÍPIO BÁSICO DO FUNCIONAMENTO DO ALGORITMO GENÉTICO

O princípio de funcionamento de um algoritmo genético padrão e os passos fundamentais são ilustrados na figura 2.16 e figura 2.17. As principais etapas envolvidas são a geração, geração de uma população de soluções, a determinação da função objetivo e da função de operadores genéticos [72].

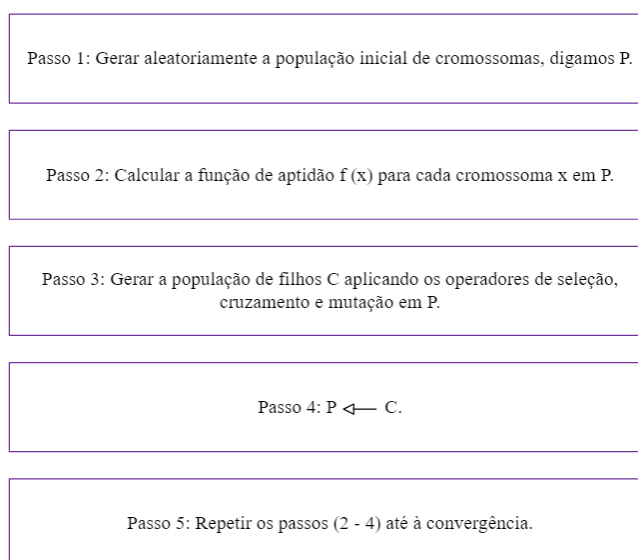


Figura 2.16. Algoritmo genético padrão (adaptado de [74])

A figura 2.16 mostra um AG típico conforme descrito em [73]. Os constituintes comuns dependentes de problemas do AG incluem o seguinte: a codificação e as funções de avaliação. O pesquisador começou com a codificação de problemas como um primeiro passo para traduzir o problema para a linguagem computacional [74]. A natureza do problema a ser resolvido geralmente dita a representação do problema. Existem vários métodos de representação, tais como:

1. **Codificação binária:** Cada cromossoma nesta técnica é representado usando uma sequência de bits 0's e 1's. Um exemplo do uso da codificação binária é o problema da mochila [75].
2. **Codificação de valores:** os cromossomas são representados através de uma sequência de alguns valores, tais como números reais, caracteres ou objetos. Estes valores são possivelmente caracteres, números reais, etc [76].
3. **Codificação em árvore:** Cada cromossoma é tratado como uma árvore de certos itens, tais como comandos ou funções. A codificação em árvore é boa para programas em evolução [76] utilizados na programação genética. A figura 2.17 resume como o AG funciona [77].

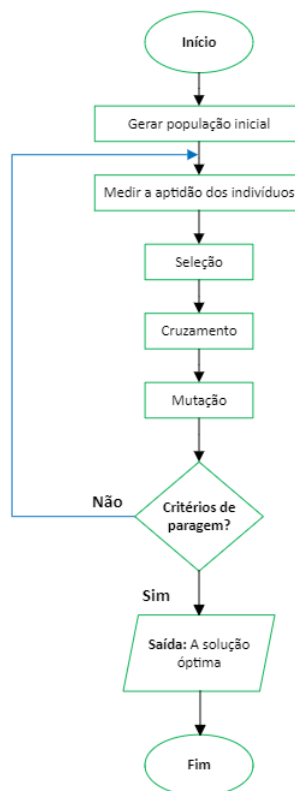


Figura 2.17. Princípio de funcionamento de um algoritmo genético (adaptado de [74])

Em seguida explicam-se os blocos apresentados no fluxograma da Figura 2.17:

-
1. **Início:** Este é o ponto de partida do algoritmo. A execução do AG começa aqui [78].
 2. **Gerar população inicial:** Nesta etapa, o AG cria uma população inicial de indivíduos e esses indivíduos são normalmente representados como cadeias de bits, vetores ou outra estrutura de dados, dependendo do problema. Cada indivíduo representa uma solução potencial para o problema que está a ser resolvido.
 3. **Medir a aptidão dos indivíduos:** a) Depois de gerar a população, é necessário medir a "qualidade" de cada indivíduo. b) Isso é feito através de uma função de aptidão (*fitness*), que avalia cada solução com base em critérios previamente definidos. O objetivo é selecionar os indivíduos mais aptos para gerar a próxima geração [79].
 4. **Seleção:** A seleção escolhe os melhores indivíduos da população com base em sua aptidão. Os indivíduos mais aptos têm uma maior probabilidade de serem selecionados para reprodução [82]. Existem diferentes métodos de seleção, como roleta, torneio, entre outros [83], onde os melhores indivíduos têm mais chances de sobreviver e se reproduzir.
 5. **Cruzamento:** Também conhecido como crossover, é o processo de combinar dois indivíduos (pais) para gerar novos indivíduos (filhos). O cruzamento é feito misturando as características dos pais para criar filhos que herdem atributos dos dois. O objetivo é explorar novas combinações de soluções.
 6. **Mutação:** a) A mutação introduz variações aleatórias nos indivíduos, alterando alguns de seus genes. b) Esse processo é importante para garantir diversidade genética na população e evitar que o AG fique preso em um ótimo local. Normalmente, a mutação ocorre com uma baixa probabilidade.
 7. **Crítérios de paragem?** Aqui, o algoritmo verifica se algum critério de parada foi atingido. Esses critérios podem ser, por exemplo [80]: a) Um número máximo de gerações foi alcançado; b) Uma solução com aptidão suficiente foi encontrada; c) Não há mais melhora significativa entre as gerações. Se o critério de parada for atendido, o algoritmo termina.
 8. **Saída: A solução ótima:** Se o critério de parada foi alcançado, o AG retorna a melhor solução encontrada durante o processo.

Fim: O algoritmo termina e apresenta a solução final, que é a melhor solução encontrada durante a execução do AG.

2.8. CRITÉRIOS E MÉTRICAS

Nesta seção são apresentadas as equações que foram utilizadas como critérios e métricas para avaliar o desempenho dos modelos com vista a representar a dinâmica para cada subsistema.

2.8.1. CRITÉRIOS DE INFORMAÇÃO DE AKAIKE

O Critério de Informação de Akaike (AIC) foi criado por *Hirotsugu Akaike*, um pioneiro na modelação e identificação de modelos estatísticos. Baseado na minimização da informação de *Kullback-Leibler* [81] e proposto por Akaike em 1973, o AIC é um critério que avalia a qualidade do ajuste de um modelo paramétrico, estimado pelo método da máxima verossimilhança e definido pela equação (42):

$$AIC = 2k - 2 \log L(\hat{\theta}) \quad (42)$$

em que $L(\hat{\theta})$ traduz a função de verossimilhança maximizada e k corresponde ao número de parâmetros livres do modelo. O termo $2k$ traduz a penalização, funcionando como uma compensação pelo viés de ajustamento quando se utilizam os estimadores de máxima verossimilhança. O AIC é um critério assintoticamente eficiente. No entanto, sendo uma aproximação assintótica, pode não ter um bom desempenho em amostras pequenas.

2.8.2. CRITÉRIO DE INFORMAÇÃO DE BAYES

O Critério de Informação de *Bayes* (BIC), proposto por *Schwarz* em 1978 [82], é um critério que avalia modelos com base na probabilidade a posteriori. O BIC é definido pela equação (43):

$$BIC = -2 \log L(\hat{\theta}) + k \log (n) \quad (43)$$

em que k corresponde ao número de parâmetros a serem escolhidos, $L(\hat{\theta})$ traduz a função de verossimilhança maximizada e n é o número de observações da amostra. De acordo com

[67], o AIC e o BIC são critérios assintóticos utilizados para comparar modelos aninhados, mas também podem ser aplicados a modelos não aninhados.

2.8.3. CRITÉRIO DE INFORMAÇÃO DE HANNAN-QUINN

O Critério de Informação de *Hannan-Quinn* (HQIC) é uma medida de adequação de um modelo estatístico, sendo frequentemente utilizado como alternativa ao AIC e BIC para selecionar um modelo a partir de um conjunto finito de modelos. Proposto por [86], é definido pela equação (44):

$$HQIC = -2 k \log(\log(n)) - 2 \log L(\hat{\theta}) \quad (44)$$

onde $L(\hat{\theta})$ traduz a função de verossimilhança maximizada, k corresponde ao número de parâmetros, e n é o número de observações da amostra. É importante notar que, ao contrário do AIC, o HQIC e o BIC não se originam da minimização da informação de *Kullback-Leibler*.

A literatura também oferece critérios para a seleção de modelos não paramétricos. Em [83], é fornecida uma visão geral dessa metodologia.

2.8.4. ERRO QUADRÁTICO MÉDIO

O erro médio quadrático (*Mean Squared Error*, MSE) mede a média dos quadrados dos erros - ou seja, a diferença média quadrática entre os valores estimados e o valor real. O erro MSE também pode ser associado a uma função de risco, correspondendo ao valor esperado da perda de erro ao quadrado. O MSE é uma medida da qualidade de um estimador e o facto de o MSE ser quase sempre estritamente positivo deve-se à aleatoriedade ou ao facto de o estimador não ter em conta informações que poderiam produzir uma estimativa mais precisa [84].

Como é derivado do quadrado da distância euclidiana, é sempre um valor positivo que diminui à medida que o erro se aproxima de zero e a mesma é representada pela equação (45).

Considerando um vetor de n previsões, gerado a partir de uma amostra de n pontos de dados em todas as variáveis, o valor de MSE é calculado como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (45)$$

onde \hat{Y}_i traduz o valor previsto, Y_i corresponde ao valor observado e n o número de observações.

2.8.5. SOMA DOS QUADRADOS DOS ERROS

A soma dos quadrados do erro (*Sum of Squared Errors*, SSE) é a soma dos desvios entre os valores previstos e os valores observados [85]. O SSE é representado pela equação (46).

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (46)$$

2.8.6. ERRO MÉDIO RELATIVO

O erro relativo médio (*Mean Relative Error*, MRE) é uma medida de erro frequentemente utilizada na aprendizagem automática para problemas de regressão. O MRE mede o erro relativo médio, ou seja, o quanto as previsões do modelo diferem em percentagem. É também referido como erro percentual absoluto médio (*Mean Absolute Percentage Error*, MAPE).

O MRE é representado pela equação [86]:

$$MRE = \frac{1}{n} \sum_{i=0}^{n-1} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (47)$$

Por sua vez, o MAPE é representado pela equação:

$$MAPE = MRE * 100 \quad (48)$$

2.8.7. RAIZ DO ERRO QUADRÁTICO MÉDIO

A raiz do erro quadrático médio (*Root Mean Square Error*, RMSE) mede a diferença média entre os valores previstos de um modelo estatístico e os valores reais. Também é uma das métricas de avaliação amplamente utilizada e reconhecida na comunidade de *machine learning* para medir o desempenho de modelos de regressão [87]. Para o cálculo do RMSE é utilizada a equação (49) seguinte:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (49)$$

2.8.8. R-QUADRADO (R^2)

O R^2 é uma métrica que indica a proporção da variação da variável dependente explicada pelas variáveis independentes em um modelo de regressão. No contexto de modelagem ARMAX, o R^2 é utilizado para avaliar a qualidade do ajuste do modelo ao conjunto de dados de treino. Ele mede a proporção da variância da variável dependente (geralmente uma série temporal) que é explicada pelas variáveis independentes (entradas exógenas) e os termos autorregressivos e de média móvel. Um R^2 mais próximo de 1 indica que o modelo ARMAX tem bom desempenho ao capturar a dinâmica dos dados. A equação de R^2 é dada por [88]:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} \quad (50)$$

Onde:

\bar{y}_i é a média dos valores observados y_i .

3. TRABALHO REALIZADO

Este capítulo começa por abordar a arquitetura considerada para permitir a integração dos quatros sistemas de aquecimento de água da piscina. Nas seções seguintes são descritos o algoritmo, a plataforma criada e a metodologia de interligação entre o algoritmo e a interface *web*. Por fim serão descritos os passos necessários para que o utilizador tenha acesso à plataforma e realize a gestão pretendida.

3.1. CARACTERIZAÇÃO DO SISTEMA A CONTROLAR

O sistema que se pretende controlar é composto por vários subsistemas térmicos representados na figura 3.1, em concreto:

1. Coletores solares;
2. PCM;
3. Piso coletor;
4. Acumulador geotérmico

A tabela 3.1 refere-se aos sistemas utilizados para captação e armazenamento de calor. Inclui coletores solares térmicos, acumuladores de PCM e geotérmicos, além de um piso coletor. Esses sistemas trabalham juntos para otimizar o uso da energia térmica.

Tabela 3.1. Resumo da caracterização dos subsistemas

| Subsistemas | Função | Dimensões e especificações |
|----------------------------|--|---|
| Coletores solares térmicos | Captação de calor para (1) aquecimento da água da piscina, (2) acumulação no PCM, (3) acumulação no PCGeo | 25 m ² orientados para sul com uma inclinação de 37° |
| Acumulador de PCM | É utilizado para (1) acumulação de calor para uma gama de temperaturas de acordo com um ponto de regulação e um diferencial de temperatura especificados; (2) libertar o calor acumulado quando a temperatura da água da piscina desce abaixo de uma determinada gama de valores predefinida | 3 m ³ |
| Piso coletor | Transferir o calor acumulado pela radiação solar incidente no solo | 60 m ² |
| Acumulador geotérmico | Acumulação de calor em excesso para eventual utilização no aquecimento de divisões, por exemplo | |

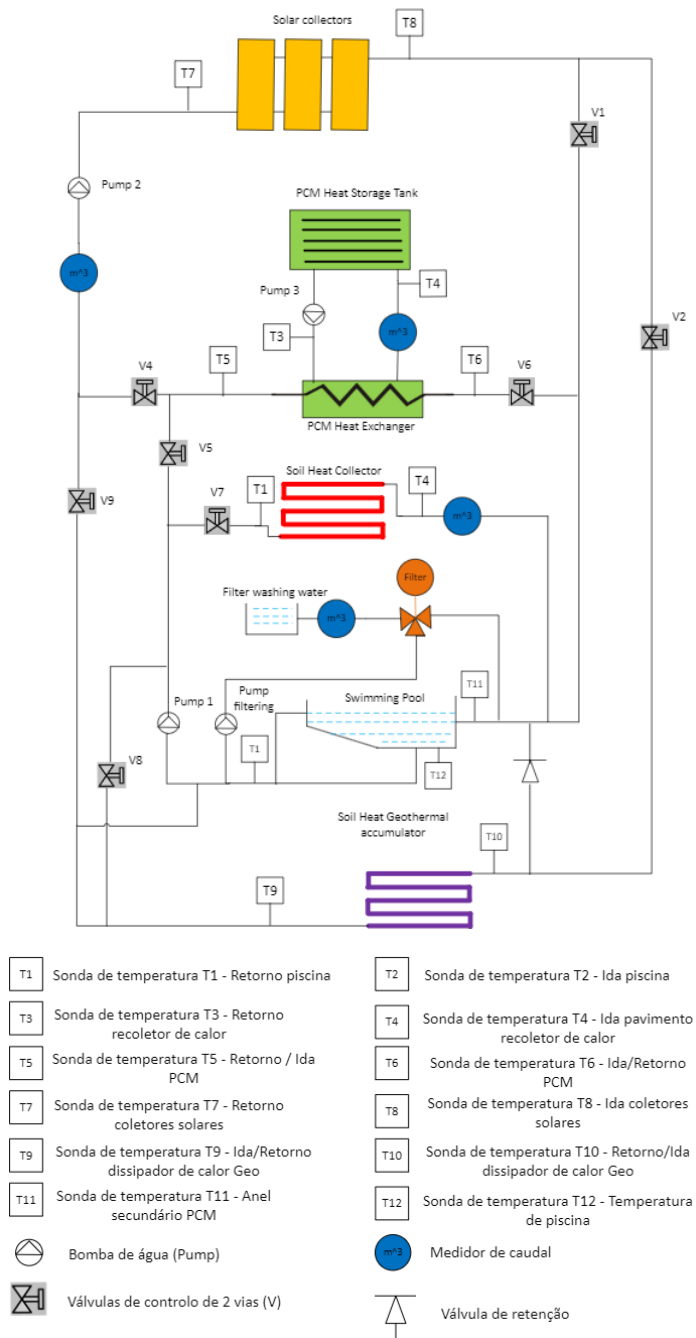


Figura 3.1. Arquitetura da integração dos sistemas de aquecimento

A Tabela 3.2 apresenta os cenários de funcionamento dos quatro subsistemas térmicos utilizados nesta dissertação, detalhando as válvulas e bombas ativadas em cada cenário. Estes cenários de configuração foram programados no PLC para garantir o controlo preciso e eficiente da temperatura da piscina.

Tabela 3.2. Configuração dos cenários de funcionamento, onde: Vi corresponde à válvula i, Bj corresponde à bomba j, A significa aberta, L significa ligada

| Descrição | V | V | V | V | V | V | V | V | V | B | B | B |
|------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 |
| | A | A | A | A | A | A | A | A | A | L | L | L |
| Parado | | | | | | | | | | | | |
| Solar => PCM | x | | | x | | x | | | | | x | x |
| Solar => Piscina | | x | | | | | | | x | | x | |
| PCM => Piscina | | | | | x | x | | | | x | | |
| Solar e PCM => Piscina | | x | | | x | x | | | x | x | x | |
| Piso => Piscina | | | | | | | x | | | x | | |
| Solar e Piso => Piscina | | x | | | | | x | | x | x | x | |
| PCM e Piso => Piscina | | | | | x | x | x | | | x | | |
| Solar, PCM e Piso => Piscina | | x | | | x | x | x | | x | x | x | |

3.2. MODELAÇÃO DOS SUBSISTEMAS

A principal razão para a utilização de um sistema inteligente é a de fornecer uma previsão para um passo de tempo especificado no futuro, em função do subsistema térmico selecionado. Aqui, assumimos que as variáveis que influenciam os parâmetros do modelo são conhecidas (previsíveis e disponíveis) para o horizonte de previsão do esquema de controlo.

Numa fase inicial, o funcionamento do sistema é baseado em dados simulados, utilizando a ferramenta de simulação *TRNSYS* 18.0 [89] que se suporta nos modelos matemáticos para os subsistemas que compõem o sistema global.

Numa fase posterior, para cada subsistema utiliza-se o protótipo desenvolvido. Em ambos os casos, a abordagem à modelação requer a utilização de dados resultantes da amostragem para efetuar a identificação do modelo. O procedimento de identificação do modelo utilizado neste trabalho é dado pelo Algoritmo 1. O seu objetivo é encontrar um modelo ARMAX para cada um dos subsistemas térmicos.

| |
|---|
| <p>Algoritmo 1</p> <p>Entrada: uma matriz dos dados amostrados no passado para a entrada n_{inp} (inclui três variáveis meteorológicas, velocidade do vento, temperatura ambiente, humidade relativa) e os dados amostrados na saída, tamanho dos dados de treino ($Train_size$), tamanho dos dados de teste ($Test_size$)</p> <p>Saída: a melhor representação do sistema de acordo com o seu desempenho na previsão dos dados de teste (Melhor_modelo)</p> |
| <p>Gerar um conjunto de M modelos ARMAX com ordem dos polinómios escolhida aleatoriamente</p> <p>Para cada modelo ARMAX</p> <p>Estimar os parâmetros polinomiais utilizando o estimador de máxima semelhança de vizinhança (MLE)</p> <p>Calcular a soma dos erros quadrados (SSE) e o R^2 nos dados de treino.</p> <p>Calcular os valores de previsão sobre os dados de teste</p> <p>Calcular o valor SSE para os valores previstos (SSE_predict)</p> <p>Melhor modelo = o modelo com o menor SSE_predict</p> <p>Devolve. Melhor modelo</p> |

3.3. CONTROLO ON-OFF

A abordagem típica de controlo para a regulação da temperatura da água da piscina baseia-se em metodologias de controlo clássicas em que os permutadores de calor, tais como bombas ou caldeiras, são ligados ou desligados. Em [16] é utilizada uma caldeira a gás. Este tipo de ação pode ser classificado como esforço garantido. A temperatura da água de saída da caldeira

(*Bout*) é regulada e considerada como podendo ser atingida instantaneamente (dentro de um passo de controlo de duração pré-definida). Este cenário é representado na Figura 3.2, à esquerda.

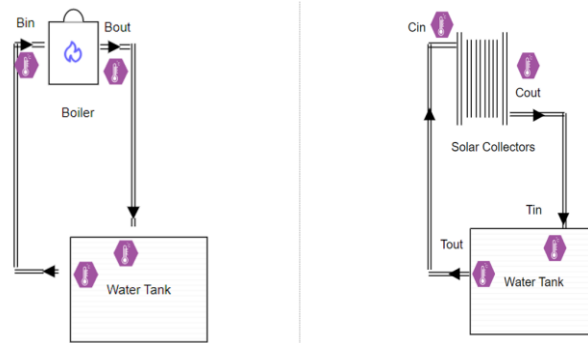


Figura 3.2. O cenário considerado para a regulação da temperatura da água. Esquerda: a caldeira é utilizada como elemento de aquecimento. Direita: um coletor solar é utilizado como elemento de aquecimento

A Figura 3.3 mostra a regra de controlo implementada para a regulação da temperatura quando a componente inteligente da plataforma não está funcional ou ativa. Esta metodologia deverá estar associada a uma circunstância de sistema em modo segurança, ou *fail-safe* e não será a situação usada correntemente. Trata-se por isso de uma situação alternativa ao controlo inteligente proposto neste trabalho. Neste caso, o princípio de controlo enquadra-se no controlo ON/OFF, com histerese, em que é estabelecido um ΔT em torno de um valor de referência ou desejado, $T_{setpoint}$. Assim, quando a temperatura da água observada (decrecente) atinge um valor abaixo de $T_{setpoint} - \Delta T/2$, o sinal de controlo passa para ON, o que traduz o ligar das electroválvulas. No caso em que a temperatura observada (ascendente) atinge um valor acima de $T_{setpoint} + \Delta T/2$ o sinal de controlo OFF traduz o desligar das electroválvulas. Quando a temperatura observada atinge o valor de $(T_{setpoint} + \Delta T/2)$ e a temperatura da água à saída dos coletores solares é superior à da água do tanque, é realizada a circulação da água pelo subsistema geotérmico, de modo que este retenha a energia em excesso.

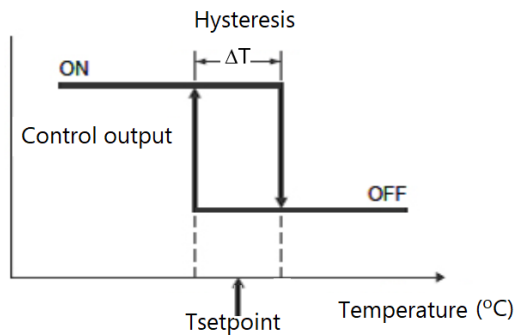


Figura 3.3. Princípio de controlo ON-OFF

Na figura 3.4 ilustra-se a simulação do sistema com controlo (With Control) e sem controlo (No Control), no cenário Solar e Geo -> Piscina, que integra os coletores solares com área de 50 m² e o tanque da piscina.

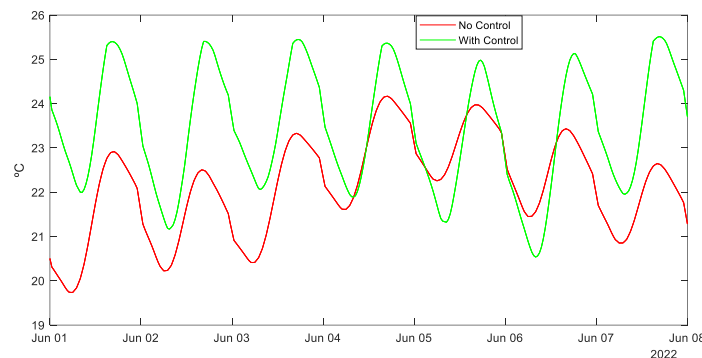


Figura 3.4. Modelação numérica para os meses de maio e junho, comparando as situações com e sem controlo, considerando um valor desejado de temperatura de 25°C e $\Delta T=2^{\circ}\text{C}$

O tipo de serviço suportado pela caldeira é comparado com o suportado por um coletor solar de 50m², representado na Figura 3.2 à direita. O tipo de serviço prestado por esta caldeira pode ser considerado como esforço garantido, uma vez que garante que a temperatura alvo possa ser atingida, embora dependa de condições climáticas que não são controláveis.

O núcleo do sistema de controlo baseia-se num sistema de gestão de energia (SGE), com um controlador preditivo de modelos, que é responsável por encontrar o valor ótimo para a temperatura de entrada da água na piscina (sinal de controlo) e de acordo com os valores previstos para as variáveis meteorológicas. Neste sentido, o sistema tem em consideração as preferências do utilizador ao estimar o conjunto necessário de temperaturas da água de saída da caldeira, que conduzirá a temperatura da água da piscina ao ponto de regulação desejado. Este cenário é conseguido com a ajuda de um otimizador, utilizando o AG.

3.3.1. CONTROLO PREDITIVO BASEADO EM MODELO

Em cenários onde as especificações definidas são cumpridas, a aplicação do controlo preditivo permite antecipar a ação de controlo, considerando a correta identificação prévia do sistema e, conseqüentemente, a capacidade do modelo para prever o comportamento do sistema em momentos futuros. No trabalho realizado, foi utilizado o MPC, que se destaca por sua abordagem na resolução de problemas de controlo, otimizando a sequência de entradas ao longo de um horizonte de predição, enquanto satisfaz as restrições do sistema. Essa metodologia, que recorre ao conceito de otimização para determinar o valor mais eficaz da entrada de controlo u a ser aplicada ao sistema, é ilustrada na Figura 3.5.

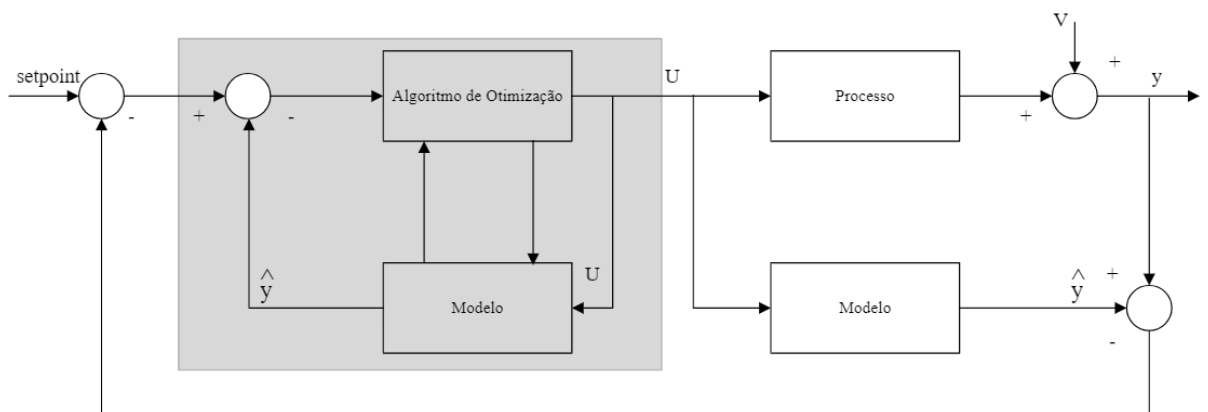


Figura 3.5. Diagrama de fluxo para Controlo Preditivo de Modelos

A figura 3.5 apresentada ilustra o funcionamento do MPC. Abaixo, é explicado o papel de cada bloco e como eles interagem entre si:

- **Algoritmo de Otimização:**

Este bloco é responsável por tomar as decisões de controlo. Ele recebe como entrada a diferença entre o valor de referência (*setpoint*) e a saída prevista pelo modelo (\hat{y}). Com base nessa diferença, o algoritmo de otimização calcula a melhor ação de controle U a ser aplicada ao sistema. No contexto do trabalho, o algoritmo de otimização utiliza vários modelos estimados pelo método ARMAX, que representam o comportamento dos diferentes subsistemas térmicos. A otimização ocorre com base na previsão desses modelos, selecionando a configuração do controlo que melhor atende aos critérios de desempenho, como minimização de erros e cumprimento de restrições.

- **Modelo (dentro do Algoritmo de Otimização):**

Este bloco contém vários modelos estimados pelo ARMAX, cada um dos quais captura o comportamento dinâmico dos diferentes subsistemas térmicos. Esses modelos são usados

pelo algoritmo de otimização para prever a saída do sistema (\hat{y}) para diferentes valores de U . A previsão feita por esses modelos é crucial para a etapa de otimização, pois permite ao controlador antecipar o comportamento futuro do sistema e, assim, calcular a melhor ação de controle.

- **Processo:**

Este é o sistema real ou planta que está a ser controlado. A entrada de controle U , determinada pelo algoritmo de otimização, é aplicada ao processo. A saída real do processo é denotada por y , que é então comparada com a saída prevista pelo modelo (\hat{y}) e com o valor de referência para ajustar as futuras ações do controle.

- **Modelo (fora do Algoritmo de Otimização):**

Este bloco utiliza o mesmo modelo ARMAX para gerar uma previsão da saída do processo (\hat{y}). Essa previsão é comparada com a saída real y para avaliar a precisão do modelo. A diferença entre y e (\hat{y}) também é usada para ajustar e refinar o modelo ao longo do tempo, garantindo que ele permaneça representativo do sistema real.

- **Somador e Comparador:**

Esses blocos são utilizados para calcular as diferenças entre os valores de referência e as saídas previstas/medidas. Essa diferença, ou erro, é uma informação crítica para o algoritmo de otimização, pois define o quão longe o sistema está do comportamento desejado, e orienta as ações do controle subsequentes.

Resumindo, o bloco de modelo dentro do algoritmo de otimização contém vários modelos estimados pelo ARMAX que representam o comportamento de cada subsistema térmico. O algoritmo de otimização utiliza esses modelos para calcular e aplicar a melhor configuração do controle ao processo, garantindo que o sistema opere de forma eficiente e dentro dos limites estabelecidos

Como se mostra na figura 3.6, e utilizando os dados de N_e amostras passadas, é efetuada uma estimativa do melhor modelo. O modelo é então utilizado para fornecer uma sequência de sinais do controle ao sistema para o horizonte de previsão, ou N_p passos futuros.

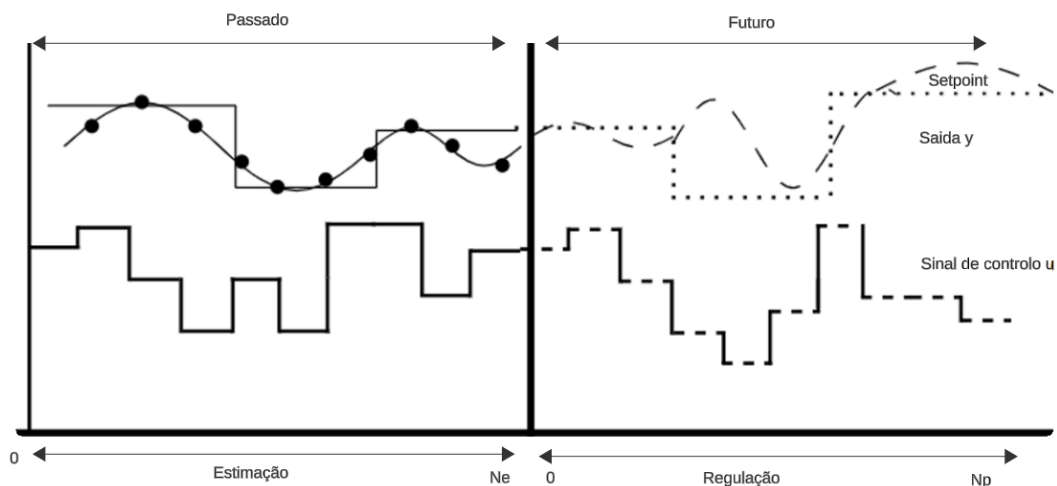


Figura 3.6. Resposta do sistema baseada no controlo preditivo de modelos generalizado

Utilizando o optimizador (baseado na evolução e na seleção natural, por exemplo), é possível determinar uma solução entre um conjunto de alternativas que melhor satisfaçam as restrições impostas e, no final, selecionar a melhor.

3.3.2. ALGORITMO GENÉTICO PARA CONTROLO DO SUBSISTEMA DE CALDEIRA

O algoritmo genético foi utilizado para determinar o conjunto mais adequado de temperaturas futuras da água na saída da caldeira, ao longo de um horizonte de previsão previamente definido.

Neste caso, a codificação do cromossoma no algoritmo genético consiste numa lista de genes de valor real em que o i -ésimo elemento de posição codifica o valor esperado da temperatura de saída da caldeira no i -ésimo passo de tempo futuro, para uma gama pré-definida de valores possíveis. Por exemplo, para um horizonte de previsão de 4 passos, e valores possíveis no intervalo $[4.0, 33.0]$, o i -ésimo cromossoma individual poderia ser traduzido por:

| | | | |
|-----|------|------|------|
| 4.0 | 15.0 | 23.4 | 33.0 |
|-----|------|------|------|

Como *crossover* foi utilizado um operador uniforme que utiliza um único ponto. A probabilidade de mutação foi baseada numa distribuição gaussiana. Em resumo, foram utilizados por omissão os operadores de cruzamento e de mutação.

O cruzamento consiste em combinar genes de dois progenitores para gerar descendentes. Por exemplo, supondo que temos dois indivíduos, $[4.0, 15.0]$ e $[23.4, 33.0]$, que serão

cruzados, escolhe-se aleatoriamente um ponto de cruzamento entre os genes. Admitindo que esse ponto seria entre o primeiro e o segundo gene, após o cruzamento, poderíamos obter descendentes como [4.0, 33.0] e [23.4, 15.0].

A mutação implementada consistiu em selecionar aleatoriamente um gene e alterar seu valor utilizando uma distribuição gaussiana. Por exemplo, se desejamos mutar o terceiro gene, cujo valor é 23,4, escolhemos um valor aleatório a partir de uma distribuição gaussiana com média (μ) e desvio padrão (σ). Suponhamos que o valor aleatório obtido seja $\delta = 0,5$. O novo valor do gene após a mutação será $23,4 + \delta = 23,4 + 0,5 = 23,9$.

A distribuição gaussiana determina a probabilidade de mutação e a magnitude das alterações nos genes. A média da distribuição pode ser ajustada para controlar a direção da mutação (positiva ou negativa), enquanto o desvio padrão regula a intensidade da mutação. Valores próximos à média são mais prováveis de serem escolhidos, enquanto valores mais distantes são menos prováveis.

A mutação adiciona diversidade genética à população, possibilitando a exploração de novas soluções. O cruzamento combina as características vantajosas de diferentes indivíduos, visando aprimorar a qualidade das soluções ao longo das gerações.

Em resumo, ao aplicar mutação e cruzamento com base em uma distribuição gaussiana nos genes [4.0, 15.0, 23.4, 33.0], estamos a utilizar métodos eficazes para explorar e melhorar soluções em algoritmos genéticos, aproveitando a aleatoriedade controlada da distribuição gaussiana para otimização e adaptação genética. O critério de terminação utilizado foi o número de gerações.

3.4. ESTRATÉGIA DE CONTROLO INTELIGENTE PROPOSTA

Dada a natureza e complexidade do sistema, a abordagem proposta é baseada na teoria de controlo preditivo (ver Capítulo 2 [51]), no qual é necessário conter definido um modelo que represente o funcionamento do sistema. Sabendo-se que o sistema no seu todo tem diferentes combinações de configurações (que resultam em diferentes subsistemas térmicos a contribuir para a regulação da água do tanque da piscina), o algoritmo a implementar foi desenhado tendo por base 2 fases:

- Fase 1 – Modelação dos subsistemas;
- Fase 2 – Aplicação do otimizador genético.

A estratégia de controlo global é resumida pelo algoritmo 2:

| |
|---|
| Algoritmo 2: Estratégia de Controlo Preditivo O objetivo é maximizar a utilização da piscina de acordo com as especificações fornecidas pelos utilizadores |
| Entradas: Especificações do subsistema do utilizador (U_SPECS), variáveis meteorológicas previstas (W_VAR), horizonte de previsão (P_HOR), horizonte de controlo (C_HOR), modelos de subsistemas térmicos ($MODELS$) Resultado: a melhor combinação dos subsistemas térmicos ao longo do horizonte de previsão P_HOR , o menor tempo estimado de cumprimento das especificações do utilizador |
| <ol style="list-style-type: none">1. Modelação do subsistema2. Otimização |

Os passos relativos ao ponto 1 do algoritmo 2, são resumidos pelo algoritmo 3:

| |
|---|
| Algoritmo 3: Modelação do sistema O objetivo é encontrar um modelo ARMAX para cada um dos subsistemas térmicos Entrada: uma matriz dos dados amostrados no passado para a entrada n_inp (inclui três variáveis meteorológicas, velocidade do vento, temperatura ambiente, humidade relativa) e os dados amostrados na saída, tamanho dos dados de treino ($Train_size$), tamanho dos dados de teste ($Test_size$) Resultado: a melhor representação do sistema de acordo com o seu desempenho na previsão dos dados de teste (Melhor_modelo) |
| Gerar um conjunto de N modelos ARMAX com ordem dos polinómios escolhida aleatoriamente Para cada modelo ARMAX Estimar os parâmetros polinomiais utilizando o estimador de máxima semelhança de vizinhança (MLE) Calcular a soma dos erros quadrados (SSE) nos dados de treino ($SSE_training$). Calcular os valores de previsão sobre os dados de teste Calcular o valor SSE para os valores previstos ($SSE_predict$) Melhor modelo = o modelo com o menor $SSE_predict$ |

Os passos relativos ao ponto 2 do algoritmo 2, são resumidos pelo algoritmo 4:

| |
|---|
| Algoritmo 4: Optimizador GA ($GAOpt$). O objetivo é encontrar a melhor combinação dos subsistemas disponíveis em relação a determinadas especificações Entrada: o conjunto de modelos de sistemas térmicos, ponto de regulação requerido, variáveis meteorológicas previstas, horizonte de previsão (P_HOR), Max_Gen , p_mut , $cross_rate$ Saída: a melhor combinação dos subsistemas térmicos ao longo do horizonte de previsão P_HOR |
| Enquanto a produção for inferior a Max_Gen ou o critério de paragem não for cumprido Criar uma população com indivíduos ou cenários POP , sendo cada cenário uma combinação distinta do conjunto de subsistemas térmicos. Para o i -ésimo cenário no POP Para a j -ésima configuração no i -ésimo cenário Enquanto o tempo t for inferior ao tempo atual mais P_HOR Estimar a produção y do sistema para o instante t , utilizando os modelos do sistema na j -ésima configuração Avaliar o cenário, utilizando um critério de desempenho baseado na produção prevista Classificar os cenários POP Formar descendência através da recombinação dos pais selecionados, considerando a taxa de cruzamento, $cross_rate$ Aplicar um operador de mutação genética com probabilidade p_mut geração = geração + 1 Classificar os cenários POP Obter o cenário com o melhor critério de desempenho |

Dado que as fontes de origem renovável são caracterizadas pela sua incerteza e não garantem disponibilidade imediata dos sinais de contro necessários, neste trabalho foi adotada uma abordagem alternativa ao contro preditivo generalizado clássico, caracterizado por alguns inconvenientes, em particular: (1) cada um dos subsistemas apresenta respostas lentas de entrada-saída, e (2) gama limitada de temperatura de entrada de água na piscina (ou seja, embora seja possível regular o caudal de água, não é possível regular a temperatura da água). A este respeito, o contro preditivo generalizado da figura 3.5 é a base para uma segunda fase do esquema de contro proposto que se assemelha a um problema de programação de tarefas. Dito isto, cada subsistema térmico será identificado com uma resposta particular (ou uma configuração de contro) e o objetivo é encontrar o melhor intervalo de tempo para colocar cada uma das configurações.

Neste contexto, um algoritmo genético é utilizado como um otimizador para ajudar a determinar a melhor sequência temporal de cenários de contro para um horizonte de previsão pré-definido. A melhor sequência é encontrada se a saída do sistema der a resposta mais próxima da temperatura desejada para a água da piscina, definida como ponto de regulação.

A figura 3.7 mostra o diagrama temporal ilustrativo desta metodologia.

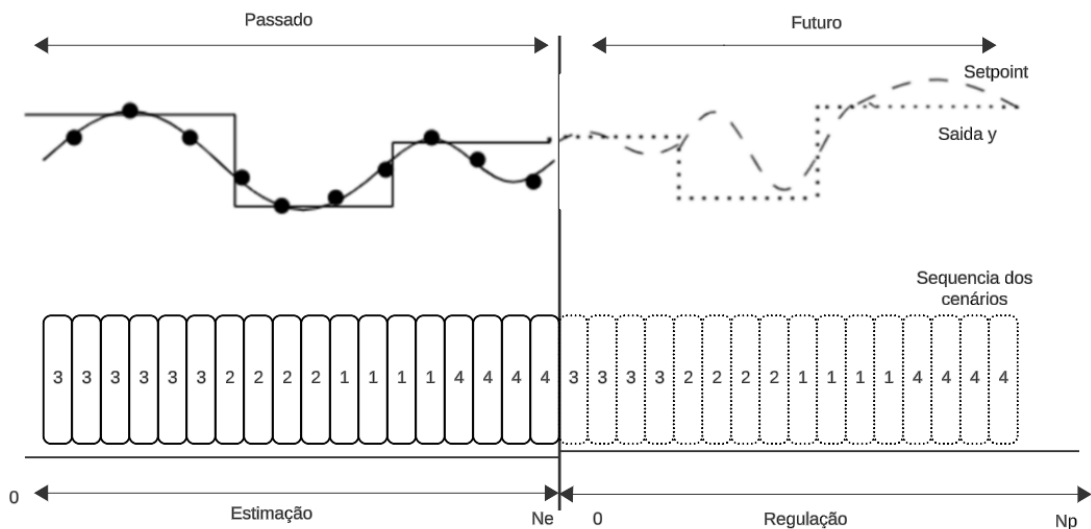


Figura 3.7. Diagrama temporal da abordagem proposta

Nas condições acima referidas, a estratégia de contro a adotar deve considerar a modelação de cada um dos subsistemas que constituem o sistema global, para cada um dos cenários de contro.

Assim, as fontes renováveis integradas no sistema não permitem a disponibilidade imediata dos sinais de contro necessários, pelo que deve ser adotada uma abordagem alternativa.

Assim, do ponto de vista do controlo ótimo, o sistema apresenta um inconveniente: não só cada um dos subsistemas tem uma inércia elevada (ou seja, respostas lentas de entrada-saída), como também não pode fornecer imediatamente a temperatura de entrada de água necessária na piscina (a regulação do fluxo de água é possível, mas não a regulação da temperatura da água), porque dependem muito das condições meteorológicas.

3.5. PLATAFORMA INTELIGENTE

A figura 3.8 ilustra a interligação entre os componentes que compõem o sistema que permite o controlo da temperatura da água no tanque da piscina. A plataforma inteligente é composta por vários elementos. Na Internet encontram-se a plataforma Internet das Coisas (IoT), suportada no *emoncms*, e o servidor da *OpenWeather*® [90].

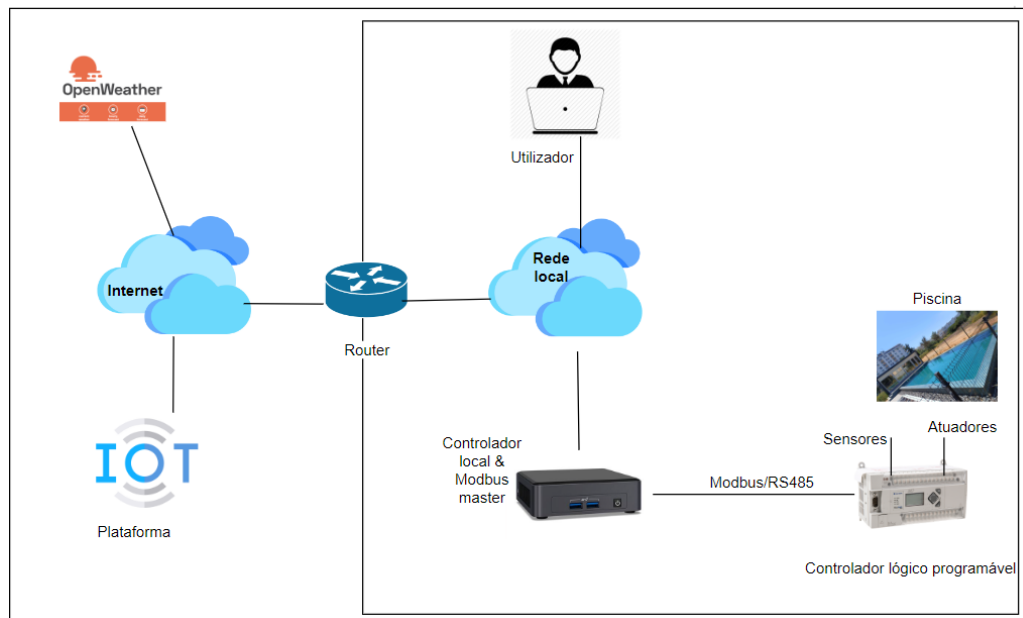


Figura 3.8. Arquitetura global do sistema desenvolvido

Na rede *Internet Protocol* (IP) interna, encontra-se o sistema de controlo inteligente (ou otimizador) que executa como uma aplicação independente, tomando as decisões relativas às etapas de controlo realizadas. O otimizador requer a obtenção de variáveis meteorológicas de previsão em períodos de uma hora, no horizonte de previsão. Essas previsões são recolhidas por meio de solicitações *Web API* ao servidor da *OpenWeather*® (localizado em <https://openweathermap.org/api>). Os dados devolvidos são: a humidade relativa, a velocidade do vento e a temperatura ambiente, sendo usados como entradas do modelo para estimação da temperatura da água do tanque da piscina.

Na área técnica, o controlador local atua como uma unidade central que também é configurada como *Modbus master* e é responsável por: (1) enviar os dados das medições obtidas dos sensores e sistemas da piscina para a plataforma *emoncms*, (2) resolver solicitações feitas pelo utilizador via interface *web*, incluindo controlo de acesso, personalização dos parâmetros do sistema, e (3) acesso aos sensores e atuadores para leitura periódica de valores referentes, por exemplo, o caudal da água, temperatura da água, pressão da água e, nível de água nas diferentes partes do sistema.

O controlador local é executado em *Python* e usa o protocolo HTTP como uma interface uniforme para lidar com recursos da *web* entre diferentes plataformas.

A figura 3.9 complementa a figura 3.8 anterior pois explicita as componentes existentes e a sua interligação

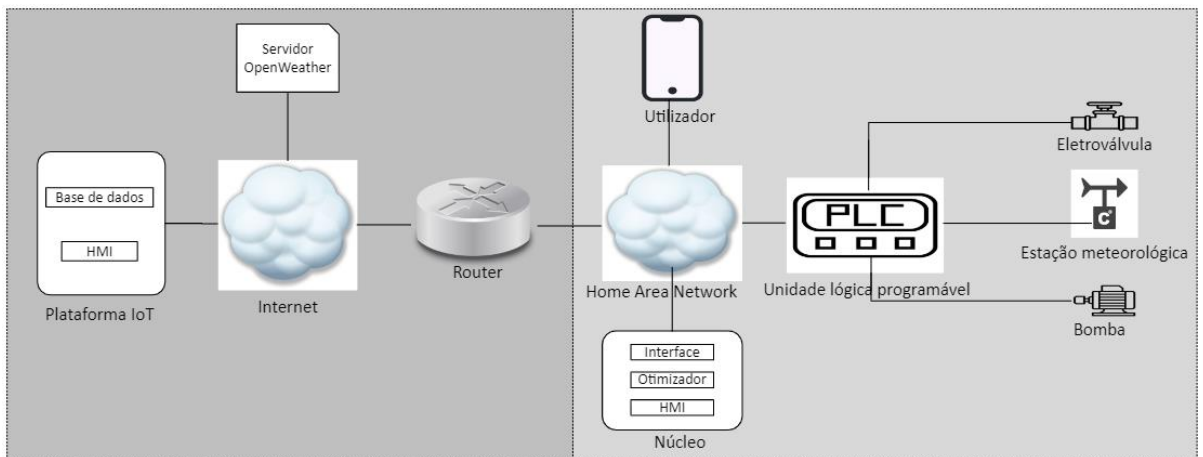


Figura 3.9. Diagrama relacional entre as entidades constituintes da plataforma inteligente

Como forma de interação com o utilizador, foi desenvolvida uma interface baseada: (1) na arquitetura *REST* em linguagem *Python* versão 3.9.5 [91], na vertente consola/ecrã e (2) uma interface *HMI* na vertente *web*. O diagrama de sequência da Figura 3.10, resume a interação entre o cliente e a plataforma IoT no âmbito da *WebApp* desenvolvida.

A tabela 3.3 faz uma descrição mais detalhada sobre as funcionalidades de cada uma das componentes.

Tabela 3.3. Entidades e resumo de funcionalidades

| | |
|--------------------------------------|---|
| Servidor Openweathermap | Disponibiliza informação de dados atmosféricos atuais e respetivas previsões futuras. Dados obtidos com intervalos de uma hora. Serviço online disponibilizado pelo <i>openweathermap</i> |
| Otimizador | Responsável pela implementação do algoritmo de controlo e supervisão do sistema. Estabelece comunicação bidirecional com a plataforma IoT e servidor online (dados meteorológicos) |
| Interface Homem-Máquina (HMI) | Interface gráfica que permite ao utilizador visualizar e alterar parâmetros, via Webservice ou consola (ecrã local). Funcionalidades implementadas incluem solicitação de acesso ao sistema, interação com diversos <i>widjets</i> e a comunicação com a Plataforma IoT para aceder a dados históricos. |
| Plataforma IoT | Serviço responsável pelo armazenamento e gestão do acesso aos dados. Permite ainda a configuração de dashboards e gráficos para a consulta de históricos dos dados armazenados. Acesso aos dados é realizado via API. |
| Interface | Unidade central localizada próximo do sistema de aquisição de dados PLC. Componente responsável pela disponibilização do acesso à internet. É também responsável pela atualização de dados na plataforma IoT, e pela resolução de pedidos realizados pela interface HMI, nomeadamente o controlo de acessos e respetiva autenticação. |
| Base de dados (BD) | Armazena dados recolhidos do sistema tais como a informação recolhida pela unidade de sensorização (PLC) e os dados atmosféricos devolvidos pela estação meteorológica |
| Estação Meteorológica | Responsável pela recolha e armazenamento temporário de dados atmosféricos locais |
| PLC | Permite a leitura de valores periódicos referentes a por exemplo, caudal, temperatura, pressão, e nível da água em diversos locais do sistema. Permite a escrita de parâmetros definidos pelo utilizador tais como setpoint, diferencial de temperatura, etc. É responsável pela consulta de dados da estação meteorológica, consulta e ajuste dos parâmetros das unidades de sensorização e atuação integradas no sistema de bombagem e sistemas térmicos. Contém código de controlo básico do sistema. |

A figura 3.10 apresenta o fluxograma da interface homem-máquina, que descreve as interações entre o utilizador e a plataforma IoT. Este fluxograma ilustra como o utilizador pode interagir com diferentes funcionalidades da plataforma para monitorar e controlar os dados da piscina.

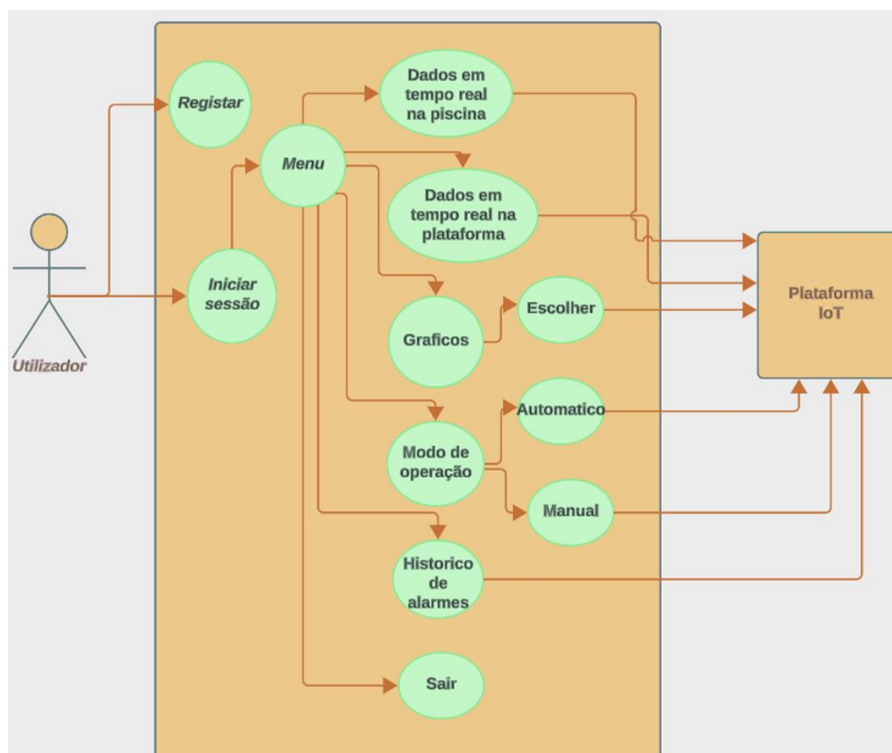


Figura 3.10. Fluxograma da interface homem máquina

Este estilo de arquitetura é simplificado na figura 3.11. O principal motivo da escolha desta arquitetura prende-se com a sua simplicidade, escalabilidade e ausência de estado [92]

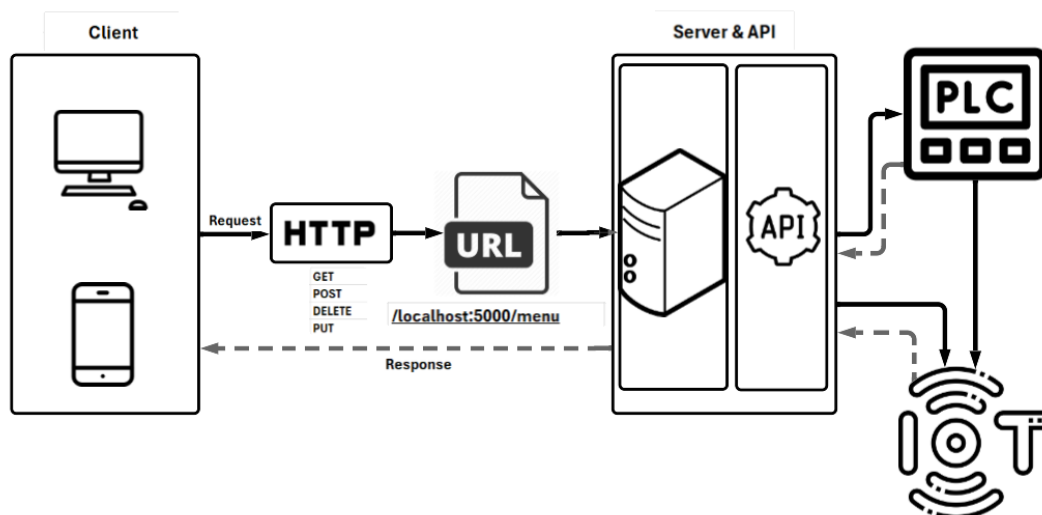


Figura 3.11. Fluxograma da aplicação inteligente baseado na arquitetura REST

3.6. DESENVOLVIMENTO DE API RESTFUL PARA GESTÃO DE PISCINAS COM INTEGRAÇÃO EM FLASK, HTML E CSS

A interface desenvolvida permite ao utilizador gerir a piscina utilizando uma API RESTful. Este tipo de API suporta métodos HTTP como *POST*, *GET*, *PUT*, *DELETE*, *HEAD*, *OPTIONS*

e *TRACE* [93]. API significa Interface de Programação de Aplicações (*Application Programming Interface*). Nesse contexto, o termo "Aplicação" refere-se a qualquer software que desempenhe uma função específica. A interface atua como um acordo de serviço entre dois programas, estabelecendo as regras para a comunicação entre eles por meio de requisições e respostas.

No caso particular da API desenvolvida, apenas se utilizaram os métodos HTTP POST e GET. As operações suportadas incluem a monitorização do funcionamento da piscina, a alteração de parâmetros operacionais e a criação de novos utilizadores.

O método *HTTP GET* é utilizado para obter informações sobre os recursos existentes e monitorizar o funcionamento da piscina. A título de exemplo refira-se o *GET/read_emoncms/status*, que devolve o status operacional da piscina, incluindo parâmetros como a temperatura da água, o *setpoint* e os subsistemas atualmente em uso para aquecimento da piscina.

O método *HTTP POST* é utilizado para criar recursos na API, como adicionar novos utilizadores e alterar parâmetros dos subsistemas. Como por exemplo:

- *POST /users*: Permite a criação de novos utilizadores na piscina.
- *POST /write_plc*: Este endpoint permite ao utilizador definir o *setpoint* desejado para o funcionamento da piscina, bem como alterar parâmetros dos subsistemas, como seleccionar o cenário de funcionamento a ser utilizado para o aquecimento da piscina.

Resumindo as operações *GET* e *POST*:

- Monitorização: o HMI utiliza o método *GET*, para obter informações atualizadas sobre a operação da piscina, como temperatura, *setpoint* e os subsistemas em uso.
- Criação e Alteração: Utilizando o método *POST*, é possível criar utilizadores, definir novos *setpoints* e alterar parâmetros operacionais dos subsistemas da piscina.

Com estes métodos, a API *Restful* proporciona uma forma eficiente e estruturada de gerir a piscina, garantindo que as operações de monitorização e configuração sejam realizadas de maneira consistente e segura.

A API foi desenvolvida utilizando uma combinação de *Python*, HTML e CSS para fornecer uma interface *web* interativa e funcional para os utilizadores. Aqui está uma visão geral de como cada tecnologia foi utilizada:

(1) *Python (Flask)*

- *Backend*: O *Flask* é utilizado para criar o *backend* da aplicação. Ele gere as rotas HTTP

, processa as solicitações dos utilizadores, autênticas sessões, e interage com o *hardware* do PLC e com APIs externas, como a plataforma *emoncms* utilizado neste trabalho.

- Autenticação: A autenticação de utilizadores é gerida de forma a garantir que apenas utilizadores registados e autenticados possam aceder a certas funcionalidades. O processo de criação de utilizadores é gerido e os dados são armazenados num ficheiro em formato *pickle*, que atua como uma base de dados leve. Este ficheiro armazena as informações dos utilizadores registados, como nome de utilizador, e-mail e palavra-passe. A API verifica as credenciais armazenadas nesse ficheiro sempre que um utilizador tenta fazer login, permitindo o acesso seguro a funcionalidades restritas.
- Processamento de Dados: Lógica de negócios, como leitura e escrita de dados no PLC, execução de algoritmos genéticos para modo automático, e leitura de dados históricos de alarmes.

(2) *HyperText Markup Language (HTML)*

- Estrutura das Páginas: HTML é a linguagem usada para estruturar e organizar conteúdo em páginas web. Foi utilizado HTML para desenvolver a Interface Homem-Máquina (HMI), criando a base visual e funcional da interface.
- Formulários: o HTML também é usado para criar formulários que permitem aos utilizadores inserir dados, como *setpoints* para o PLC, credenciais de *login*, e informações de registo.

(3) *Cascading Style Sheets (CSS)*

- Estilização: o CSS é usado para estilizar os elementos HTML, tornando a interface do utilizador visualmente atraente e fácil de usar.
- Responsividade: A utilização de CSS garante que as páginas web sejam responsivas e possam ser acedidas de diferentes dispositivos, como computadores, *tablets* e *smartphones*.

A combinação do *Python*, HTML e CSS permite criar uma aplicação *web* robusta, interativa e amigável para a gestão de piscinas. O *Python*, através do *Flask*, gere a lógica do servidor e a interação com o hardware, enquanto HTML e CSS cuidam da apresentação e da experiência do utilizador.

4. RESULTADOS

Neste capítulo, serão apresentados e analisados os resultados das simulações, assim como a parte referente ao HMI, dividindo-se em três secções: 1) Resultados utilizando o *MATLAB*, 2) Resultados utilizando o *Python* e 3) Plataforma Web. A Secção 1 apresenta os resultados comparativos das diversas metodologias estudadas: Rede Neuronal Perceptron Multicamadas (MLP), Rede RNN-Long Short-Term Memory (RNN-LSTM), Média Móvel Autoregressiva com Entrada Exógena (ARMAX), Controlo MPC Generalizado utilizando o Algoritmo Genético (Caso 1) e a Proposta de Controlo utilizando o Algoritmo Genético (Caso 2). A Secção 2 está subdividida nas seguintes subsecções: Média Móvel Integrada Autoregressiva Sazonal com Entrada Exógena ARIMAX e Proposta de Controlo utilizando o Algoritmo Genético. A Secção 3 apresenta um resumo de funcionalidades da plataforma Web e está subdividida em 2 subsecções: Monitorização em Tempo Real e, Alteração dos Parâmetros dos Subsistemas através da Plataforma *Web*.

4.1. PREPARAÇÃO E DIVISÃO DE DADOS PARA TREINO E TESTE UTILIZANDO O *MATLAB*

Nesta seção, é abordada a preparação e divisão de dados para treino e teste utilizando o *MATLAB*, destacando as implicações dos resultados alcançados a partir das simulações realizadas. Para garantir a precisão dos resultados, foram utilizados dados de amostragem com um intervalo de 6 minutos, gerados pela simulação numérica TRNSYS da piscina. Esses dados foram organizados em conjuntos de treino e teste, com o objetivo de treinar e validar o modelo desenvolvido. Além disso, foram incorporados registos meteorológicos, como temperatura ambiente, velocidade do vento e humidade relativa, para enriquecer o conjunto de dados de treino.

As figuras 4.1 e 4.2 apresentam os dados utilizados para o treino da rede neural (entre o dia 01 de março às 09:00 e até ao dia 14 de março às 00:00) e para o teste da rede (do dia 14 de março às 09:00 até 19 de março às 21:00) utilizando uma amostragem e representação hora a hora.

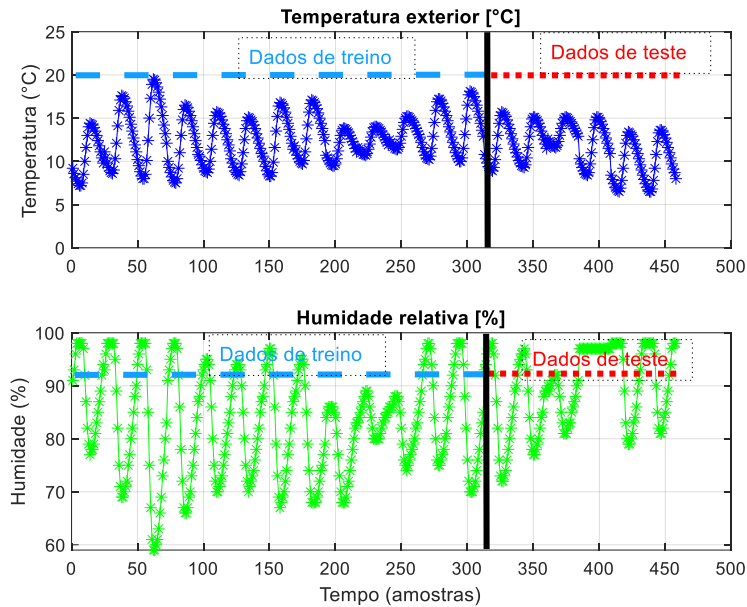


Figura 4.1. Divisão dos dados de temperatura exterior ambiente e humidade nos conjuntos de treino e teste

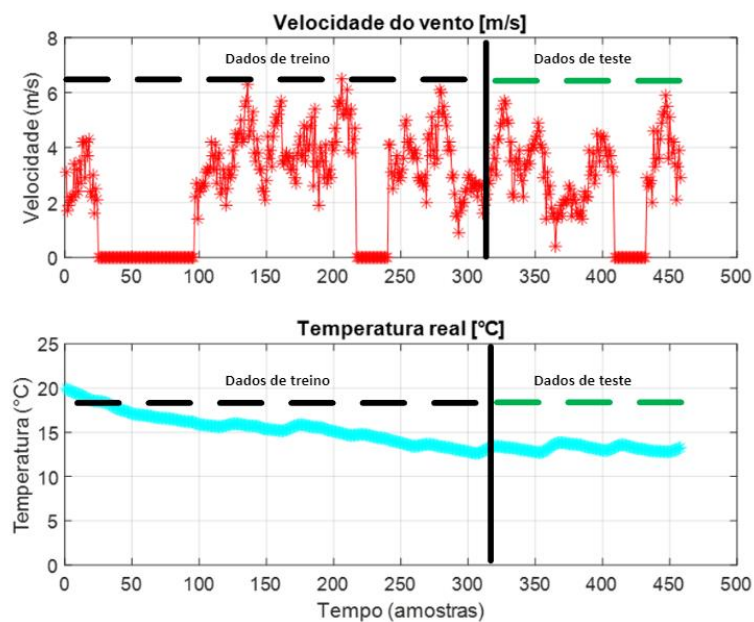


Figura 4.2. Divisão dos dados de temperatura real da água do tanque e velocidade do vento nos conjuntos de treino e teste

O desempenho dos modelos foi avaliado através de várias métricas. As métricas foram o RMSE e o MRE. A primeira métrica foi também utilizada pelos algoritmos de pesquisa durante a estimação do modelo, enquanto a segunda métrica é um valor chave para comparar o desempenho entre soluções.

4.1.1. REDE PERCEPTRON MULTI - CAMADAS

Como primeiro cenário de teste recorreu-se à rede neuronal MLP cuja finalidade foi fazer o mapeamento do sistema, fornecidas os dados de entrada e de saída.

Em MATLAB, o treino com base em redes MLP requer que se indique a quantidade de camadas escondidas e neurónios por cada camada. A quantidade de neurónios na camada de entrada e de saída depende do problema em causa. A figura 4.3 apresenta um exemplo de estrutura de rede MLP para uma estrutura de MLP com 4 entradas (*input*), n neurónios na única camada oculta existente, e uma saída (*output*).

Sendo o objetivo a obtenção do melhor modelo para o comportamento do sistema da piscina, os testes consistiriam em alterar as quantidades de neurónios para uma rede MLP com uma camada oculta. Para comparar as diversas soluções, registaram-se os valores dos erros médios entre os dados provenientes da simulação em TRNSYS e a temperatura prevista pela rede MLP (ver apêndice A.1 que contém o código desenvolvido para as simulações das figuras 4.1, 4.2, 4.3 e as tabelas 4.6, 4.7 e 4.8).

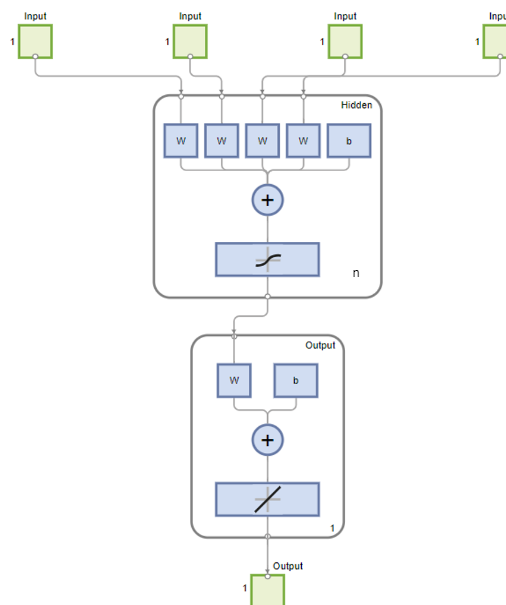


Figura 4.3. Estrutura do modelo Perceptron Multicamadas, onde n representa o número de neurónios, w representa o peso e b representa os biases

A Tabela 4.1 apresenta um resumo das simulações que investigam o impacto do número de neurónios na precisão do modelo. A comparação entre as diferentes parametrizações é avaliada segundo os critérios RMSE e MRE [%].

Tabela 4.1. Resultado de simulação variando o número de neurónios n entre 5 e 90, em intervalos de 5

| Quantidade de neurónios (n) | RMSE | MRE [%] |
|-----------------------------|-------------|-------------|
| 5 | 0,34 | 1,60 |
| 10 | 0,34 | 1,57 |
| 15 | 0,32 | 1,50 |
| 20 | 0,33 | 1,55 |
| 25 | 0,32 | 1,49 |
| 30 | 0,32 | 1,48 |
| 35 | 0,32 | 1,49 |
| 40 | 0,30 | 1,39 |
| 45 | 0,31 | 1,46 |
| 50 | 0,31 | 1,45 |
| 55 | 0,32 | 1,48 |
| 60 | 0,31 | 1,47 |
| 65 | 0,31 | 1,44 |
| 70 | 0,30 | 1,41 |
| 75 | 0,30 | 1,40 |
| 80 | 0,29 | 1,35 |
| 85 | 0,29 | 1,37 |
| 90 | 0,30 | 1,39 |

Ao realizar simulações com diferentes valores de neurónios, concluiu-se que, no intervalo de 5 a 90 neurónios, a configuração com 80 neurónios resultou no menor erro, com um valor de RMSE de 0,29 e um MRE [%] de 1,35. A figura 4.4 ilustra a dinâmica do sistema para 80 neurónios fazendo a comparação entre a curva da temperatura simulada alvo (proveniente da simulação em TRNSYS) e temperatura prevista (ou real) resultante do treino da rede MLP.

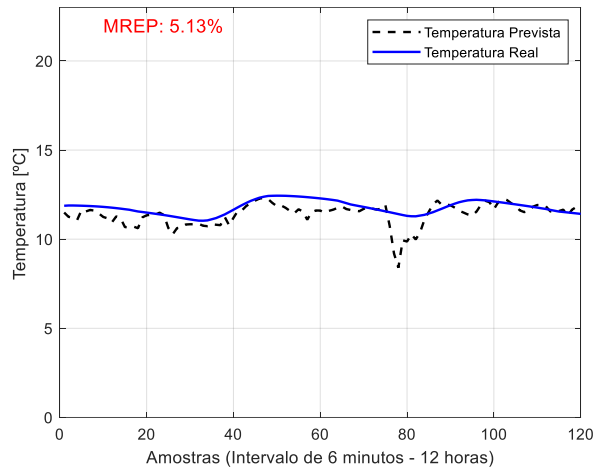


Figura 4.4. Comparação entre a temperatura prevista pelo modelo e a temperatura real do sistema, obtida através de simulação no TRNSYS utilizando rede MLP

Os resultados obtidos demonstram que a metodologia aplicada foi eficaz no treino e teste do modelo. As simulações mostraram uma correspondência consistente entre os dados reais e os resultados previstos pelo modelo. No entanto, é relevante reconhecer as limitações associadas à generalização dos resultados, considerando que os dados utilizados são específicos de um período e de condições meteorológicas particulares. Comparações com estudos semelhantes indicam que os resultados estão alinhados com a literatura existente, embora exista espaço para aperfeiçoamentos.

4.1.2. REDE RNN-LONG-SHORT-TERM MEMORY

No caso da rede RNN-LSTM estimaram-se os parâmetros da rede, nomeadamente o número de camadas ocultas, o número de épocas (ou iterações), o número de camadas e o tamanho dos *minibatches*, com base no erro médio relativo (ver apêndice A.2, que inclui o código desenvolvido). Foram testados diferentes valores para esses parâmetros, tendo o desempenho sido avaliado de acordo com os resultados ilustrados na Tabela 4.2

Tabela 4.2. Resultado da simulação encontrado para diferentes parâmetros da rede

| MiniBatchSize | Número de épocas | Quantidade de neurônios nas camadas de entrada | Quantidade de neurônios nas camadas ocultas | MRE [%] – Treino | MRE [%] -Teste |
|---------------|------------------|--|---|------------------|----------------|
| 10 | 20 | 50 | 100 | 19,11 | 11,72 |
| 15 | | | | 27,82 | 14,74 |
| 20 | | | | 16,86 | 12,62 |
| 25 | | | | 23,78 | 13,73 |
| 30 | | | | 21,12 | 6,99 |
| 35 | | | | 33,11 | 12,47 |
| 40 | | | | 30,53 | 9,45 |
| ... | | | | | |
| 10 | 100 | 100 | 100 | 9,05 | 4,16 |
| 15 | | | | 9,17 | 7,60 |
| 20 | | | | 7,77 | 1,11 |
| 25 | | | | 15,52 | 7,11 |
| 30 | | | | 17,62 | 10,03 |
| 35 | | | | 6,36 | 6,02 |
| 40 | | | | 11,33 | 5,84 |
| ... | | | | | |
| 10 | 100 | 100 | 200 | 15,79 | 2,61 |
| 15 | | | | 6,31 | 10,29 |
| 20 | | | | 4,39 | 4,20 |
| 25 | | | | 7,95 | 18,55 |
| 30 | | | | 6,76 | 4,79 |
| 35 | | | | 7,60 | 2,74 |
| 40 | | | | 8,54 | 8,75 |

Ao examinar os resultados apresentados na tabela 4.2, ficou claro que a melhor configuração para a rede LSTM, dentro dos parâmetros definidos, foi:

- *MiniBatchSize* = 20;
- Números de épocas = 100;
- Quantidade de neurónios nas camadas de entrada = 100;
- Quantidade de neurónios nas camadas ocultas = 200.

Com esta configuração, o erro mais baixo obtido durante o treino da rede foi de 4,39%, enquanto para o teste o erro foi de 4,20%.

Com a melhor configuração identificada anteriormente, seguidamente é apresentada a comparação entre a resposta da rede LSTM e a temperatura real (de entrada obtida do TRNSYS) conforme ilustrado na Figura 4.5.

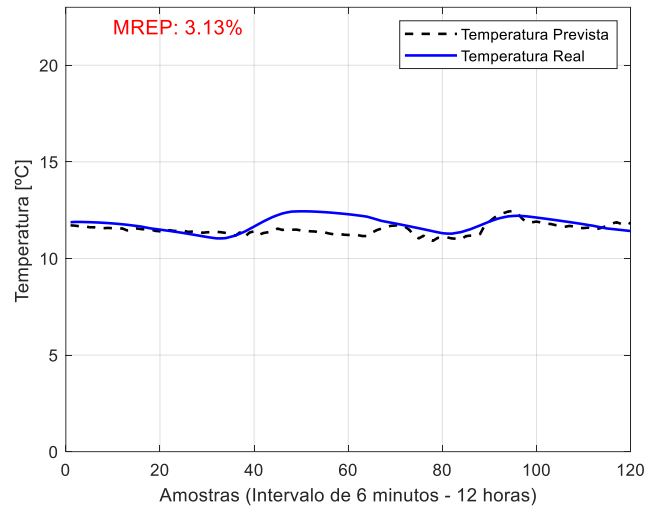


Figura 4.5. Comparação entre a temperatura prevista pelo modelo e a temperatura real do sistema, obtida através de simulação no TRNSYS. A curva vermelha indica o MRE entre as duas temperaturas utilizando rede LSTM

4.1.3. MÉDIA MÓVEL AUTORREGRESSIVA COM ENTRADA EXÓGENA (ARMAX)

Foi desenvolvido o código que se encontra no apêndice A.3, baseado no modelo ARMAX, que permite encontrar os modelos que melhor representam a dinâmica/comportamento para cada subsistema: Piscina, Coletores solares, Pavimentos e Geotermia, numerados de 1 a 4 de acordo com a tabela 4.3.

Tabela 4.3. Subsistema predefinido e respetiva numeração

| Cenários | Sistemas |
|----------|---|
| 1 | Piscina isolada |
| 2 | Piscina com efeito do coletor solar |
| 3 | Piscina com efeito do pavimento |
| 4 | Piscina com efeito do coletor com cobertura |

Para encontrar os modelos que melhor representam a dinâmica de cada sistema, tendo dados simulados de um ano civil completo e resultantes da simulação do software TRNSYS, foram escolhidos os meses de fevereiro, março e julho. Para cada um desses períodos, procuraram-

se os modelos que melhor permitem efetuar a previsão da dinâmica de cada sistema correspondente. Adicionalmente, considerou-se que as variáveis ambientais como a velocidade do vento, temperatura exterior e humidade relativa seriam as entradas exógenas para cada subsistema. Como saída, considerou-se a temperatura prevista para a água, ou real.

Após a simulação, foi identificada a melhor dinâmica para cada sistema, como mostrado nas tabelas 4.4, 4.5 e 4.6, respetivamente para os meses de fevereiro, março e julho.

Nessas mesmas tabelas, o comportamento do sistema é aproximado por um modelo cuja complexidade é determinada pelo grau dos três polinomiais que definem este tipo de modelo e que estão associados aos instantes de tempo passado ou regressores. Como referido na seção 2.6.3, as polinomiais ϕ , β , e ψ distinguem-se de acordo com a sua função:

1) Auto-regressão (ϕ):

Esta polinomial representa a influência dos valores passados da variável do sistema (como por exemplo a temperatura da piscina) sobre o seu estado atual.

2) Entradas exógenas (β):

β representa um conjunto de polinomiais que são em quantidade igual ao número de entradas exógenas.

3) Componente adicional (Ψ):

Ψ é outra polinomial que pode representar um ruído constante, um parâmetro fixo ou algum fator de perturbação no sistema.

Na tabela seguinte a primeira coluna, define a parametrização (grau dos polinómios e respetiva complexidade) de um modelo ARMAX. Esta pode ser resumida por $\{p, [n_1, n_2, \dots, n_p], q\}$, considerando p o grau da polinomial ϕ , n_i o grau do i -ésimo polinómio de β , i.e., β_i e q o grau da polinomial Ψ . $i=1, \dots, n$, em que n é representa a quantidade de variáveis de entrada.

Em concreto, um modelo ARMAX definido pelos parâmetros $\{1, [6,9,3],4\}$ indica que há uma componente regressiva na qual a variável a estimar (ou saída) tem uma dependência do passado recente, i.e., requer o acesso aos dois últimos instantes de amostragem da variável de saída.

Os coeficientes presentes em $[6, 9, 3]$ indicam o grau das polinomiais associadas às três variáveis exógenas utilizadas, que em concreto no sistema da piscina, poderia representar a sensibilidade do modelo a três fatores externos ou exógenos:

- a) Velocidade do vento: Com um coeficiente de 6, o que sugere uma influência moderada, pois tem uma dependência do passado, até ao 6º anterior instante de amostragem.

- b) Humidade relativa: Com um coeficiente mais alto de 9, o que indica que a humidade tem uma influência maior no sistema, pois tem uma dependência do passado, até ao 9º anterior instante de amostragem.
- c) Temperatura exterior: Com um coeficiente de 3, que indica que esta variável tem uma influência moderada, semelhante à da velocidade do vento, pois tem uma dependência do passado, até ao 3º anterior instante de amostragem.

O coeficiente 4 indica o grau da polinomial de Ψ e que estará associada à capacidade do modelo em eliminar o ruído presente, e, portanto, não está associado diretamente a nenhuma das variáveis de entrada selecionadas.

As restantes colunas das tabelas 4.4, 4.5 e 4.6 (SSE (Dados de Treino), SSE (Dados de Teste), e MRE (Dados de Treino)) representam o valor do erro registado durante a fase de modelação do sistema para cada subsistema.

Tabela 4.4. Modelos encontrados em fevereiro para vários subsistemas

| Modelo | Grau polinomiais $\{\phi, \beta i, \psi\}$ | SSE (Dados de treino) | SSE (Dados de teste) | MRE (Dados de treino) |
|-------------------|--|-----------------------|-----------------------|-----------------------|
| Piscina | {10, [6 8 10],3} | $9,67 \times 10^{-1}$ | $8,53 \times 10^{-1}$ | $8,53 \times 10^{-1}$ |
| Coletores solares | {5, [5 8 9],2} | $1,08 \times 10^{-2}$ | $7,85 \times 10^{-1}$ | $7,85 \times 10^{-1}$ |
| Pavimentos | {2, [4 1 6],4} | $1,98 \times 10^{-2}$ | $1,58 \times 10^{-1}$ | $1,58 \times 10^{-1}$ |
| Geotermia | {10, [2 8 8],6} | $2,29 \times 10^{-1}$ | $3,63 \times 10^{-1}$ | $3,63 \times 10^{-1}$ |

Tabela 4.5. Modelos encontrados em março para vários subsistemas

| Modelos | Grau polinomiais $\{\phi, \beta i, \psi\}$ | SSE (Dados de treino) | SSE (Dados de teste) | MRE (Dados de treino) |
|-------------------|--|-----------------------|-----------------------|-----------------------|
| Piscina | {2, [6 9 6],4} | $2,42 \times 10^{-3}$ | $1,67 \times 10^{-1}$ | $3,75 \times 10^{-1}$ |
| Coletores solares | {3, [4 5 5],3} | $1,55 \times 10^{-1}$ | $5,69 \times 10^{-1}$ | $6,12 \times 10^{-1}$ |
| Pavimentos | {10, [1 4 7],10} | $6,35 \times 10^{-3}$ | $5,12 \times 10^{-1}$ | $9,87 \times 10^{-1}$ |

| | | | | |
|-----------|-----------------|-----------------------|-----------------------|-----------------------|
| Geotermia | {7, [10 3 9],1} | $6,63 \times 10^{-1}$ | $1,59 \times 10^{-1}$ | $7,72 \times 10^{-1}$ |
|-----------|-----------------|-----------------------|-----------------------|-----------------------|

Tabela 4.6. Modelos encontrados em julho para vários subsistemas

| Modelo | Grau polinomiais $\{\phi, \beta_i, \psi\}$ | SSE (Dados de treino) | SSE (Dados de teste) | MRE (Dados de treino) |
|-------------------|--|-----------------------|-----------------------|-----------------------|
| Piscina | {9, [8 4 2],4} | $7,11 \times 10^{-2}$ | $1,79 \times 10^{-1}$ | $1,79 \times 10^{-1}$ |
| Coletores solares | {1, [2 4 4],7} | $9,49 \times 10^{-1}$ | $5,26 \times 10^{-1}$ | $5,26 \times 10^{-1}$ |
| Pavimentos | {9, [9 3 3],4} | $3,82 \times 10^{-1}$ | $4,07 \times 10^{-1}$ | $4,07 \times 10^{-1}$ |
| Geotermia | {4, [8 1 4],9} | $1,78 \times 10^{-2}$ | $8,92 \times 10^{-1}$ | $8,92 \times 10^{-1}$ |

4.1.3.1. MÉDIA MÓVEL AUTORREGRESSIVA COM ENTRADA EXÓGENA PARA VÁRIOS TAMANHOS DAS JANELAS DE DADOS

A finalidade desta simulação foi comparar o modelo ARMAX para diferentes janelas de dados de treino, e a estrutura do modelo foi obtida gerando aleatoriamente as diferentes combinações entre graus das polinomiais. Os objetivos passam por encontrar o melhor modelo em função do tamanho da janela. Considerando que cada amostra está disponível a cada 6 minutos e para poder ter uma representação temporal adequada, diferentes tamanhos de janela foram especificados. Assim, usaram-se janelas com 1920, 320, 160 e 80 amostras de instantes de tempo. A comparação entre os modelos foi realizada tendo em conta a capacidade de previsão para 30 passos futuros, ou seja, 3 horas. Em seguida são mostradas as figuras 4.6 e 4.7 resultantes da previsão obtida pelo modelo para diferentes tamanhos de janela, iniciada pelas 04:48 do dia 17 de março de 2022.

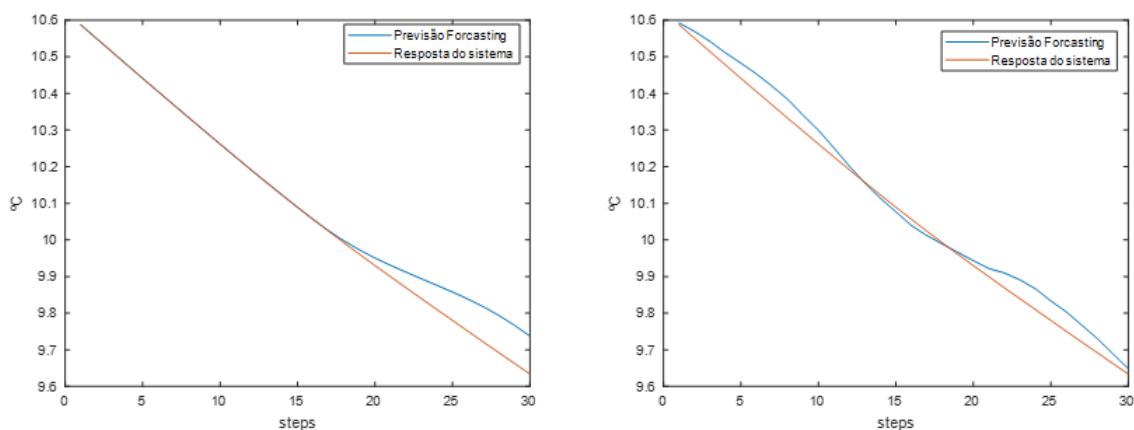


Figura 4.6. Previsão utilizando janelas de 80 dados à esquerda e de 160 à direita

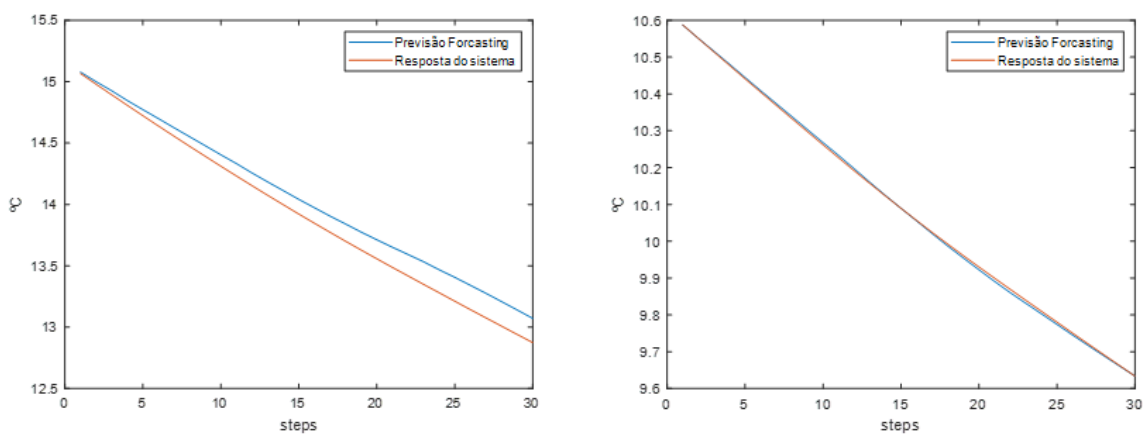


Figura 4.7. Previsão utilizando janelas de 320 dados à esquerda e de 1920 à direita

A Tabela 4.7 fornece informações importantes sobre o impacto do tamanho do conjunto de dados de treino no desempenho de diferentes modelos. A análise da tabela sugere que os modelos treinados em conjuntos de dados maiores geralmente são mais precisos do que os modelos treinados em conjuntos de dados menores. No entanto, a escolha do tamanho ideal do conjunto de dados de treino depende de vários fatores e deve ser feita com base numa análise cuidadosa dos dados e do modelo específicos.

Tabela 4.7. Métricas de desempenho dos modelos para diferentes tamanhos das janelas

| Janela | Grau dos polinómios $\{\phi, \beta L, \psi\}$ | SSE (Treino) | SSE (Teste) | MRE % (Teste) | MSE |
|--------|---|--------------|-------------|---------------|-----------------------|
| 80 | $\{5, [4\ 2\ 2\ 5], 1\}$ | 5,97 | 5,98 | 1,96 | $1,23 \times 10^{-4}$ |
| 160 | $\{8 [2\ 9\ 10\ 6], 9\}$ | 0,06 | 1,46 | 0,91 | $9,39 \times 10^{-5}$ |
| 320 | $\{5, [2\ 4\ 10\ 6], 10\}$ | 0,28 | 1,03 | 0,79 | $2,38 \times 10^{-4}$ |
| 1920 | $\{6, [5\ 9\ 3\ 5], 10\}$ | 0,17 | 0,09 | 0,24 | $1,18 \times 10^{-4}$ |

4.1.4. CONTROLO MPC GENERALIZADO UTILIZANDO ALGORITMO GENÉTICO (CASO 1)

Ao obter o modelo do sistema, encontrado através da metodologia ARMAX, o próximo passo foi o de entregar a resposta desse modelo 3 horas à frente (a previsão a um horizonte de predição de 30 amostras) ao otimizador (algoritmo genético) com a finalidade de devolver a sequência futura ideal e necessária para a temperatura da água introduzida no tanque da piscina.

Neste caso foram usadas as bibliotecas *built-in* do *MATLAB* para execução do algoritmo genético (*Matlab GA*). Assim, recorrendo à função *GA* e implementada a função *fitness*, as soluções propostas pelo *GA* tinham como principal critério a qualidade da previsão do valor da temperatura desejada, ou *Tsetpoint*.

O cenário considerado, assemelha-se à utilização de uma fonte de calor com capacidade calorífica instantânea, por exemplo uma caldeira (tal como representado na figura 4.8).

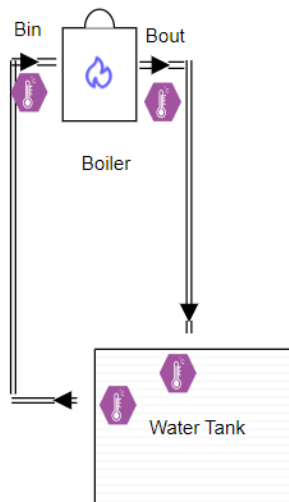


Figura 4.8. Utilização de Caldeira para aquecimento da água do tanque da piscina

Nos resultados seguintes o intervalo de valores considerado para a temperatura da água (*Bout*) proveniente da caldeira foi $[1, 50]$ °C. Em todos os casos, os modelos ARMAX obtidos são estimados usando uma janela temporal com 1920 amostras (de instantes de tempo anteriores), os critérios de avaliação dos modelos usados foram R^2 e valor de percentagem de MRE.

No primeiro cenário a temperatura de *setpoint* desejada foi definida como 22 graus, para uma temperatura inicial de 18 °C. Para este efeito, executa-se o algoritmo genético e, juntamente com as previsões do modelo e os dados meteorológicos, estima-se a temperatura

da água necessária à saída da caldeira (*Bout*). A hora inicial para o início da aplicação do controlo MPC é 05h00 do dia 13 de abril de 2022.

A tabela 4.8 mostra os resultados obtidos para um conjunto de polinomiais. Assinalado a negrito está o melhor modelo selecionado entre os vários testados. Este foi considerado o melhor porque, segundo uma das métricas definidas, o modelo que apresentar o maior R^2 é considerado o melhor

Tabela 4.8. Graus das polinomiais e respetivo valor de R^2

| Graus polinomiais $\{\phi, \beta t, \psi\}$ | R^2 treino | R^2 teste |
|---|-------------------|-------------|
| {9, [4 8 6 2],2} | 0,99999137 | 0,99999905 |
| {10, [7 8 5 10],7} | 0,99999147 | 0,99999885 |
| {5, [1 7 3 9],10} | 0,99999145 | 0,99999932 |
| {1, [9 6 2 2],7} | 0,99998218 | 0,99994849 |

Utilizando o melhor modelo, foi realizada uma simulação de modo a prever a temperatura da piscina para um momento futuro utilizando o algoritmo genético, conforme ilustrado na figura 4.9, considerando $T_{setpoint}$ igual a 22°C.

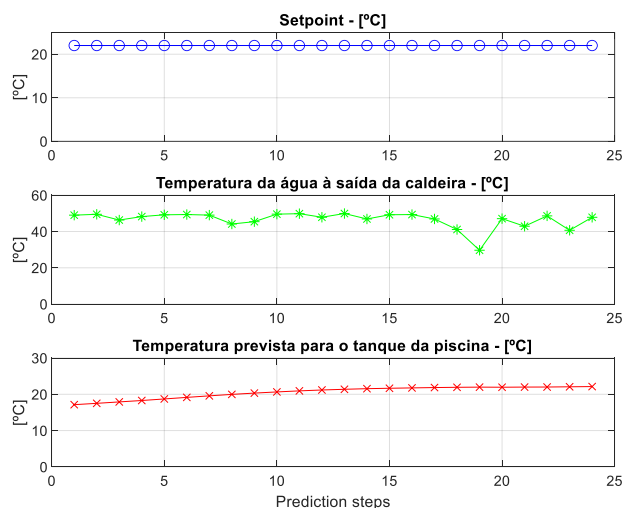


Figura 4.9. Utilização do algoritmo genético: topo: $T_{setpoint}$ de 22 °C; meio: temperatura da água à saída da caldeira; abaixo: Temperatura prevista para o tanque da piscina

Analisando a figura 4.9, observa-se que o otimizador consegue regular a temperatura da água à saída da caldeira, de forma a tender para a temperatura desejada de 22 °C, fazendo-o a partir do tempo futuro correspondente ao passo 15.

Numa segunda simulação foi especificado que o valor para $T_{setpoint}$ fosse 25° C de modo a realizar a comparação com outras estratégias de controlo (figura 4.10).

A figura 4.10 ilustra uma comparação entre diferentes estratégias de controlo de temperatura da piscina ao longo de um dia. São analisados três cenários distintos: o controlo baseado no esforço garantido (Caldeira + Temperatura da piscina), o melhor esforço (Temperatura da água do coletor) e a ausência de controlo. A análise desses cenários permite entender como cada estratégia impacta na manutenção da temperatura desejada da piscina e na eficiência energética do sistema.

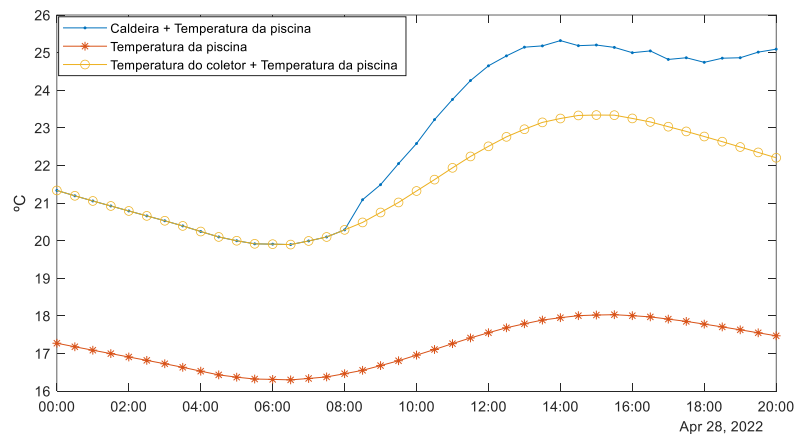


Figura 4.10. Comparação entre o controlo baseado no esforço assegurado (Caldeira + Temperatura da piscina) com um valor de MRE% de 6,15%, o melhor esforço (Temperatura da água do coletor) com um valor de MRE% de 14,10% e nenhum controlo

No primeiro cenário na figura 4.10 (curva contínua), o controlo baseado no esforço assegurado (Caldeira + Temperatura da piscina) apresenta um valor de MRE de 6,15%. Este método utiliza a caldeira para auxiliar na manutenção da temperatura da piscina, resultando numa curva de temperatura mais estável e próxima do *setpoint* desejado ao longo do tempo. No segundo cenário, a estratégia de melhor esforço (utilizando coletor solar para aquecer a piscina) apresenta um valor de MRE% de 14,10%. Este método, que depende exclusivamente da temperatura captada pelos coletores solares, mostra uma maior variação na temperatura da piscina, especialmente em períodos de menor incidência solar. O terceiro cenário, sem qualquer controlo nem sistema de aquecimento (indicada por Temperatura da piscina no

gráfico), e como é evidente, resultou em maiores flutuações de temperatura não permitindo atingir a temperatura desejada estabelecida.

Esses resultados destacam a importância de utilizar uma estratégia de controlo adequada para manter a temperatura da piscina dentro de uma faixa desejada, minimizando o erro em relação ao valor de $T_{setpoint}$. O uso da caldeira como suporte ao sistema solar permite uma manutenção mais eficiente e precisa da temperatura, resultando num menor MRE% comparado ao uso exclusivo dos coletores solares. Este equilíbrio entre precisão e eficiência energética é crucial para a otimização dos sistemas de aquecimento de piscinas.

Através da Figura 4.11 é possível observar os valores para a temperatura da água à saída da caldeira (com variações entre 20 °C e 50 °C) estimados pelo otimizador para o horizonte de previsão das 12 horas, iniciando às 8h00 e terminado pelas 20h00.

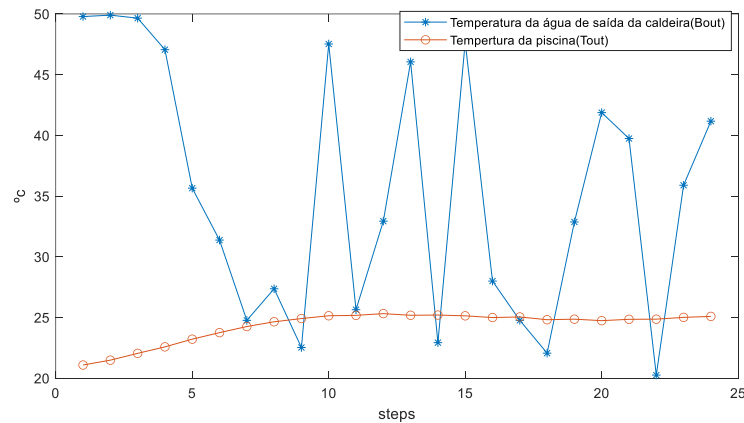


Figura 4.11. Comparação entre: (*) os valores estimados da temperatura da água de saída da caldeira como resultado da aplicação do controlo baseado no esforço garantido e (+) a temperatura da água de saída dos coletores solares como resultado do controlo baseado no melhor esforço

4.1.5. PROPOSTA DE CONTROLO UTILIZANDO ALGORITMO GENÉTICO (CASO 2)

Conforme a secção 4.1.3 para fazer a previsão futura da temperatura da água, foi necessário encontrar em primeiro lugar a dinâmica que representa cada sistema e essa dinâmica representa uma sequência com numeração atribuída na tabela 4.6 (por exemplo a piscina isolada é representada pelo número 1, piscina com efeito do coletor solar pelo número 2, etc).

Determinados os modelos para os diversos sistemas, é agora possível adotar a estratégia indicada no capítulo 3, seção 3.4, neste sentido, o algoritmo genético neste caso funciona como um escalonador porque devolve a melhor sequência, representando a melhor combinação

futura entre os diferentes cenários ou *setups*, para atingir o funcionamento pretendido do sistema.

Concretamente, para o algoritmo genético, os seguintes hiper-parâmetros foram estabelecidos: a) número de indivíduos = 40, b) número de gerações = 15, c) número de modelos = 4, d) percentagem de cruzamento = 60 %, e) horizonte de previsão = 24 (para resultar num horizonte temporal de 12 horas).

A codificação do cromossoma usada implementada no algoritmo genético consiste numa lista de números inteiros, porque o *i-ésimo* gene de posição codifica o número do cenário (um entre os da lista de cenários) no *i-ésimo* passo de tempo futuro. Por exemplo, a partir do instante de tempo $t=0$, para um horizonte de previsão = 24 passos e uma lista de 4 cenários possíveis teríamos:

| | | | | | | | | | | | | | | | | | | | | | | | |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|----------|
| 4 | 3 | 4 | 1 | 2 | 1 | 1 | 1 | 2 | 4 | 3 | 2 | 2 | 4 | 1 | 4 | 1 | 4 | 4 | 4 | 1 | 3 | 1 | 2 |
| $t=0+0$ | 30 | 60 | | | | | | | | | ...330... | | | | | | | | | | | 690 (s) | |
| 08:00 | 08:30 | 09:30 | 10:00 | 10:30 | 11:00 | 11:30 | 12:00 | 12:30 | 13:00 | 13:30 | 14:00 | 14:30 | 15:00 | 15:30 | 16:00 | 16:30 | 17:00 | 17:30 | 18:00 | 18:30 | 19:00 | 19:30 | 20:00(h) |

Se cada passo de tempo corresponder a 30 minutos, o indivíduo anterior diria ao sistema automatizado para começar com o cenário 4, depois mudar para o cenário 3 no instante de tempo $t = 30$ minutos, voltar ao cenário 4 no instante de tempo $t = 60$ minutos, mudar para o cenário 1 em $t = 90$ minutos e, finalmente, escolher os cenários possíveis até ao final dos 24 passos.

| | | | | | | | | | | | | | | | | | | | | | | | |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|----------|
| 4 | 1 | 4 | 1 | 2 | 1 | 1 | 1 | 2 | 4 | 3 | 2 | 2 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 4 | 4 |
| $t=0+0$ | 30 | 60 | | | | | | | | | ...330... | | | | | | | | | | | 690 (s) | |
| 08:00 | 08:30 | 09:30 | 10:00 | 10:30 | 11:00 | 11:30 | 12:00 | 12:30 | 13:00 | 13:30 | 14:00 | 14:30 | 15:00 | 15:30 | 16:00 | 16:30 | 17:00 | 17:30 | 18:00 | 18:30 | 19:00 | 19:30 | 20:00(h) |

O gráfico da figura 4.12 mostra a previsão de temperatura correspondente aos diferentes cenários devolvidos pelo algoritmo genético para a temperatura ao longo do intervalo de 24 passos. O intervalo de 0 para a frente corresponde à previsão ao longo de 24 passos.

O tempo anterior a 0 corresponde aos dados passados utilizados para fazer a previsão futura

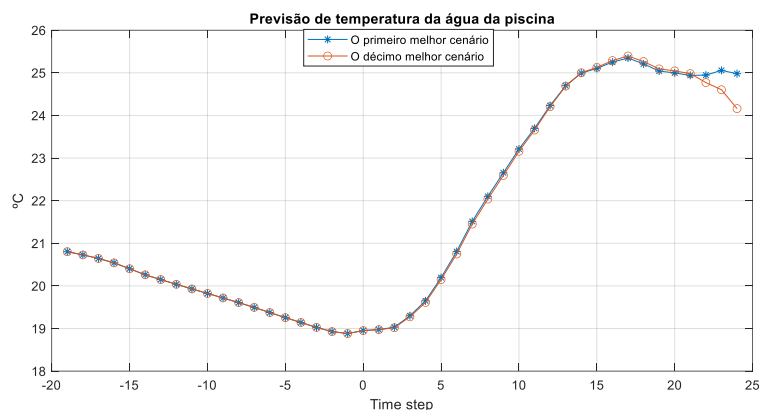


Figura 4.12. Primeira e décima melhores configurações obtidas pelo algoritmo genético para o funcionamento do sistema durante 12 horas

4.2. PREPARAÇÃO E DIVISÃO DE DADOS PARA TREINO E TESTE UTILIZANDO O SARIMAX

Todos os resultados desta seção foram produzidos através do *software TRNSYS*. Em concreto, os dados foram capturados a cada 30 minutos, ou seja, todos os instantes de tempo considerando correspondem a períodos de meia hora. Reportando aos *packages* disponíveis em python, recorreu-se a modelos SARIMAX para encontrar os modelos dos diversos subsistemas, e posteriormente combina-se o otimizador com os modelos de modo a implementar a estratégia de controlo proposta, embora em python.

Para a modelação dos subsistemas foi utilizado o SARIMAX para encontrar o modelo com a melhor dinâmica de cada cenário. O conjunto de cenários considerado é composto por 4 subsistemas: (1) piscina isolada, (2) piscina com efeito do coletor solar térmico, (3) piscina com efeito do pavimento e por fim (4) piscina com efeito da geotermia.

Durante a simulação para encontrar a melhor dinâmica para cada um dos subsistemas foram utilizados como entradas exógenas as seguintes variáveis: Temperatura exterior, velocidade do vento e, humidade relativa. A sequência temporal destas variáveis está apresentada na figura 4.13.

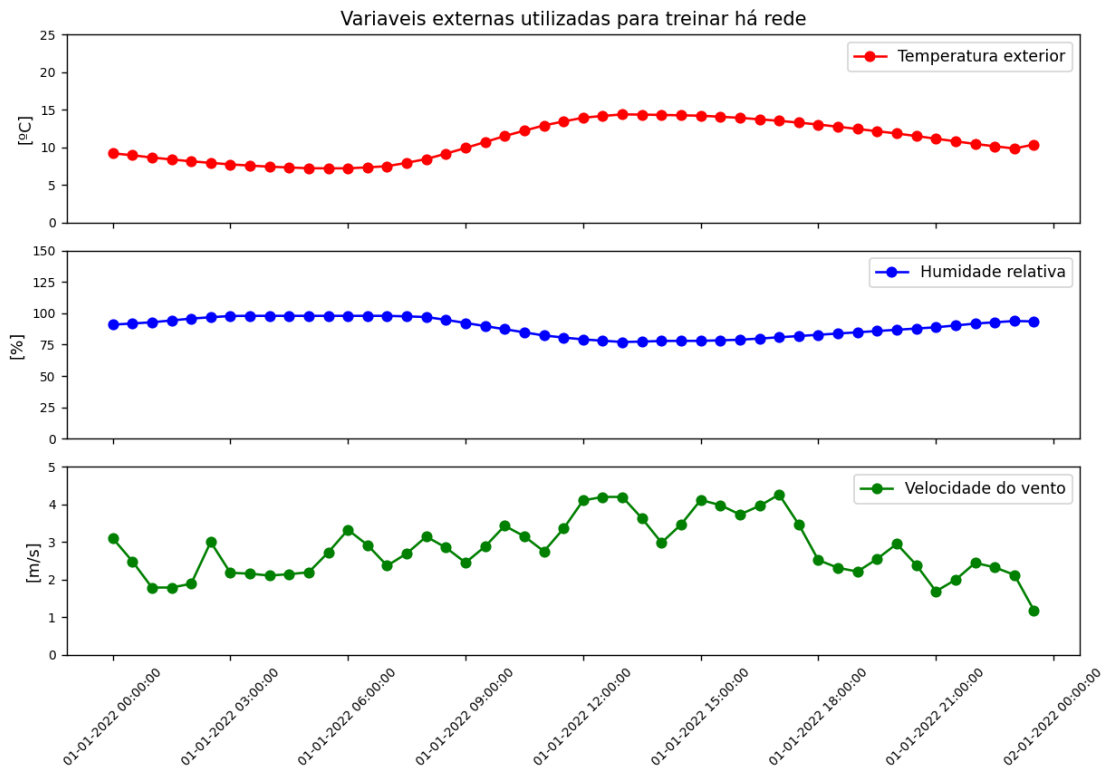


Figura 4.13. Janelas de dados correspondente a dois dias (48 dados de 30 em 30 minutos), utilizadas para treinar a rede

A Tabela 4.9 destaca as melhores ordens para os modelos SARIMAX dos quatro subsistemas analisados. A análise mostra que cada subsistema requer um grau da polinomial SARIMAX específica. A seleção desses graus foi baseada na análise dos critérios de informação disponíveis, como AIC, BIC e HQIC (ver capítulo 2, seção 2.8). Conforme indicado na tabela, a ordem (2,2,3) foi considerada a melhor, pois apresentou os menores valores de AIC, BIC e HQIC, sendo assim a mais adequada para o modelo

Tabela 4.9. Resumo das ilustrações de alguma ordem encontrada para os quatros subsistemas

| Ordem (p, d, q) | AIC | BIC | HQIC | Subsistema |
|----------------------------|-------------|-------------|-------------|-------------------|
| (1,1,1) | -6,83 | 4,27 | -2,65 | 1 |
| | - 188,29 | - 177,19 | - 184,12 | 2 |
| | - 273,88 | - 262,77 | - 269,69 | 3 |
| | 233,64 | 244,74 | 237,82 | 4 |
| (1,1,2) | -12,49 | 0,46 | -7,62 | 1 |
| | - 218,42 | - 205,47 | - 213,55 | 2 |
| | - 292,63 | - 279,69 | - 287,76 | 3 |
| | 235,19 | 248,15 | 240,07 | 4 |
| (2,1,2) | -41,56 | -26,76 | -35,98 | 1 |
| | - 215,69 | - 200,89 | - 210,12 | 2 |
| | - 288,06 | - 273,26 | - 282,49 | 3 |
| | 232,22 | 247,03 | 237,79 | 4 |
| (2,2,3) | -31,84 | -15,38 | -25,67 | 1 |
| | - 215,73 | - 199,27 | - 209,57 | 2 |
| | - 295,97 | - 279,51 | - 289,81 | 3 |
| | 233,83 | 250,28 | 239,99 | 4 |

As Figuras 4.14, 4.15, 4.16 e 4.17, na parte superior ilustram a série temporal dos dados utilizados para o treino do modelo, para os diferentes subsistemas, respetivamente. Por outro lado, as Figuras 4.14, 4.15, 4.16 e 4.17, na parte inferior tanto a série temporal dos instantes futuros correspondente a 12 horas, como a previsão do modelo de cada subsistema, respetivamente. Adicionalmente, é apresentado ainda o valor do erro médio relativo em percentagem.

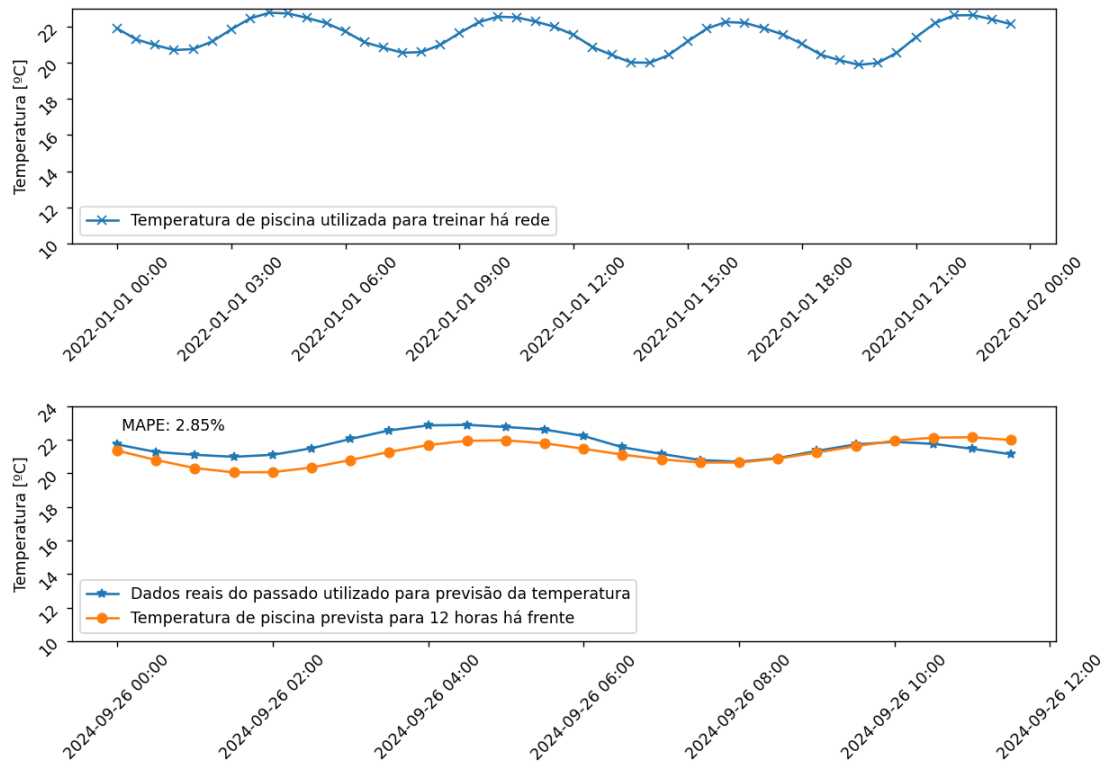


Figura 4.14. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado

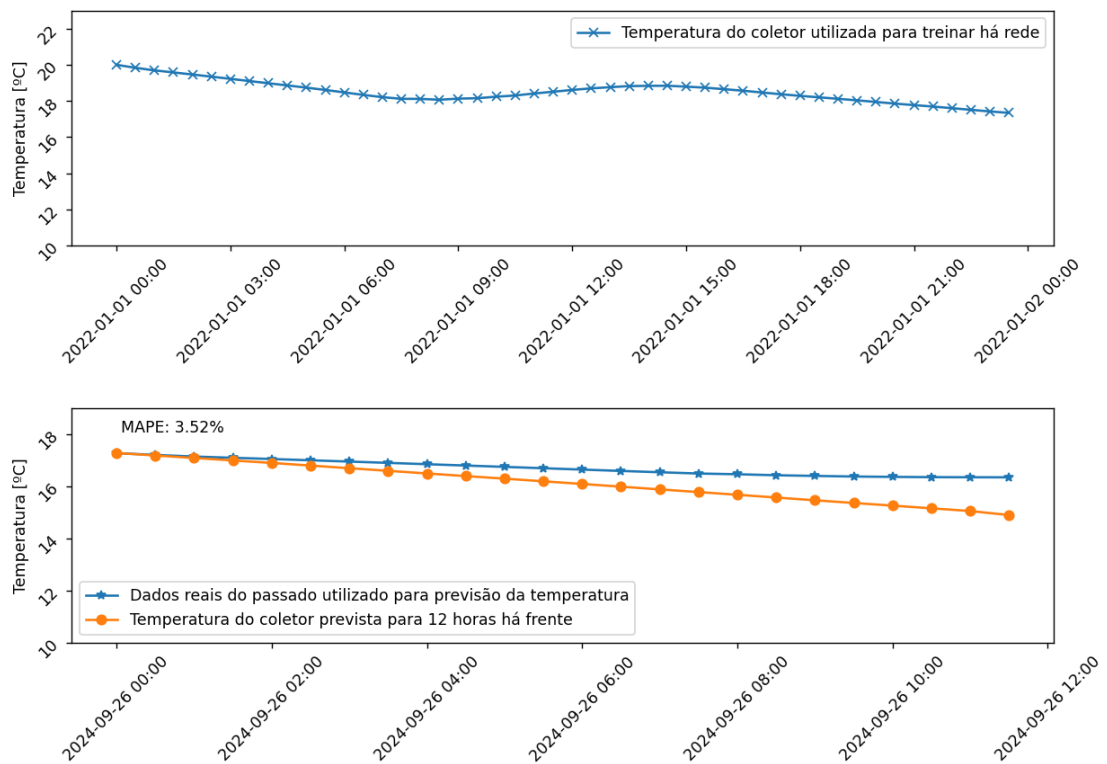


Figura 4.15. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado

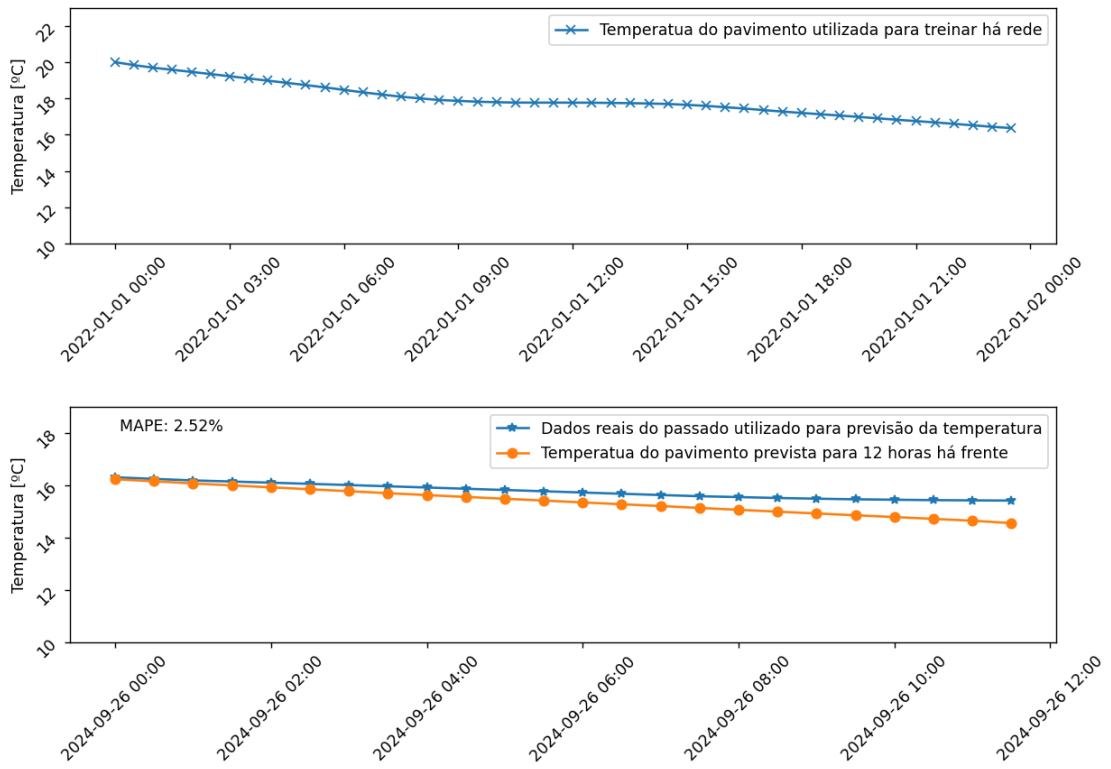


Figura 4.16. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado

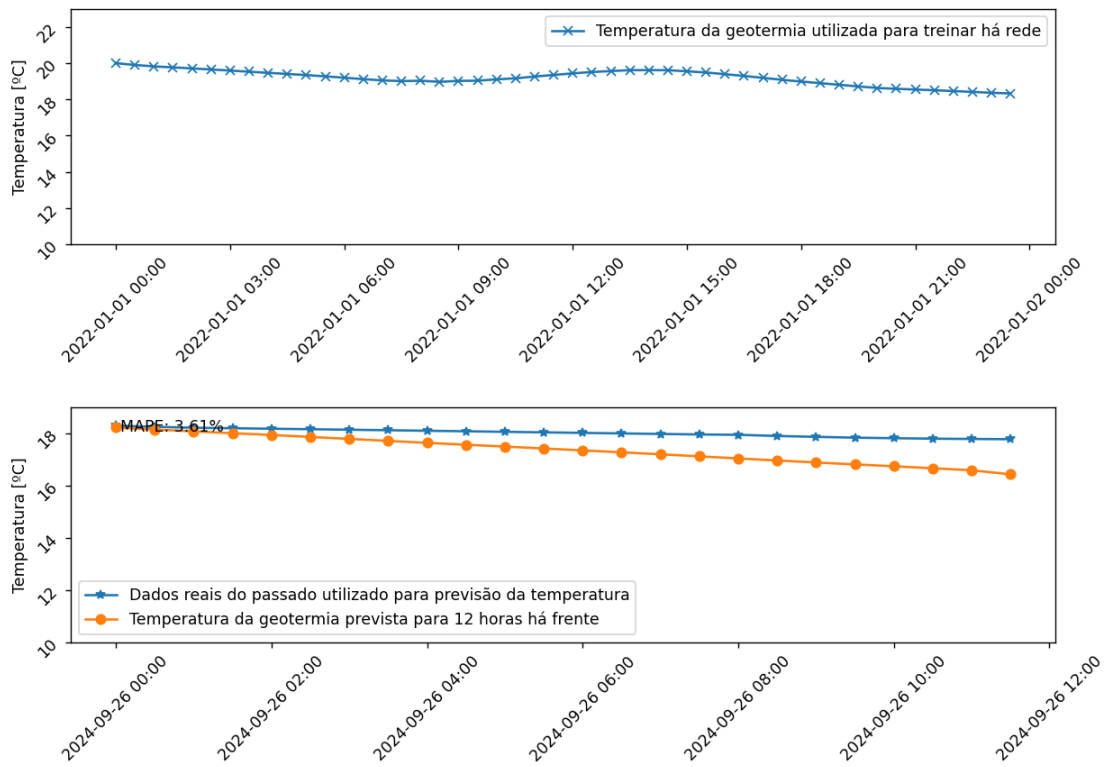


Figura 4.17. Gráfico superior: dados utilizados para treinar a rede; gráfico inferior: comparação entre a previsão da temperatura a 12 horas e valor esperado

Ao analisar as figuras 4.14, 4.15, 4.16 e 4.17 verifica-se que para o primeiro subsistema se obteve um valor de MAPE de 2,85% que representa a piscina isolada, valor de MAPE de 3.52% para o subsistema da piscina com efeito do coletor solar, valor de MAPE de 2,52% para o subsistema da piscina com efeito do pavimento e valor de MAPE 3.16% para o subsistema da piscina com efeito da geotermia.

Com base na solução realizada (sequência entre sistemas) e considerado os quatro modelos que representam os subsistemas, foi então utilizado o otimizador para coordenar o funcionamento do sistema como um todo, a fim de manter a temperatura da água da piscina conforme o desejado durante um determinado período.

Para a simulação foi considerado uma população de 200 indivíduos, codificação do cromossoma com valores inteiros no intervalo [13], e 50 gerações. Foi definida como temperatura alvo, 24 °C ao longo de todo o horizonte de previsão.

A figura 4.18 ilustra as variações do fitness de acordo com o tamanho da população ao longo das gerações. Os valores do fitness apresentados nesta figura foram obtidos a partir da função fitness descrita no Apêndice B.1, criada para avaliar cada código genético dos indivíduos, para uma população gerada conforme o conjunto de indivíduos mencionados acima.

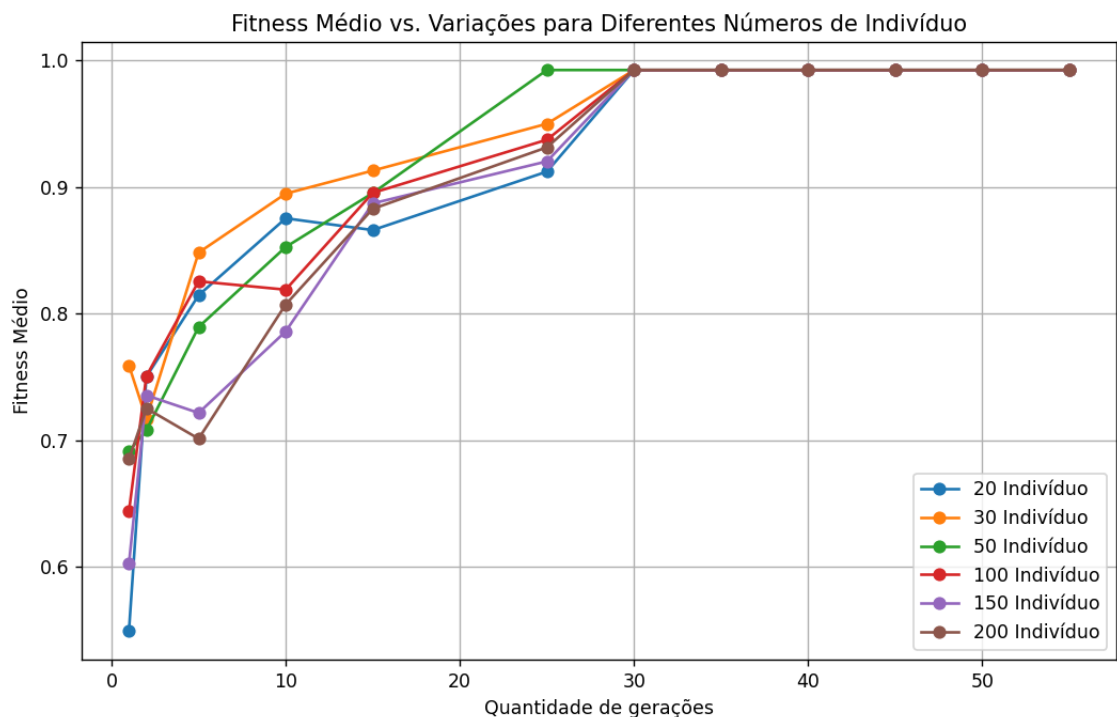


Figura 4.18. Variações do fitness de acordo com o tamanho da população ao longo das gerações

Ao analisar a Figura 4.18, observamos a evolução do fitness médio ao longo das gerações, que ilustra claramente a dinâmica do algoritmo evolutivo em questão. No início do processo, o aumento do fitness médio indica que o algoritmo está explorando o espaço de soluções de forma eficaz. No entanto, a partir de aproximadamente 25 gerações, a estabilização do fitness sugere uma convergência para um ótimo local, levantando a necessidade de estratégias adicionais, como reinicialização ou mutação, para evitar a estagnação.

Além disso, a análise do desempenho em relação ao tamanho da população, especialmente a eficácia de uma população de 50 indivíduos, destaca a importância deste parâmetro na otimização do algoritmo. A Figura 4.18 serve como uma representação visual das tendências identificadas, oferecendo um contexto prático para a compreensão da eficiência do algoritmo evolutivo ao longo das gerações.

O algoritmo decidiu que para alcançar a temperatura desejada pelo utilizador, a melhor coordenação para o funcionamento dos subsistemas será:



Onde:

- 1: representa a temperatura da piscina quando opera sem suporte adicional, apenas com seus próprios sistemas internos.
- 2: representa a situação em que a piscina funciona com um coletor solar. Esse subsistema ajuda a aumentar a temperatura da água através da energia solar, mostrando um aumento significativo na temperatura.
- 3: indica a operação da piscina com o suporte do PCM, que ajuda a estabilizar a temperatura da água através do armazenamento e liberação de energia térmica.

A figura 4.19 representa a melhor coordenação para o funcionamento dos subsistemas térmicos, conforme determinado pelo algoritmo genético. A coordenação mostrada na figura 4.18 é a melhor estratégia desenvolvida pelo algoritmo genético para manter a temperatura da água durante um período de 12 horas, conforme a necessidade do utilizador. O algoritmo avalia diversas combinações para manter a temperatura desejada.

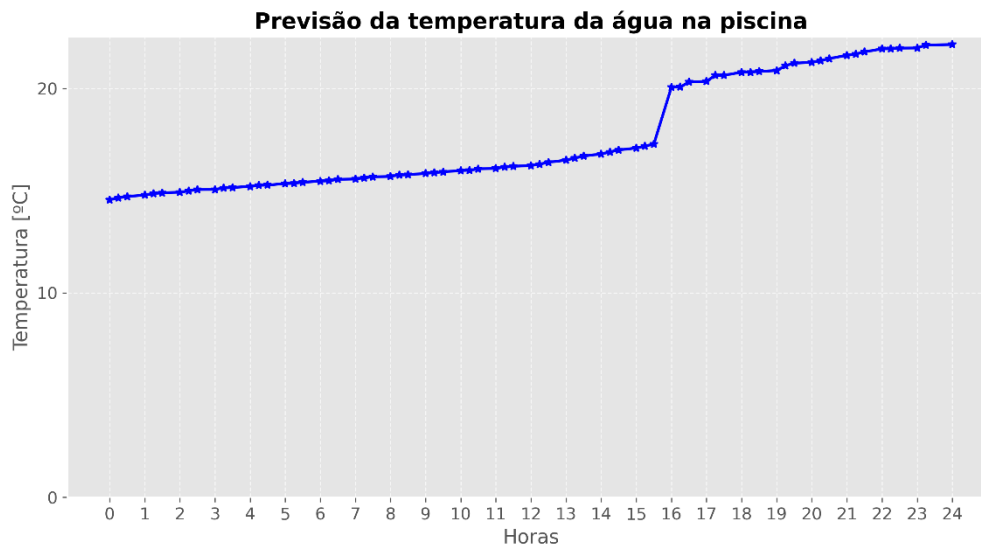


Figura 4.19. Previsão da temperatura da piscina baseada na melhor coordenação dos subsistemas térmicos, para um período de 8 horas de previsão (das 16h até às 24h)

A curva apresentada na figura 4.19 reflete a eficácia dos subsistemas em subir a temperatura da piscina, utilizando a melhor coordenação possível conforme determinado pelo algoritmo genético. Esta abordagem otimizada garante que a temperatura da água seja mantida de acordo com a necessidade do utilizador, promovendo conforto e eficiência energética.

4.3. PLATAFORMA WEB

A plataforma web foi desenvolvida utilizando o *flask*, que por sua vez utiliza a linguagem de programação *PYTHON*, HTML e CSS como *front-end*.

O utilizador para ter acesso à plataforma *Web* em primeiro lugar tem de efetuar o registo (apêndice B.5 ilustra o código feito em *PYTHON* para permitir o registo do utilizador e através do formulário feito em HTML conforme apêndice C.1) na plataforma conforme mostra a figura 4.20. Só depois pode ter acesso às funcionalidades contidas nela.

Ao selecionar o botão registar da figura 4.20, aparece uma tela onde o utilizador pode introduzir os seus dados de registo na plataforma, que são guardados ao clicar no botão registar.



Figura 4.20. Aspeto da plataforma Web. Ilustração das telas mostrando os passos para o registo e login na plataforma

4.3.1. MONITORIZAÇÃO EM TEMPO REAL

Para efeitos de monitorização das informações na plataforma, foi desenvolvido um código em *PYTHON* (ver apêndice B.7) e *HTML* (ver apêndice C.3 que permite fazer *requests/solicitações* diretamente ao autómato utilizando o protocolo *Modbus* no caso de querer saber as informações ou parâmetros dos subsistemas em tempo real.

Quando se pretende ver o histórico de informações acumulado durante algum tempo (dia, mês, ano, etc.), são feitas as solicitações diretamente à plataforma *IoT*, porque a mesma contém uma base de dados integrada. A figura 4.21 ilustra o caso da monitorização de informação em tempo real na plataforma *IoT*, com histórico de informações durante alguns meses e o histórico a nível dos alarmes sobre algumas ocorrências envolvendo os subsistemas.



Figura 4.21. Aspeto da plataforma Web. Exemplo de telas com a monitorização através da plataforma

4.3.2. ALTERAÇÃO DOS PARÂMETROS DOS SUBSISTEMAS

Para alteração dos parâmetros dos subsistemas foi desenvolvido uma função em *PYTHON* (ver apêndice B.8). Ao serem introduzidos dados através de um formulário HTML (ver apêndice C.4) o utilizador terá o botão alterar cenário, onde vai clicar para ativar os parâmetros desejado.

Antes do utilizador introduzir os parâmetros desejados, tem como opção a seleção do cenário de funcionamento que deseja, podendo optar por ser manual e ou automático, conforme ilustrado na figura 4.22.

O funcionamento automático do sistema será realizado de acordo com o *script* programado no PLC, enquanto o funcionamento manual apenas será acionado quando o utilizador desejar alterar a temperatura da água. Nesse caso, o utilizador introduzirá os parâmetros desejados (mais concretamente, a temperatura pretendida) e, assim que o sistema receber o *input*, deixará de funcionar no modo automático, passando a operar no modo manual.



Figura 4.22. Aspeto da plataforma Web. Exemplo de telas com o modo de funcionamento e formulário para alterar os parâmetros desejados

5. CONCLUSÕES E TRABALHOS FUTUROS

5.1. ANÁLISE DO TRABALHO REALIZADO

Este trabalho demonstrou a viabilidade de aplicar tecnologias avançadas no controle inteligente da temperatura de piscinas, utilizando fontes de energia renováveis e algoritmos de otimização, como o genético. A revisão bibliográfica destacou as limitações dos sistemas tradicionais de aquecimento em termos de consumo energético e mostrou que a integração de fontes renováveis, como energia solar e geotérmica, associada ao controle preditivo por modelo (MPC), oferece uma solução mais sustentável. A simulação dos diferentes cenários, combinando subsistemas como coletores solares, pavimentos aquecidos e geotermia, proporcionou uma visão abrangente das possíveis soluções para manter a temperatura da água dentro das preferências do utilizador, garantindo eficiência energética e sustentabilidade.

Os resultados indicaram que o uso de fontes renováveis é eficaz, sobretudo em condições ambientais favoráveis, enquanto o uso de caldeiras pode ser necessário em condições menos ideais para garantir o conforto térmico desejado. O algoritmo genético mostrou-se uma ferramenta eficiente na coordenação dos subsistemas, equilibrando conforto e consumo energético. A plataforma inteligente desenvolvida permite o monitoramento e ajuste em tempo real, otimizando o uso dos recursos.

A aplicação prática destas soluções poderá contribuir significativamente para a redução da dependência de fontes de energia não renováveis e promover um uso mais sustentável dos recursos, garantindo economia e eficiência energética. Em resumo, o sistema proposto oferece uma abordagem inovadora para o controle de piscinas exteriores, alinhando as necessidades dos utilizadores com a sustentabilidade ambiental.

O algoritmo desenvolvido foi capaz de utilizar como entradas conjuntas de dados simulados pelo TRNSYS e, com base neles, controlar de forma otimizada os diferentes subsistemas da piscina analisada, calculando os erros percentuais associados a cada subsistema. O algoritmo demonstrou um bom desempenho ao prever o comportamento dos subsistemas, resultando em valores de MAPE de 2,85% para o sistema de piscina isolada, 3,52% para o subsistema da piscina com o efeito do coletor solar, 2,52% para o subsistema com o efeito do pavimento e 3,16% para o subsistema com o efeito da geotermia. Esses valores demonstram a eficiência do algoritmo em estimar com precisão as variações entre os dados simulados e os resultados previstos para cada subsistema.

Apesar dos resultados promissores, há margem para melhorias, principalmente na integração de técnicas que assegurem maior precisão em diferentes condições meteorológicas ao longo do ano.

5.2. TRABALHOS FUTUROS

Propõe-se adaptar o trabalho desenvolvido a uma nova dinâmica de funcionamento por forma que a técnica a ser desenvolvida futuramente seja aplicado a toda estação do ano melhorando o desempenho da mesma incluindo o inverno com o foco em:

➤ **Otimização do sistema de controlo:**

Implementação de estratégias de controlo preditivo para antecipar as necessidades de aquecimento e ao mesmo tempo minimizar o consumo de energia.

➤ **Utilização de cobertura térmica:**

A cobertura da piscina durante a noite ou em dias com pouca radiação solar pode ajudar a reter o calor da água e diminuir significativamente as perdas térmicas. A escolha do tipo de cobertura ideal dependerá do tamanho da piscina, do clima local e das preferências estéticas dos utilizadores.

➤ **Melhoria do isolamento térmico:**

A redução das perdas de calor da piscina é crucial para a eficiência do sistema, especialmente em climas mais frios. O investimento em isolamento térmico de alta qualidade para as paredes e o fundo da piscina pode minimizar a dissipação de calor e reduzir a demanda por aquecimento.

5.3. PUBLICAÇÕES E CONTRIBUIÇÕES RELACIONADAS COM O TRABALHO

REALIZADO

A presente dissertação contribuiu significativamente para a elaboração de dois destes artigos, os quais se encontram listados a seguir:

1. Cabrita, C., Carvalho, J., Inverno, A., Monteiro, J., Oliveira, M.J. (2023). Smart Control System for Sustainable Swimming Pools. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloat, P.M. (eds) Computational Science – ICCS 2023. ICCS 2023. Lecture Notes in Computer Science, vol 10477. Springer. DOI: https://doi.org/10.1007/978-3-031-36030-5_44.

-
2. Cristiano Cabrita, Jailson Carvalho, Armando Inverno, Jânio Monteiro, Miguel J. Oliveira, “Model Based Control System for Outdoor Swimming Pools”, International Conference on Human-Computer Interaction, HCII 2024). Lecture Notes in Computer Science, vol 14696. Springer. DOI: https://doi.org/10.1007/978-3-031-60875-9_16.

REFERÊNCIAS

- [1] EU, “Official Site of the European Union (EU) Energy Policies.” Accessed: Jun. 17, 2024. [Online]. Available: https://energy.ec.europa.eu/index_en
- [2] R. N. Índice, “Consumo de energia eléctrica per capita : total e por tipo de consumo.” Accessed: May 17, 2024. [Online]. Available: <https://www.pordata.pt/Portugal/Consumo+de+energia+primária+total+e+por+tipo+d+e+fonte+de+energia-1130>
- [3] “International Energy Agency, Solar Heating and Cooling Programme, SHC Task 29: Solar Crop Drying.” Accessed: Apr. 13, 2024. [Online]. Available: <https://task29.iea-shc.org/>
- [4] M. Abdunnabi, I. H. Tawil, M. Benabeid, M. A. Elhaj, and F. Mohamed, “Design of Solar Powered Space Heating and Domestic Hot Water System for Libyan Common House,” *2021 12th Int. Renew. Energy Congr. IREC 2021*, 2021, doi: 10.1109/IREC52758.2021.9624800.
- [5] “U.S. Census Bureau,” *Choice Rev. Online*, vol. 49, no. 06, pp. 49-3060-49-3060, 2012, doi: 10.5860/choice.49-3060.
- [6] S. Gössling *et al.*, “Tourism and water use: Supply, demand, and security. An international review,” *Tour. Manag.*, vol. 33, no. 1, pp. 1–15, 2012, doi: 10.1016/j.tourman.2011.03.015.
- [7] R. Fernandez, “Final Report Summary - POOLSAFE (A novel swimming pool water treatment for the detection and elimination of excess cyanuric acid.),” *European Commission*. 2016. [Online]. Available: <https://cordis.europa.eu/project/id/604884/reporting>
- [8] T. Gallion, T. Harrison, R. Hulverson, and K. Hristovski, “Estimating Water, Energy, and Carbon Footprints of Residential Swimming Pools,” *Water Reclam. Sustain.*, pp. 343–359, 2014, doi: 10.1016/B978-0-12-411645-0.00014-6.
- [9] J. C. Lam and W. W. Chan, “Life cycle energy cost analysis of heat pump application for hotel swimming pools,” *Energy Convers. Manag.*, vol. 42, no. 11, pp. 1299–1306,

- 2001, doi: 10.1016/S0196-8904(00)00146-1.
- [10] Y. Li, G. Huang, H. Wu, and T. Xu, “Feasibility study of a PCM storage tank integrated heating system for outdoor swimming pools during the winter season,” *Appl. Therm. Eng.*, vol. 134, pp. 490–500, 2018, doi: 10.1016/j.applthermaleng.2018.02.030.
- [11] F. H. Fahmy, A. A. Nafeh, N. M. Ahamed, and H. M. Farghally, “A simulation model for predicting the performance of PV powered space heating system in Egypt,” *ICCCE 2010 - 2010 Int. Conf. Chem. Chem. Eng. Proc.*, pp. 173–177, 2010, doi: 10.1109/ICCCENG.2010.5560389.
- [12] G. Simoes, C. Dionisio, A. Gloria, P. Sebastiao, and N. Souto, “Smart System for Monitoring and Control of Swimming Pools,” *IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc.*, vol. 829–832, pp. 829–832, 2019, doi: 10.1109/WF-IoT.2019.8767240.
- [13] J. M. Marais, D. V. Bhatt, G. P. Hancke, and T. D. Ramotsoela, “A web-based swimming pool information and management system,” *IEEE Int. Conf. Ind. Informatics*, vol. 0, pp. 980–985, 2016, doi: 10.1109/INDIN.2016.7819304.
- [14] C.-W. Hsu, “‘Applied Energy,’ *Appl. Energy*,” vol. 100, no. 205–217, 2012.
- [15] Y. Li and G. Huang, “Temperature control of a PCM integrated open-air swimming pool in cold season: A numerical and experimental study,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 238, no. 1, 2019, doi: 10.1088/1755-1315/238/1/012001.
- [16] C. D. Popa and C. Ungureanu, “Analysis of a Hybrid Water Heating System for a Swimming Pool,” *SIELMEN 2021 - Proc. 11th Int. Conf. Electromechanical Energy Syst.*, pp. 521–526, 2021, doi: 10.1109/SIELMEN53755.2021.9600361.
- [17] J. P. Delgado Marín, F. Vera García, and J. R. García Cascales, “Use of a predictive control to improve the energy efficiency in indoor swimming pools using solar thermal energy,” *Sol. Energy*, vol. 179, pp. 380–390, 2019, doi: 10.1016/j.solener.2019.01.004.
- [18] Y. Dong, H. Yonghong, and G. Xu, “Design of indoor swimming pool water temperature control system based on fuzzy controller and Smith predictor,” *Proc. 2011 Int. Conf. Electron. Mech. Eng. Inf. Technol. EMEIT 2011*, vol. 9, pp. 4678–4681, 2011, doi: 10.1109/EMEIT.2011.6024079.
- [19] G. . Yurievich, A.N.; Gennadievich, “Onopriychuk Denis Alekseevich Solar Batteries—Reality or Fantasy? *Quest. Sci. Educ.*,” p. 38, 2018.
- [20] E. Becquerel, “Mémoire sur les effets électriques produits sous l’influence des rayons solaires,” *Comptes Rendus*, vol. 9. pp. 561–567, 1839.
- [21] R. Dallaev, T. Pisarenko, N. Papež, and V. Holcman, “Overview of the Current State

- of Flexible Solar Panels and Photovoltaic Materials,” *Materials (Basel)*, vol. 16, no. 17, pp. 5–6, 2023, doi: 10.3390/ma16175839.
- [22] A. D. Dhass, D. Patel, and B. Patel, “Estimation of power losses in single-junction gallium-arsenide solar photovoltaic cells,” *Int. J. Thermofluids*, vol. 17, 2023, doi: 10.1016/j.ijft.2023.100303.
- [23] Y. Tian and C. Y. Zhao, “A review of solar collectors and thermal energy storage in solar thermal applications,” *Appl. Energy*, vol. 104, pp. 538–553, 2013, doi: 10.1016/j.apenergy.2012.11.051.
- [24] K. S. Lee J., Wi S., Yun B. Y., Yang S., Park J. H., “Development and Evaluation of Gypsum/Shape-Stabilization PCMs Using Large-Capacity Vacuum Impregnator for Inorganic Salt Hydrate for Thermal Energy Storage,” pp. 278–290, 2019.
- [25] M. A. T. Al-Abidi A. A., Mat S. B., Sopian K., Sulaiman M., “Applications for Latent Heat TES: A Review, Renewable and Sustainable Energy Reviews,” pp. 353–363, 2013.
- [26] F. S. Javadi, H. S. C. Metselaar, and P. Ganesan, “Performance improvement of solar thermal systems integrated with phase change materials (PCM), a review,” *Sol. Energy*, vol. 206, pp. 330–352, 2020, doi: 10.1016/j.solener.2020.05.106.
- [27] G. Dogkas *et al.*, “Investigating the performance of a thermal energy storage unit with paraffin as phase change material, targeting buildings’ cooling needs: an experimental approach,” *Int. J. Thermofluids*, vol. 3–4, pp. 3–4, 2020, doi: 10.1016/j.ijft.2020.100027.
- [28] S. I. N. Rasta I. M., “Study on Thermal Properties of Bio-PCM Candidates in Comparison with Propylene Glycol and Salt Based PCM for sub-Zero Energy Storage Applications,” vol. 494, pp. 12–14, 2018.
- [29] B. D. Sharma A., Tyagi V. V., Chen C. R., “Review on TES with PCMs and Applications,” *Renew. Sustain. Energy Rev.*, vol. C 13(2), pp. 318–345, 2009.
- [30] Z. Ajji and H. Jouhara, “Investigation of the effects of thermal, oxidative and irradiation treatments on the behaviour of poly-ethylene glycol as a phase change material in thermal energy storage systems,” *Energy*, vol. 136, pp. 196–200, 2017, doi: 10.1016/j.energy.2016.01.104.
- [31] Z. Ling, J. Liu, Q. Wang, W. Lin, X. Fang, and Z. Zhang, “Solar Energy Materials and Solar Cells material with improved thermal reliability and enhanced thermal conductivity,” *Sol. Energy Mater. Sol. Cells*, vol. 172, no. 3, pp. 195–201, 2017.
- [32] N. Xie, Z. Huang, Z. Luo, X. Gao, Y. Fang, and Z. Zhang, “Inorganic salt hydrate for

- thermal energy storage,” *Appl. Sci.*, vol. 7, no. 12, pp. 13–17, 2017, doi: 10.3390/app7121317.
- [33] J. Huang, J. Dai, S. Peng, T. Wang, and S. Hong, “Modification on hydrated salt-based phase change composites with carbon fillers for electronic thermal management,” *Int. J. Energy Res.*, vol. 43, no. 8, pp. 3550–3560, 2019, doi: 10.1002/er.4502.
- [34] S. Sari A., Bicer A., Al-Ahmed A., Al-Sulaiman F. A., Zahir M. H., Mohamed S. A., “Capric Acid-Palmitic Acid Composite PCM Doped with CNTs for TES, Solar Energy Materials and Solar Cells,” pp. 353–361, 2018.
- [35] M. E. Darzi, S. I. Golestaneh, M. Kamali, and G. Karimi, “Thermal and electrical performance analysis of co-electrospun-electrosprayed PCM nanofiber composites in the presence of graphene and carbon fiber powder,” *Renew. Energy*, vol. 135, pp. 719–728, 2019, doi: 10.1016/j.renene.2018.12.028.
- [36] A. Karaipekli and A. Sari, “Preparation, thermal properties and thermal reliability of eutectic mixtures of fatty acids/expanded vermiculite as novel form-stable composites for energy storage,” *J. Ind. Eng. Chem.*, vol. 16, no. 5, pp. 767–773, 2010, doi: 10.1016/j.jiec.2010.07.003.
- [37] Y. Li, Z. Ding, and Y. Du, “Techno-economic optimization of open-air swimming pool heating system with PCM storage tank for winter applications,” *Renew. Energy*, vol. 150, pp. 878–890, 2020, doi: 10.1016/j.renene.2020.01.029.
- [38] E. Douvi *et al.*, “Phase change materials in solar domestic hot water systems: A review,” *Int. J. Thermofluids*, vol. 10, p. 100075, 2021, doi: 10.1016/j.ijft.2021.100075.
- [39] S. A. Veerakumar C., “PCM Based Cold TES: Materials, Techniques and Applications – A Review,” vol. 67, pp. 271–289, 2016.
- [40] G. Gholamibozanjani and M. Farid, “Application of an active PCM storage system into a building for heating/cooling load reduction,” *Energy*, vol. 210, p. 118572, 2020, doi: 10.1016/j.energy.2020.118572.
- [41] P. T. Sardari, R. Babaei-Mahani, D. Giddings, S. Yasserli, M. A. Moghimi, and H. Bahai, “Energy recovery from domestic radiators using a compact composite metal Foam/PCM latent heat storage,” *J. Clean. Prod.*, vol. 257, p. 120504, 2020, doi: 10.1016/j.jclepro.2020.120504.
- [42] D. Qin, Z. J. Yu, T. Yang, S. Li, and G. Zhang, “Thermal performance evaluation of a new structure hot water tank integrated with phase change materials,” *Energy Procedia*, vol. 158, pp. 5034–5040, 2019, doi: 10.1016/j.egypro.2019.01.659.
- [43] H. Jouhara, N. Khordehgah, S. Almahmoud, B. Delpech, A. Chauhan, and S. A. Tassou,

- “Waste heat recovery technologies and applications,” *Therm. Sci. Eng. Prog.*, vol. 6, pp. 268–289, 2018, doi: 10.1016/j.tsep.2018.04.017.
- [44] E. Barbier, “Geothermal energy technology and current status: An overview,” *Renew. Sustain. Energy Rev.*, vol. 6, no. 1–2, pp. 3–65, 2002, doi: 10.1016/S1364-0321(02)00002-3.
- [45] S. Watson, “How Stuff Works. com: Energy Production.” Accessed: Jul. 28, 2024. [Online]. Available: <http://wcpssccsi.pbworks.com/w/file/fetch/67747770/geothermal-11162012150546.pdf>.
- [46] M. Ławryńczuk and P. Ocoń, “Model Predictive Control and energy optimisation in residential building with electric underfloor heating system,” *Energy*, vol. 182, pp. 1028–1044, 2019, doi: 10.1016/j.energy.2019.06.062.
- [47] Rockwell Automation, “Micrologix Programmable Controllers.” Accessed: Aug. 04, 2024. [Online]. Available: <https://docs.rs-online.com/8616/0900766b80dfc363.pdf>
- [48] H. M. K. K. M. B. Herath, S. V. A. S. H. Ariyathunge, and H. D. N. S. Priyankara, “Development of a Data Acquisition and Monitoring System Based on MODBUS RTU Communication Protocol,” *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 6, pp. 433–440, 2020, doi: 10.38124/ijisrt20jun479.
- [49] K. Najim and M. M. Saad, “Adaptive control: theory and practical aspects,” *J. Process Control*, vol. 1, no. 2, pp. 84–95, 1991, doi: 10.1016/0959-1524(91)80004-4.
- [50] F. Bukhari, “Adaptive Control of a Production-Inventory Model with Uncertain Deterioration Rate,” *Appl. Math.*, vol. 02, no. 09, pp. 1170–1174, 2011, doi: 10.4236/am.2011.29162.
- [51] G. Shi, C. Dai, M. Ma, D. Li, and Y. Ding, “Process-Model-Free Design of MPC via a Reference-Model-Based PI Controller for the Gas Turbine,” *2023 9th Int. Conf. Electr. Eng. Control Robot. EECR 2023*, pp. 169–177, 2023, doi: 10.1109/EECR56827.2023.10149918.
- [52] J. V Candy, “Model Reference Adaptive Control (MRAC) for Additive Manufacturing Model Reference Adaptive Control (MRAC) for Additive Manufacturing (AM),” 2021.
- [53] A. A. Abdelrauf, M. Abdel-Geliel, and E. Zakzouk, “Adaptive PID controller based on model predictive control,” *2016 Eur. Control Conf. ECC 2016*, pp. 746–751, 2016, doi: 10.1109/ECC.2016.7810378.
- [54] B. P. Campos and M. R. Da Silva, “Demand forecasting in residential distribution feeders in the context of smart grids,” *2016 12th IEEE Int. Conf. Ind. Appl. INDUSCON 2016*, pp. 1–6, 2017, doi: 10.1109/INDUSCON.2016.7874464.

- [55] M. Usman, U. H. Suleiman, M. Usman, and T. Sadiq, "Design of an Optimal Linear Quadratic Regulator (LQR) Controller for the Ball-On-Sphere System," *Int. J. Eng. Manuf.*, vol. 10, no. 3, pp. 56–70, 2020, doi: 10.5815/ijem.2020.03.05.
- [56] I. Awaludin, R. Ibrahim, and K. S. R. Rao, "Conventional ARX and artificial neural networks ARX models for prediction of oil consumption in Malaysia," *2009 IEEE Symp. Ind. Electron. Appl. ISIEA 2009 - Proc.*, vol. 1, pp. 23–28, 2009, doi: 10.1109/ISIEA.2009.5356496.
- [57] A. A. Abdelrauf, W. W. Saad, A. Hebala, and M. Galea, "Model Predictive Control Based PID Controller for PMSM for Propulsion Systems," *2018 IEEE Int. Conf. Electr. Syst. Aircraft, Railw. Sh. Propuls. Road Veh. Int. Transp. Electr. Conf. ESARS-ITEC 2018*, 2018, doi: 10.1109/ESARS-ITEC.2018.8607585.
- [58] K. Goswami, A. Ganguly, and A. Kumar Sil, "Comparing univariate and multivariate methods for short term load forecasting," *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 972–976, 2019, doi: 10.1109/GUCON.2018.8675059.
- [59] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, vol. 236, pp. 1078–1088, 2019, doi: 10.1016/j.apenergy.2018.12.042.
- [60] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, and P. Chatellier, "Modeling and forecasting building energy consumption: A review of data-driven techniques," *Sustain. Cities Soc.*, vol. 48, 2019, doi: 10.1016/j.scs.2019.101533.
- [61] F. Amara, K. Agbossou, Y. Dube, S. Kelouwani, and A. Cardenas, "Estimation of temperature correlation with household electricity demand for forecasting application," *IECON Proc. (Industrial Electron. Conf.)*, pp. 3960–3965, 2016, doi: 10.1109/IECON.2016.7793935.
- [62] "'Arx', MathWorks." Accessed: Jul. 31, 2024. [Online]. Available: <https://www.mathworks.com/help/ident/ref/arx.html>
- [63] "ARIMA." Accessed: Jul. 31, 2024. [Online]. Available: <https://www.statsmodels.org/stable/>
- [64] F. A. Ruslan, K. Haron, A. M. Samad, and R. Adnan, "Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang," *Proc. - 2017 IEEE 13th Int. Colloq. Signal Process. its Appl. CSPA 2017*, pp. 179–184, 2017, doi: 10.1109/CSPA.2017.8064947.
- [65] Z. Hui, L. L. Qin, and W. Gang, "Modeling and simulation of greenhouse temperature hybrid system based on ARMAX model," *Chinese Control Conf. CCC*, pp. 2237–2241,

- 2017, doi: 10.23919/ChiCC.2017.8027690.
- [66] L. C. Jain, M. Seera, C. P. Lim, and P. Balasubramaniam, “A review of online learning in supervised neural networks,” *Neural Comput. Appl.*, vol. 25, no. 3–4, pp. 491–509, 2014, doi: 10.1007/s00521-013-1534-4.
- [67] C. M. Boné R, Assaad M, “Boosting recurrent neural networks for time series prediction. In: Pearson DW, Steele NC, Albrecht RF (eds) Artificial neural nets and genetic algorithms. Springer, Vienna,” 2013.
- [68] A. D. B. Oliveira, S. N. Barros, and R. B. C. Lima, “Controlador NMPC Embarcado para Aeropêndulo Utilizando MLP,” *2021 14th IEEE Int. Conf. Ind. Appl. INDUSCON 2021 - Proc.*, pp. 1469–1475, 2021, doi: 10.1109/INDUSCON51756.2021.9529867.
- [69] J. Hu, X. Wang, Y. Zhang, D. Zhang, M. Zhang, and J. Xue, “Time Series Prediction Method Based on Variant LSTM Recurrent Neural Network,” *Neural Process. Lett.*, vol. 52, no. 2, pp. 1485–1500, 2020, doi: 10.1007/s11063-020-10319-3.
- [70] M. J. Mahmoodabadi, A. Adljooy Safaie, A. Bagheri, and N. Nariman-Zadeh, “A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model,” *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2577–2591, 2013, doi: 10.1016/j.asoc.2012.11.028.
- [71] J. H. . Holland, “Genetic Algorithms understand Genetic Algorithms,” *Sci. Am.*, vol. 267, no. 1, pp. 66–73, 1992.
- [72] S. Mirjalili, S., & Mirjalili, “Genetic algorithm. Evolutionary algorithms and neural networks,” *Theory Appl.*, pp. 43–55, 2019.
- [73] G. C. Pereira, M. M. F. d. Oliveira, and N. F. F. Ebecken, “Genetic Optimization of Artificial Neural Networks to Forecast Virioplankton Abundance from Cytometric Data,” *J. Intell. Learn. Syst. Appl.*, vol. 05, no. 01, pp. 57–66, 2013, doi: 10.4236/jilsa.2013.51007.
- [74] H. de Garis, “Introduction to Evolutionary Computing,” *Evol. Comput.*, vol. 12, no. 2, pp. 269–271, 2004, doi: 10.1162/evco.2004.12.2.269.
- [75] R. Carvalho and K. C. Poldi, “Problema da Mochila.”
- [76] R. Kumar and Jyotishree, “Novel Encoding Scheme in Genetic Algorithms for Better Fitness,” *Int. J. Eng. Adv. Technol.*, vol. 1, no. 6, pp. 214–218, 2012.
- [77] W.-J. Shyr, “Parameters Determination for Optimum Design by Evolutionary Algorithm,” *Converg. Hybrid Inf. Technol.*, 2010, doi: 10.5772/9638.
- [78] J. H. (John H. Holland and J. H., “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence,”

- Univ. Michigan Press*, p. 211, 1992.
- [79] E. O. Alkafaween, “Novel Methods for Enhancing the Performance of Genetic Algorithms,” 2018.
- [80] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, “On stopping criteria for genetic algorithms,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3171, pp. 405–413, 2004, doi: 10.1007/978-3-540-28645-5_41.
- [81] A. K. Seghouane and S. I. Amari, “The AIC criterion and symmetrizing the Kullback-Leibler divergence,” *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 97–106, 2007, doi: 10.1109/TNN.2006.882813.
- [82] G. Schwarz, “Estimating the Dimension of a Model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 2007, doi: 10.1214/aos/1176344136.
- [83] R. L. Eubank and J. D. Hart, “Commonality of cusum, von Neumann and smoothing-based goodness-of-fit tests,” *Biometrika*, vol. 80, no. 1, pp. 89–98, 1993, doi: 10.1093/biomet/80.1.89.
- [84] P. J. Bickel and K. A. Doksum, “Mathematical statistics: Basic ideas and selected topics, volumes I-II package,” *Math. Stat. Basic Ideas Sel. Top. Vol. I-II Packag.*, vol. 1–2, pp. 1–1065, 2015, doi: 10.1201/9781315369266.
- [85] M. Korkmaz, “A study over the general formula of regression sum of squares in multiple linear regression,” *Numer. Methods Partial Differ. Equ.*, vol. 37, no. 1, pp. 406–421, 2021, doi: 10.1002/num.22533.
- [86] “Erro Médio Relativo.” Accessed: Aug. 01, 2024. [Online]. Available: <https://permetrics.readthedocs.io/en/latest/pages/regression/MRE.html>
- [87] J. E. (2022) Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J. and Boylan, “Forecasting theory and practice - International Journal of Forecasting.,” 2022.
- [88] J. Gao, “Office of Productivity, Efficiency and Staffing, US Department of Veterans Affairs,” vol. 5, no. 4, 2024.
- [89] “Statsmodels.tsa.arima_model.ARMA.fit.” Accessed: Aug. 02, 2024. [Online]. Available: https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima_model.ARMA.fit.html#statsmodels.tsa.arima_model.ARMA.fit
- [90] “Open-Source Web-App for Processing.” Accessed: Jul. 01, 2024. [Online]. Available: <https://emoncms.org/>

- [91] “Python.” Accessed: Apr. 01, 2024. [Online]. Available: <https://www.python.org/>
- [92] L. Gupta, “Rest API Tutorial,” RESTfull API. Accessed: Jul. 10, 2024. [Online]. Available: <https://restfulapi.net/>
- [93] A. Golmohammadi, A., Zhang, M., & Arcuri, “Testing RESTful APIs: A Survey. ACM Transactions on Software Engineering and Methodology”, [Online]. Available: <https://doi.org/10.1145/3617175>.

APÊNDICES

- Apêndice A – Matlab
- Apêndice B – Python
- Apêndice C – Html



MATLAB.

A.1 APÊNDICES I

IMPLEMENTAÇÃO DA REDE NEURAL FEEDFORWARD

Este código implementa uma rede neural feedforward para prever a temperatura de uma piscina usando variáveis de entrada contidas em um arquivo Excel

```
% Carregar o ficheiro Excel
[filename, pathname] = uigetfile('data.xlsx');
file = fullfile(pathname, filename);
data = xlsread(file);

% Separar os dados de entrada e saída
% Supondo que as quatro primeiras colunas são as entradas (X) e a última é a
saída (Y)
X = data(:, 1:4); % As quatro entradas
Y = data(:, 5); % A saída (variável alvo para previsão)

% Dividir manualmente os dados em treino (320 primeiras amostras) e teste (120
amostras restantes)
X_train = X(1:320, :);
Y_train = Y(1:320);
X_test = X(321:end, :);
Y_test = Y(321:end);

% Garantir que Y_test tem exatamente 120 valores
Y_test = Y_test(1:120); % Selecionar apenas os primeiros 120 valores reais

% Criar a rede neural feedforward
hiddenLayerSize = 80; % Número de neurónios na camada oculta
net = feedforwardnet(hiddenLayerSize);

% Treinar a rede apenas com os 320 primeiros dados
[net, tr] = train(net, X_train', Y_train');
% Fazer previsões para o conjunto de teste
Y_test_pred = net(X_test'); % Previsões no conjunto de teste

% Garantir que estamos pegando apenas os primeiros 120 valores previstos
```

```

Y_test_pred = Y_test_pred(1:120); % Pegar apenas os primeiros 120 dados
previstos

% Calcular o MRE EM PERCENTAGEM
MRE = mean(abs((Y_test - Y_test_pred) ./ Y_test)) * 100;
MRE = mean(MRE);
% Exibir o valor do MRE no console
fprintf('O Erro Percentual Relativo Médio (MRE) é: %.2f%%\n', MRE);

% Plotar apenas os dados de teste reais vs previsões
figure;

% Plot das previsões no conjunto de teste (preto)
plot(1:120, Y_test_pred, 'k--', 'LineWidth', 1.5);
hold on;

% Plot dos dados reais (azul) - teste
plot(1:120, Y_test, 'b', 'LineWidth', 1.5);

% Plot dos dados MRE EM PERCENTAGEM
%plot(1:120, MRE, 'r', 'LineWidth', 1.5);

% Legendas e etiquetas
legend('Temperatura Prevista', 'Temperatura Real');
xlabel('Amostras (Intervalo de 6 minutos - 12 horas)');
ylabel('Temperatura [°C]');
ylim([0 23]);
grid on;
hold off;

% Exibir o MRE EM PERCENTAGEM no gráfico como texto
text(10, 22, sprintf('MRE: %.2f%%', MRE), 'FontSize', 12, 'Color', 'r');
.

```

A.2 APÊNDICES II

```

IMPLEMENTAÇÃO REDE LONG-SHORT-TERM MEMORY
% Carregar o ficheiro Excel
[filename, pathname] = uigetfile('data.xlsx');
file = fullfile(pathname, filename);
data = xlsread(file);

% Separar os dados de entrada e saída
% Supondo que as quatro primeiras colunas são as entradas (X) e a última é a
saída (Y)
X = data(:, 1:4); % As quatro entradas
Y = data(:, 5); % A saída (variável alvo para previsão)

% Verificar e remover qualquer NaN nas saídas (Y)
nanIndices = isnan(Y);
X(nanIndices, :) = []; % Remover linhas com NaN de X
Y(nanIndices) = []; % Remover os NaN de Y

% Dividir manualmente os dados em treino (320 primeiras amostras) e teste (120
amostras restantes)
X_train = X(1:320, :);
Y_train = Y(1:320);
X_test = X(321:end, :);

```

```

Y_test = Y(321:end);

% Garantir que Y_test tem exatamente 120 valores
Y_test = Y_test(1:120); % Selecionar apenas os primeiros 120 valores reais

% Converter os dados de entrada em cell arrays (sequências de entradas)
X_train_cell = num2cell(X_train', 1); % Cada célula contém [4 x 1]
X_test_cell = num2cell(X_test', 1); % Cada célula contém [4 x 1]

% Manter Y_train como um vetor (saídas contínuas para regressão)
Y_train_vector = Y_train; % Não usar cell arrays para regressão

% Criar a arquitetura da rede LSTM
numHiddenUnits = 200; % Número de unidades na camada LSTM
numLayers = 100; % Número de camadas LSTM

% Definir a rede LSTM
layers = [ ...
    sequenceInputLayer(4) % Camada de entrada com 4 entradas (variáveis)
    lstmLayer(numHiddenUnits, 'OutputMode', 'last') % Camada LSTM com 200
unidades ocultas e previsão da última saída
    fullyConnectedLayer(1) % Camada totalmente conectada com 1 saída
(previsão única)
    regressionLayer]; % Camada de regressão para previsão de valores
contínuos

% Definir as opções de treino
options = trainingOptions('adam', ... % Optimizador Adam
    'MaxEpochs', 100, ... % Número máximo de
épocas
    'MiniBatchSize', 20, ... % Tamanho do mini-batch
    'InitialLearnRate', 0.01, ... % Taxa de aprendizado
inicial
    'GradientThreshold', 1, ...
    'Plots', 'training-progress', ... % Mostrar progresso
durante o treino
    'Verbose', 0);

% Treinar a rede LSTM
net = trainNetwork(X_train_cell, Y_train_vector, layers, options);

% Fazer previsões para o conjunto de teste
Y_test_pred = predict(net, X_test_cell); % Prever os dados de teste

% Garantir que estamos pegando apenas os primeiros 120 valores previstos
Y_test_pred = Y_test_pred(1:120); % Pegar apenas os primeiros 120 dados
previstos

% Calcular o MRE EM PERCENTAGEM
MRE = mean(abs((Y_test - Y_test_pred) ./ Y_test)) * 100;

% Exibir o valor do MRE EM PERCENTAGEM no console
fprintf('O Erro Percentual Relativo Médio (MRE) é: %.2f%%\n', MRE);

% Plotar apenas os dados de teste reais vs previsões
figure;

% Plot das previsões no conjunto de teste (preto)
plot(1:120, Y_test_pred, 'k--', 'LineWidth', 1.5);

```

```

hold on;

% Plot dos dados reais (azul) - teste
plot(1:120, Y_test, 'b', 'LineWidth', 1.5);
hold on;

% Plot dos dados MRE EM PERCENTAGEM
%plot(1:120, MRE, 'r', 'LineWidth', 1.5);

% Legendas e etiquetas
legend('Temperatura Prevista', 'Temperatura Real');
xlabel('Amostras (Intervalo de 6 minutos - 12 horas)');
ylabel('Temperatura [°C]');
ylim([0 23]);
grid on;
hold off;

% Exibir o MRE EM PERCENTAGEM no gráfico como texto
text(10, 22, sprintf('MRE: %.2f%%', MRE), 'FontSize', 12, 'Color', 'r');
.

```

A.3 APÊNDICES III

IMPLEMENTAÇÃO MÉDIA MÓVEL AUTORREGRESSIVA COM ENTRADA EXÓGENA

```

clear all;
clc;
%dataTrain = readmatrix('info_30minutesss.xlsx');
dataTrain = readmatrix('info_30minutes_rad.xlsx');
dataTrain(:,[11,12])= [];
dataCell = cell(1);
dataCell{1} = dataTrain;

entradas = [2,3,4,10];% Entradas do sistema
saidas = [5 6 8 9]; % Índice das saidas para os modelos: Pisc, Pis_Col,
Pisc_Pav e Pis_Col_Cobertura

% inicio = 1505; % dia 1 fevereiro as 08h00 % inicio = 4337; % dia 1 abril as
08h00 % inicio = 10145; % dia 1 agosto as 08h00
% inicio = 6308; % inicio em Maio as 08h00 % inicio = 11633; % inicio em
setembro as 08h00 % inicio = 8657; % inicio em julho as 08h00
inicio = 2849; % inicio em março
Ts = 0.5;cell_count = 1;
%janela = 320;
janela = 2160;
na=[];nb=[];nc=[];nk =[];
modelo_P = []; %modelo piscina
modelo_P_Col = []; %modelo piscina + Coletor
modelo_P_Pav = []; %modelo Piscina + Pavimento
modelo_P_Col_Cob = []; %modelo piscina + Coletor + Cobertura

inicio_Train = inicio-1;
idxTrain = inicio_Train - janela +1:inicio_Train;
idxTest = inicio_Train+1:inicio_Train+24;
train = false
if train == true

```

```

trainData_P{1}=iddata(dataCell{1}(idxTrain+1,saidas(1)),[dataCell{1}(idxTrain,ent
radas)],Ts);

testData_P{1}=iddata(dataCell{1}(idxTest+1,saidas(1)),[dataCell{1}(idxTest,entrad
as)],Ts);
    [R2Test,R2Train,p_cell,modelo_P]=
retornar_modelo(cell_count,trainData_P,testData_P);
    [sse_forecast_p,mre_forecast_p,YForecast_p] =
retorna_especificacoes(trainData_P,testData_P,modelo_P);

trainData_P_Col{1}=iddata(dataCell{1}(idxTrain+1,saidas(2)),[dataCell{1}(idxTrain
,entradas)],Ts);

testData_P_Col{1}=iddata(dataCell{1}(idxTest+1,saidas(2)),[dataCell{1}(idxTest,en
tradas)],Ts);
    [R2Test,R2Train,p_cell,modelo_P_Col]=
retornar_modelo(cell_count,trainData_P_Col,testData_P_Col);
    [sse_forecast_p_col,mre_forecast_p_col,YForecast_p_col] =
retorna_especificacoes(trainData_P_Col,testData_P_Col,modelo_P_Col );

trainData_P_Pav{1}=iddata(dataCell{1}(idxTrain+1,saidas(3)),[dataCell{1}(idxTrain
,entradas)],Ts);

testData_P_Pav{1}=iddata(dataCell{1}(idxTest+1,saidas(3)),[dataCell{1}(idxTest,en
tradas)],Ts);

[R2Test,R2Train,p_cell,modelo_P_Pav]=retornar_modelo(cell_count,trainData_P_Pav,t
estData_P_Pav);
    [sse_forecast_p_pav,mre_forecast_p_pav,YForecast_p_pav] =
retorna_especificacoes(trainData_P_Pav,testData_P_Pav,modelo_P_Pav);

trainData_P_Col_Cob{1}=iddata(dataCell{1}(idxTrain+1,saidas(4)),[dataCell{1}(idxT
rain,entradas)],Ts);

testData_P_Col_Cob{1}=iddata(dataCell{1}(idxTest+1,saidas(4)),[dataCell{1}(idxTes
t,entradas)],Ts);
    [R2Test,R2Train,p_cell,modelo_P_Col_Cob]=
retornar_modelo(cell_count,trainData_P_Col_Cob,testData_P_Col_Cob);
    [sse_forecast_p_col_cob,mre_forecast_p_col_cob,YForecast_p_col_cob] =
retorna_especificacoes(trainData_P_Col_Cob,testData_P_Col_Cob,modelo_P_Col_Cob);
    else
        ph = 1;

        load 08_04_2023_march_rad.mat;
        clear inicio
        %inicio = 1505; % dia 1 fevereiro as 08h00
        janela = 100;
        %inicio = 1490; % dia 1 fevereiro as 00h00 %inicio = 4337; % dia 1 abril
as 08h00
        %inicio = 2849; % inicio em março %inicio = 10145; % dia 1 agosto as
08h00 %inicio = 11633; % inicio em setembro as 08h00
        inicio = 6308; % inicio em Maio as 08h00
        %inicio = 8657; % inicio em julho as 08h00%inicio = 2850;
        Model_list={modelo_P,modelo_P_Col,modelo_P_Pav,modelo_P_Col_Cob};
        system_scheduling = [ones(1,24)*2];

```

```

    %system_scheduling = [ones(1,24)*3]; %system_scheduling = [ones(1,12)*2,
ones(1,12)*1];
    %system_scheduling = [ones(1,12)*2, ones(1,12)*3;%system_scheduling =
[ones(1,12)*2, ones(1,12)*3, ones(1,24)*3];
    %system_scheduling = [ones(1,12)*2, ones(1,12)*3, ones(1,24)*1,
ones(1,20)*2];
    ref = ones(length(system_scheduling),1)*25;
    saida_passado = 6;
    n_times = length(system_scheduling); %horizonte de previsao da simulacao;
    step=1;
    n = 1;
    while step <= n_times,

        dif=diff(system_scheduling(step:end));
        ph=min(find(dif~=0));

        if isempty(ph), ph = n_times-step + 1;end

        indice_atual = system_scheduling(step);
        modelo_atual = Model_list{indice_atual};
        if step == 1 || ph == n_times,
            idx_past = inicio - janela:inicio-1;
            idx_future = inicio:inicio+ ph + 2;

            pastSamples{1}=iddata(dataCell{1}(idx_past+1,saida_passado),[dataCell{1}(idx_past
,entradas)],Ts);

            futureSamples{1}=iddata(dataCell{1}(idx_future+1,saida_passado),[dataCell{1}(idx_
future,entradas)],Ts);
            YForecast = forecast(modelo_atual, pastSamples{1}, ph,
futureSamples{1});
            if ph == n_times,
                X= dataCell{1}(idx_past(1:end-step+1),saida_passado);
                X = [X; YForecast.OutputData];
            end
            else
                idx_past = inicio - janela + step :inicio - 1 + step;
                idx_future = inicio+step:inicio+step+ph+3;
                if n == 2,
                    X = dataCell{1}(idx_past(1:end-step+1),saida_passado);
                else
                    X(1:ph_ant)=[];
                end
                X = [X; YForecast.OutputData];
                pastSamples{1}=iddata(X,[dataCell{1}(idx_past,entradas)],Ts);

            futureSamples{1}=iddata(dataCell{1}(idx_future+1,saida_passado),[dataCell{1}(idx_
future,entradas)],Ts);
            YForecast = forecast(modelo_atual, pastSamples{1}, ph,
futureSamples{1});
            if step + ph > n_times,
                X(1:ph)=[];
                X = [X; YForecast.OutputData];
            end
            end
            ph_ant = ph;
            step=step+ph;
            n = n +1;
            %step = step + 1;

```

```

    end
    sse_forecast_fitness = norm(X(end-length(ref)+1:end) - ref)^2;
end

function [sse_forecast,mre_forecast,YForecast] =
retorna_especificacoes(trainData,testData,modelo);

    predHorizon = 1;
    OPT = predictOptions();
    predHorizon = size(testData{1}.InputData,1);
    % previsão com forecast
    YForecast = forecast(modelo, trainData{1}, predHorizon,
testData{1});
    % previsão com predict
    %YPred = predict(modelo, testData{1}, Inf, OPT);

    sse_forecast = norm(YForecast.OutputData -
testData{1}.OutputData)^2;
    mre_forecast = mean(abs((YForecast.OutputData -
testData{1}.OutputData)./testData{1}.OutputData))*100;

end

function [R2Test,R2Train,p_cell,model]=
retornar_modelo(cell_count,trainData,testData)
N = 10;
melhor_polinomio = [];
sse_min = inf;
mre_min = inf; %janelas = [80, 160, 320];
n_entradas = size(testData{1}.InputData,2);
while mre_min > 1
    na = randi(10,1,1); nb = randi(10,1,n_entradas); nc = randi(10,1,1); nk =
zeros(1,n_entradas);
    p_cell{cell_count,1}=na;
    p_cell{cell_count,2}=nb;
    p_cell{cell_count,3}=nc;
    p_cell{cell_count,4}=nk;
    cell_count = cell_count+1;
    model = armax(trainData{1}, [na nb nc nk]);
    %training Data
    predHorizon = 1;
    %compare(trainData{1}, model); % 1-step ahead prediction
    OPT = predictOptions();
    YPred = predict(model, trainData{1}, predHorizon, OPT);
    R2Train=(1- norm(YPred.OutputData-
trainData{1}.OutputData)^2/(norm(trainData{1}.OutputData-
mean(trainData{1}.OutputData))^2));
    %test Data
    predHorizon = size(testData{1}.InputData,1);
    % compare(testData{1}, model,1); % 1-step ahead prediction
    YPred = predict(model, testData{1}, 1, OPT);
    R2Test=(1- norm(YPred.OutputData-
testData{1}.OutputData)^2/(norm(testData{1}.OutputData-
mean(testData{1}.OutputData))^2));
    [sse_model,mre_model,~] =
retorna_especificacoes(trainData,testData,model);

    if sse_model < sse_min,
        best_model_ = model;

```

```
        mre_min = mre_model;  
    end  
end
```

.

.

B

PYTHON

B.1 APÊNDICES I

IMPLEMENTAÇÃO DO MODELO AUTORREGRESSIVO INTEGRADO DE MÉDIAS MÓVEIS SAZONAL COM AUTOREGRESSORAS EXÓGENAS

Este código permite encontrar os modelos que representa a dinâmica para os quatros subsistemas desejado utilizando dados simulados obtidas a partir do *software trnsys*.

```
import pickle
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
import warnings
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
import matplotlib.dates as mdates

# Oculta todos os avisos
warnings.filterwarnings("ignore")

# Diretório onde está o arquivo com os dados
diretorio = r'files/info_30.xlsx'
tabela = pd.read_excel(diretorio)

# Janela 96 dados corresponde a 2 dias de dados(considerando dados de 30 em 30 minutos)
# janela 48 dados corresponde a 1 dia e dados(considerando dados de 30 em 30 minutos)
janela = 112 #
pdq= 6
steps = 24 # passos de previsão há frente
#-----
# 12 passos corresponde a 6 horas, 24 passos corresponde a 12 horas. Dados são de 30 em 30 minutos
#-----
# variáveis dependentes que contem dados reais que vai ser utilizado para fazer previsões
temperatura_piscina = tabela.iloc[janela:janela+steps, 0]
temperatura_coletor = tabela.iloc[janela:janela+steps, 1]
temperatura_pavimento = tabela.iloc[janela:janela+steps, 2]
temperatura_geotermia = tabela.iloc[janela:janela+steps, 3]
data_inicial_dados_reais = tabela.iloc[janela:janela+steps,9]
hora_inicial_dados_reais = tabela.iloc[janela:janela+steps,10]

# Diretório onde está o arquivo pickle com o modelo
modelo_piscina = r'files/modelo_piscina.pkl'
modelo_coletor = r'files/modelo_coletor.pkl'
```

```

modelo_pavimento = r'files/modelo_pavimento.pkl'
modelo_geotermia = r'files/modelo_geotermia.pkl'
# Função para carregar dados
def carregar_dados(diretorio):
    tabela = pd.read_excel(diretorio)

    # variaveis independentes
    colunas = tabela.columns.str.strip()
    temperatura = tabela.iloc[:,janela, 4]
    humidade = tabela.iloc[:,janela, 5]
    velocidade = tabela.iloc[:,janela, 6]

    # variavies dependentes
    temperatura_piscina = tabela.iloc[:,janela, 0]
    temperatura_coletor = tabela.iloc[:,janela, 1]
    temperatura_pavimento = tabela.iloc[:,janela, 2]
    temperatura_geotermia = tabela.iloc[:,janela, 3]

    return (temperatura, humidade, velocidade, temperatura_piscina, temperatura_coletor, temperatura_pavimento,temperatura_geotermia)

# Função para carregar o modelo_piscina SARIMAX a partir de um arquivo pickle
def carregar_modelo_piscina(modelo_piscina):
    with open(modelo_piscina, 'rb') as f:
        modelo_piscina = pickle.load(f)
    return modelo_piscina

# Função para carregar o modelo_coletor SARIMAX a partir de um arquivo pickle
def carregar_modelo_coletor(modelo_coletor):
    with open(modelo_coletor, 'rb') as f:
        modelo_coletor = pickle.load(f)
    return modelo_coletor

# Função para carregar o modelo_pav SARIMAX a partir de um arquivo pickle
def carregar_modelo_pavimento(modelo_pavimento):
    with open(modelo_pavimento, 'rb') as f:
        modelo_pavimento = pickle.load(f)
    return modelo_pavimento

# Função para carregar o modelo_geo SARIMAX a partir de um arquivo pickle
def carregar_modelo_geotermia(modelo_geotermia):
    with open(modelo_geotermia, 'rb') as f:
        modelo_geotermia = pickle.load(f)
    return modelo_geotermia

# Função arimax para gerar varias order e utilizar a melhor order
def armax_v1():
    (temperatura, humidade, velocidade, temperatura_piscina, temperatura_coletor, temperatura_pavimento,temperatura_geotermia) =
    carregar_dados(diretorio)
    # Especificar as ordens a serem testadas
    p_values = list(range(1, pdq)) # ordens autoregressivas
    d_values = list(range(1, pdq)) # ordens de diferenciação
    q_values = list(range(1, pdq)) # ordens de médias móveis
    orders = []
    aic_scores = []
    bic_scores = []
    hqic_scores = []
    endog_variables = []
    exog_variables = []
    # Iterar sobre todas as combinações de ordens e ajustar os modelos SARIMAX
    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p, d, q)
                try:
                    exog_data = pd.DataFrame({'temperatura': temperatura, 'humidade': humidade, 'velocidade': velocidade})
                    exog_names = ', '.join(exog_data.columns)
                    #-----T_Piscina-----
                    modelo_piscina = SARIMAX(endog=temperatura_piscina, exog=exog_data, order=order)
                    modelo_fit_piscina = modelo_piscina.fit()
                    aic = modelo_fit_piscina.aic
                    bic = modelo_fit_piscina.bic
                    hqic = modelo_fit_piscina.hqic
                    orders.append(order)
                    aic_scores.append(aic)
                    bic_scores.append(bic)
                    hqic_scores.append(hqic)
                    endog_variables.append("Temperatura de piscina")

```

```

exog_variables.append(exog_names)

# -----T_Coletor-----
model_coletor = SARIMAX(endog=temperatura_coletor, exog=exog_data, order=order)
model_fit_coletor = model_coletor.fit()
aic = model_fit_coletor.aic
bic = model_fit_coletor.bic
hqic = model_fit_coletor.hqic
orders.append(order)
aic_scores.append(aic)
bic_scores.append(bic)
hqic_scores.append(hqic)
endog_variables.append("Temperatura do coletor")
exog_variables.append(exog_names)

# -----T_Piscina_PAV-----
model_pavimento = SARIMAX(endog=temperatura_pavimento, exog=exog_data, order=order)
model_fit_pavimento = model_pavimento.fit()
aic = model_fit_pavimento.aic
bic = model_fit_pavimento.bic
hqic = model_fit_pavimento.hqic
orders.append(order)
aic_scores.append(aic)
bic_scores.append(bic)
hqic_scores.append(hqic)
endog_variables.append("Temperatura do pavimento")
exog_variables.append(exog_names)

# -----T_Piscina_Geo-----
model_geotermia = SARIMAX(endog=temperatura_geotermia, exog=exog_data, order=order)
model_fit_geotermia = model_geotermia.fit()
aic = model_fit_geotermia.aic
bic = model_fit_geotermia.bic
hqic = model_fit_geotermia.hqic
orders.append(order)
aic_scores.append(aic)
bic_scores.append(bic)
hqic_scores.append(hqic)
endog_variables.append("Temperatura do geotermia")
exog_variables.append(exog_names)

except:
    continue

# Criar um DataFrame com as ordens, valores de AIC, nomes das variáveis endógenas e exógenas
orders_df = pd.DataFrame({'order': orders, 'AIC': aic_scores, 'BIC': bic_scores, 'HQIC': hqic_scores, 'endog': endog_variables, 'exog':
exog_variables})

# Encontrar a melhor ordem com base no menor AIC
best_index = aic_scores.index(min(aic_scores))
best_order = orders[best_index]

# Ajustar o modelo SARIMAX com a melhor ordem para piscina
model_piscina_best = SARIMAX(endog=temperatura_piscina, exog=exog_data, order=best_order)
model_fit_piscina = model_piscina_best.fit()

# Salvar o modelo piscina em um arquivo binário usando pickle
with open(f'files/modelo_piscina.pkl', 'wb') as f:
    pickle.dump(model_fit_piscina, f)

# Ajustar o modelo SARIMAX com a melhor ordem para coletor
model_coletor_best = SARIMAX(endog=temperatura_coletor, exog=exog_data, order=best_order)
model_fit_coletor = model_coletor_best.fit()
# Salvar o modelo coletor em um arquivo binário usando pickle
with open(f'files/modelo_coletor.pkl', 'wb') as f:
    pickle.dump(model_fit_coletor, f)

# Ajustar o modelo SARIMAX com a melhor ordem para pavimento
model_pavimento_best = SARIMAX(endog=temperatura_pavimento, exog=exog_data, order=best_order)
model_fit_pavimento = model_pavimento_best.fit()
# Salvar o modelo pavimento em um arquivo binário usando pickle
with open(f'files/modelo_pavimento.pkl', 'wb') as f:
    pickle.dump(model_fit_pavimento, f)

# Ajustar o modelo SARIMAX com a melhor ordem para geotermia

```

```

model_geoterminha_best = SARIMAX(endog=temperatura_geoterminha, exog=exog_data, order=best_order)
model_fit_geoterminha = model_geoterminha_best.fit()
# Salvar o modelo geoterminha em um arquivo binário usando pickle
with open(f'files/modelo_geoterminha.pkl', 'wb') as f:
    pickle.dump(model_fit_geoterminha, f)

return model_fit_piscina, model_fit_coletor, model_fit_pavimento, model_fit_geoterminha, best_order, orders_df

# Função para calcular o MAPE
mape_list = []
sse_list = []
def calcular_mape(dados_reais, previsoes):
    # Verificar se os tamanhos dos dados reais e previsões são iguais
    if len(dados_reais) != len(previsoes):
        print("Erro: Os tamanhos dos dados reais e das previsões são diferentes.")
        return None

    # Verificar se há valores zero nos dados reais
    if dados_reais.eq(0).any():
        print("Erro: Os dados reais contêm valores zero. O MAPE não pode ser calculado.")
        return None

    # Calcular o MAPE
    mape = np.mean(np.abs((dados_reais.values - previsoes) / dados_reais.values)) * 100
    print("MAPE:", "{:.2f}%".format(mape))

    # Calculando o SSE
    sse = np.sum((dados_reais.values - previsoes) ** 2)
    print("SSE:", "{:.2f}".format(sse))

    return mape

# Função para fazer previsões a alguns passos à frente utilizando o modelo SARIMAX
def fazer_previsoes(modelo, steps):
    # Repetir as últimas observações para criar os valores exógenos
    tabela = pd.read_excel(diretorio)
    ultimas_obs = tabela.iloc[-steps:4:7] # Últimas observações de text, hr e velo
    #print(f'Observações: {ultimas_obs}')
    exog = pd.concat([ultimas_obs] * 1, axis=0) # Replicar as últimas observações steps vezes
    #print(f'EXOG: {exog}')
    previsoes = modelo.get_forecast(steps=steps, exog=exog)
    # print("Previsões a", steps, "passos à frente:")
    # print(previsoes.predicted_mean)
    return previsoes.predicted_mean

# Função para fazer plotes comparando dados reais e previstos para cada subsistemas
def plotar_graficos(datas, dados_reais, dados_previstos, nomes_dados, hora_inicial_dados_reais):
    # Converter hora_inicial_dados_reais de datetime.time para datetime.datetime
    hora_inicial_dados_reais = [datetime.combine(datas[0], hora) for hora in hora_inicial_dados_reais]

    for i in range(len(dados_reais)):
        plt.figure() # Cria uma nova figura para cada gráfico
        plt.plot(hora_inicial_dados_reais, dados_reais[i], marker='o', label='Dados reais') # Dados reais
        plt.plot(hora_inicial_dados_reais, dados_previstos[i], marker='x', label='Dados previstos') # Dados previstos
        plt.xlabel('Data e Hora') # Rótulo do eixo x
        plt.ylabel('Temperatura [°C]') # Rótulo do eixo y
        plt.title(f'Comparação entre dados previstos e reais - {nomes_dados[i]}') # Título do gráfico
        plt.ylim(0, 25) # Define os limites do eixo y de 0 a 25
        plt.legend() # Adiciona a legenda
        plt.xticks(rotation=45) # Rotaciona os rótulos do eixo x para melhor visualização

        # Formatação do eixo x para exibir ano, mês, dia e hora
        formatter = mdates.DateFormatter('%Y-%m-%d %H:%M')
        plt.gca().xaxis.set_major_formatter(formatter)

        # Calcula e apresenta o MAPE
        mape = calcular_mape(dados_reais[i], dados_previstos[i])
        if mape is not None:
            plt.text(0.05, 0.95, f'MAPE: {mape:.2f}%', transform=plt.gca().transAxes, va='top', ha='left')

    plt.tight_layout() # Ajusta o layout do gráfico
    plt.show() # Abre o gráfico atual

# função para fazer plote dados utilizado no treino, dados reais e previsão
def plotar_graficos_treino_reais_previsao(datas, dados_reais, dados_reais_1, dados_previstos, hora_iniciais_reais, nomes_dados):
    # Converter hora_iniciais_reais de datetime.time para datetime.datetime

```

```

hora_iniciais_reais = [[datetime.combine(datas[0], hora) for hora in horas] for horas in hora_iniciais_reais]
#-----
data_inicio_treino = tabela.iloc[:janela, 9].astype(str) # Convertendo para string
hora_inicial_treino = tabela.iloc[:janela, 10].astype(str) # Convertendo para string
# Convertendo data_inicio e hora_inicial para datetime
data_hora_inicio_treino = pd.to_datetime(data_inicio_treino + ' ' + hora_inicial_treino)
#-----
# Plotar em duas figuras separadas
for i, (dados_reais, dados_previstos, hora_inicial_dados_reais, nome_dado) in enumerate(
    zip(dados_reais, dados_previstos, hora_iniciais_reais, nomes_dados)):

    fig, axs = plt.subplots(2, 1, figsize=(10, 7))
    # Plotar dados de treino na segunda figura
    axs[0].plot(data_hora_inicio_treino, dados_reais_1[i], marker='x', label=f'{nome_dado} utilizada para treinar há rede') # dados de
    treino
    axs[0].set_ylabel("Temperatura [°C]") # Rótulo do eixo y
    #axs[0].set_title(f'Dados de treino - Dados reais - Dados Previstos [{nome_dado}]')
    axs[0].set_ylim(0, 25) # Define os limites do eixo y de 0 a 25
    axs[0].legend() # Adiciona a legenda
    axs[0].tick_params(rotation=45) # Rotaciona os rótulos do eixo x para melhor visualização

    # Plotar t_piscina na primeira figura
    axs[1].plot(hora_inicial_dados_reais, dados_reais, marker='*', label=f'Dados reais do passado utilizado para previsão da temperatura')
# Dados reais
    axs[1].plot(hora_inicial_dados_reais, dados_previstos, marker='o', label=f'{nome_dado} prevista para {int(steps/2)} horas há frente')
# Dados previstos
    axs[1].set_xlabel("") # Rótulo do eixo x
    axs[1].set_ylabel("Temperatura [°C]") # Rótulo do eixo y
    axs[1].set_title('')
    axs[1].set_ylim(0, 25) # Define os limites do eixo y de 0 a 25
    axs[1].legend() # Adiciona a legenda
    axs[1].tick_params(rotation=45) # Rotaciona os rótulos do eixo x para melhor visualização

    # Formatação do eixo x para exibir ano, mês, dia e hora
    formatter = mdates.DateFormatter("%Y-%m-%d %H:%M")
    axs[0].xaxis.set_major_formatter(formatter)
    axs[1].xaxis.set_major_formatter(formatter)

    # Calcula e apresenta o MAPE
    mape = calcular_mape(dados_reais, dados_previstos)
    if mape is not None:
        axs[1].text(0.05, 0.95, f'MAPE: {mape:.2f}%', transform=axs[1].transAxes, va='top', ha='left')

    plt.tight_layout() # Ajusta o layout do gráfico
    plt.show() # Abre o gráfico atual
#função para fazer plots das variáveis externas utilizadas
def plot_variaveis_externas():
    tabela = pd.read_excel(diretorio)
    # Variáveis independentes
    temperatura = tabela.iloc[:janela, 4]
    humidade = tabela.iloc[:janela, 5]
    velocidade = tabela.iloc[:janela, 6]
    data_inicio = tabela.iloc[:janela, 9].astype(str) # Convertendo para string
    hora_inicial = tabela.iloc[:janela, 10].astype(str) # Convertendo para string

    # Convertendo data_inicio e hora_inicial para datetime
    data_hora_inicio = pd.to_datetime(data_inicio + ' ' + hora_inicial)

    fig, axs = plt.subplots(3, 1, figsize=(10, 7), sharex=True) # Reduzindo o tamanho da figura

    # Plotando Temperatura
    axs[0].plot(data_hora_inicio, temperatura, color='red', marker='o', label='Temperatura exterior')
    axs[0].set_title("Variáveis externas utilizadas para treinar há rede")
    axs[0].set_ylabel("[°C]")
    axs[0].legend() # Adiciona a legenda
    axs[0].tick_params(axis='y', labelsz=8) # Define o tamanho da fonte do eixo y
    axs[0].set_ylim(0, 25) # Define a escala de 0 a 25 no eixo y

    # Adicionando um eixo secundário para a data no gráfico da temperatura
    axs[0].xaxis.set_major_formatter(mdates.DateFormatter("%d-%m-%Y %H:%M:%S"))
    axs[0].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

    # Plotando Humidade
    axs[1].plot(data_hora_inicio, humidade, color='blue', marker='o', label='Humidade relativa')
    axs[1].set_ylabel("[%]")
    axs[1].legend() # Adiciona a legenda

```

```

axs[1].tick_params(axis='y', labelsz=8) # Define o tamanho da fonte do eixo y
axs[1].set_ylim(0, 150) # Define a escala de 0 a 150 no eixo y

# Adicionando um eixo secundário para a data no gráfico da humidade
axs[1].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[1].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

# Plotando Velocidade
axs[2].plot(data_hora_inicio, velocidade, color='green', marker='o', label='Velocidade do vento')
axs[2].set_ylabel('m/s')
axs[2].legend() # Adiciona a legenda
axs[2].tick_params(axis='y', labelsz=8) # Define o tamanho da fonte do eixo y
axs[2].set_ylim(0, 5) # Define a escala de 0 a 8 no eixo y

# Adicionando um eixo secundário para a data no gráfico da velocidade
axs[2].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[2].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

plt.tight_layout()
plt.show()
# função para fazer plots dos dados utilizados para treinar a rede
def plot_dados_treino(tabela, janela):
    # Variáveis dependentes
    temperatura_piscina = tabela.iloc[janela, 0]
    temperatura_coletor = tabela.iloc[janela, 1]
    temperatura_pavimento = tabela.iloc[janela, 2]
    temperatura_geotermia = tabela.iloc[janela, 3]

    # Variáveis independentes
    data_inicio = tabela.iloc[janela, 9].astype(str) # Convertendo para string
    hora_inicial = tabela.iloc[janela, 10].astype(str) # Convertendo para string

    # Convertendo data_inicio e hora_inicial para datetime
    data_hora_inicio = pd.to_datetime(data_inicio + ' ' + hora_inicial)

fig, axs = plt.subplots(3, 1, figsize=(10, 8), sharex=True) # Reduzindo o tamanho da figura

# Plotando Temperatura da piscina
axs[0].plot(data_hora_inicio, temperatura_piscina, color='purple')
axs[0].set_title('Temperatura da piscina sozinha')
axs[0].set_ylim(0, 25) # Define a escala de 0 a 20 no eixo y
axs[0].set_ylabel('°C')

# Adicionando um eixo secundário para a data no gráfico da temperatura da piscina
axs[0].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[0].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

# Plotando Temperatura do coletor
axs[1].plot(data_hora_inicio, temperatura_coletor, color='orange')
axs[1].set_title('Temperatura da piscina com efeito do coletor')
axs[1].set_ylim(0, 25) # Define a escala de 0 a 25 no eixo y
axs[1].set_ylabel('°C')

# Adicionando um eixo secundário para a data no gráfico da temperatura do coletor
axs[1].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[1].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

# Plotando Temperatura do pavimento
axs[2].plot(data_hora_inicio, temperatura_pavimento, color='brown')
axs[2].set_title('Temperatura da piscina com efeito do pavimento')
axs[2].set_ylim(0, 25) # Define a escala de 0 a 25 no eixo y
axs[2].set_ylabel('°C')

# Adicionando um eixo secundário para a data no gráfico da temperatura do pavimento
axs[2].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[2].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

# Plotando Temperatura da geotermia
axs[3].plot(data_hora_inicio, temperatura_geotermia, color='brown')
axs[3].set_title('Temperatura da piscina com efeito da geotermia')
axs[3].set_ylim(0, 25) # Define a escala de 0 a 25 no eixo y
axs[3].set_ylabel('°C')

# Adicionando um eixo secundário para a data no gráfico da temperatura da geotermia
axs[3].xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y %H:%M:%S'))
axs[3].tick_params(axis='x', rotation=45, labelsz=8) # Diminuir o tamanho da fonte do eixo x

```

```

plt.tight_layout()
plt.show()

# Chamar a função armax_v1 para gerar o modelo SARIMAX, a melhor ordem e os valores de AIC para todas as ordens
best_model_p, best_model_c, best_model_pav, best_model_geo, best_order, orders_df = armax_v1()
#chamar a função para plotar dados de variáveis independentes
plot_variaveis_externas()
#-----
# chamar a função para plotar dados utilizados para treinar a rede
#plot_dados_treino(tabela, janela)
# Salvar os valores de AIC para todas as ordens em um arquivo CSV
orders_df.to_csv(r'files/valores_de_AIC_por_order.csv', index=False)

# Imprimir a melhor ordem e o resumo do modelo
print("Melhor ordem:", best_order)
print("Resumo do modelo para Piscina:")
print(best_model_p.summary())
print("Resumo do modelo para Coletor:")
print(best_model_c.summary())
print("Resumo do modelo para Pavimento:")
print(best_model_pav.summary())
print("Resumo do modelo para Geotermia:")
print(best_model_geo.summary())
# Carregar os modelos
modelo_piscina = carregar_modelo_piscina(modelo_piscina)
modelo_coletor = carregar_modelo_coletor(modelo_coletor)
modelo_pavimento = carregar_modelo_pavimento(modelo_pavimento)
modelo_geotermia = carregar_modelo_geotermia(modelo_geotermia)
# Exemplo de fazer previsões a x passos à frente utilizando as últimas observações como exógenas
previsoes_piscina = fazer_previsoes(modelo_piscina, steps=steps)
previsoes_coletor = fazer_previsoes(modelo_coletor, steps=steps)
previsoes_pavimento = fazer_previsoes(modelo_pavimento, steps=steps)
previsoes_geotermia = fazer_previsoes(modelo_geotermia, steps=steps)
print('PREVISÃO PISCINA:', previsoes_piscina)
print('REAIS PISCINA:', temperatura_piscina)
#print('Piscina: ', modelo_p.data.endog)
print('PREVISÃO COLETOR:', previsoes_coletor)
print('REAIS COLETOR:', temperatura_coletor)
#print('Coletor: ', modelo_c.data.endog)
print('PREVISÃO PAVIMENTO:', previsoes_pavimento)
print('REAIS PAVIMENTO:', temperatura_pavimento)
#print('Pavimento: ', modelo_pav.data.endog)
print('PREVISÃO GEOTERMIA:', previsoes_geotermia)
print('REAIS GEOTERMIA:', temperatura_geotermia)
#print('Geotermia: ', modelo_geo.data.endog)
# Calcular o MAPE
calcular_mape(temperatura_piscina, previsoes_piscina)
calcular_mape(temperatura_coletor, previsoes_coletor)
calcular_mape(temperatura_pavimento, previsoes_pavimento)
calcular_mape(temperatura_geotermia, previsoes_geotermia)
#-----Guardar previsões e dados passados no mesmo arquivo-----
# Criar um DataFrame com os dados reais, as previsões e os dados exógenos
df_resultados = pd.DataFrame({
    'Previsoes_Piscina': previsoes_piscina,
    'Previsoes_Coletor': previsoes_coletor,
    'Previsoes_Pavimento': previsoes_pavimento,
    'Previsoes_Geotermia': previsoes_geotermia
})
# Salvar o DataFrame em um arquivo Excel
df_resultados.to_excel('files/resultados.xlsx', index=False)
#-----
# Variáveis que contêm informações sobre dados reais de cada subsistema incluindo datas e horas iniciais
temperatura_piscina_1 = tabela.iloc[janela, 0]
temperatura_coletor_2 = tabela.iloc[janela, 1]
temperatura_pavimento_3 = tabela.iloc[janela, 2]
temperatura_geotermia_4 = tabela.iloc[janela, 3]
dados_reais_1 = [temperatura_piscina_1, temperatura_coletor_2, temperatura_pavimento_3, temperatura_geotermia_4]
dados_reais = [temperatura_piscina, temperatura_coletor, temperatura_pavimento, temperatura_geotermia]
dados_previstos = [previsoes_piscina, previsoes_coletor, previsoes_pavimento, previsoes_geotermia]
nomes_dados = ['Temperatura de piscina', 'Temperatura do coletor', 'Temperatura do pavimento', 'Temperatura da geotermia']
hora_inicias_reais = [hora_inicial_dados_reais, hora_inicial_dados_reais, hora_inicial_dados_reais, hora_inicial_dados_reais]
data_atual = datetime.now() # Data e hora atual
intervalo = timedelta(minutes=15) # Intervalo de 15 minutos entre os pontos de dados

```

```

datas = [data_atual + i * intervalo for i in range(len(dados_reais[0]))] # Lista de datas correspondentes
#-----
# chamar a função plotar_graficos para ver a comparação entre dados treinos, reais e previstos
plotar_graficos_treino_reais_previsao(datas, dados_reais,dados_reais_1, dados_previstos, hora_inicias_reais, nomes_dados)

## chamar a função plotar_graficos para ver a comparação entre dados reais e previstos
#plotar_graficos(datas, dados_reais, dados_previstos, nomes_dados, hora_inicial_dados_reais)
#-----Print da data e hora dos dados utilizados para fazer previsão-----
print(f'Data Inicio dados reais: {data_inicial_dados_reais}')
print(f'Hora Inicio dados reais: {hora_inicial_dados_reais}')
#-----fim-----

```

B.2 APÊNDICES II

IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO – CÓDIGO 1

Este código define várias funções relacionadas ao carregamento, previsão e avaliação de um modelo autorregressivo integrado de médias móveis sazonal com regressoras exógenas. O foco principal é prever a temperatura em um cenário específico e avaliar essas previsões em relação a um conjunto de pontos de referência (setpoints).

```

import pickle
import numpy as np
# Função para carregar um modelo SARIMAX a partir de um arquivo pickle
def carregar_modelo(modelo_path):
    with open(modelo_path, 'rb') as f:
        modelo = pickle.load(f)
    return modelo
# função para prever temperatura
def fazer_previsoes(modelo, steps, matriz_exog):
    #print(modelo.data.endog)
    previsoes = modelo.get_forecast(steps=steps, exog=matriz_exog)
    previsao = previsoes.predicted_mean.tolist()
    return previsao
# função para avaliar cenario baseado em setpoint e temperatura prevista
def avaliar_cenario(temperatura_prevista, setpoints):
    if len(temperatura_prevista) != len(setpoints):
        raise ValueError("O número de sistemas e setpoints não coincide.")
    # Calcular a medida de fitness
    diff_abs_media = np.mean(np.abs(np.array(temperatura_prevista) - setpoints)/np.array(setpoints))
    print(f'MAPE: {diff_abs_media}')
    return diff_abs_media
#função para prever cenario
def preve_cenario(individuo, lista_modelo,tabela,janela_previsao):
    contador = 0
    for passos in individuo:
        steps = 1
        matriz_exog = tabela.iloc[janela_previsao - 100 + contador:janela_previsao - 100 + steps+ contador, [3, 4, 5]]
        if contador == 0:
            endog = tabela.iloc[janela_previsao - 200 + contador :janela_previsao - 100+ contador, 0]
            lista_modelo[passos].data.endog = endog
        else:
            # Supondo que endog seja uma Series do pandas
            temp = endog.copy()
            # Deslocando os valores para cima por um índice, preenchendo a última posição com NaN
            endog.iloc[0:-1] = temp.iloc[1:]
            # Atribuindo a primeira previsão à primeira posição (ajuste conforme necessário)
            endog.iloc[-1] = previsoes[0]
            # Atualizando o modelo com o novo endog
            lista_modelo[passos].data.endog = endog
            lista_modelo[passos].data.exog = tabela.iloc[janela_previsao - 200 + contador:janela_previsao - 100+ contador, [3, 4, 5]]
            previsoes = fazer_previsoes(lista_modelo[passos], steps, matriz_exog)
            contador = contador + 1
    return Endog

```

conforme o apêndice B.2 Apêndice II, contendo quatro funções: 1º - carregar modelo, 2º - fazer previsões, 3º - avaliar cenário e 4º - prever cenário.

1 - `carregar_modelo (modelo_path)`;

Esta função chamada `carregar_modelo` que carrega um modelo SARIMAX a partir de um arquivo pickle. A função abre o arquivo pickle em modo binário, carrega o modelo do arquivo e retorna o modelo carregado. A função assume que o arquivo pickle contém um objeto SARIMAX válido e usa o módulo pickle para carregar o modelo. A função abre e fecha automaticamente o arquivo pickle usando a instrução `with`.

2 - `fazer_previsoes (modelo, steps, matriz_exog)`;

Esta função chamada `fazer_previsoes` que usa um modelo SARIMAX para prever a temperatura. A função recebe um modelo treinado, o número de etapas para prever e uma matriz opcional de variáveis exógenas como entrada. A função usa o método `get_forecast` do modelo para gerar previsões, extrai a média prevista e retorna uma lista de previsões de temperatura. A função assume que o modelo foi treinado corretamente e que os dados de entrada estão no formato correto.

3 - `avaliar_cenario (temperatura_prevista, setpoint)`;

Esta função chamada `avaliar_cenario` que avalia um cenário com base em setpoints e temperatura prevista. A função verifica se o número de sistemas e setpoints coincide, calcula o MAPE (Mean Absolute Percentage Error) e imprime o valor do MAPE. A função também retorna o valor calculado do MAPE. A função assume que as temperaturas previstas e os setpoints estão na mesma ordem e representam o mesmo número de sistemas. A função usa o MAPE como medida de avaliação, mas você pode modificá-la para usar outras métricas de erro.

4 - `prever_cenario (individuo, lista_modelo, tabela, janela_previsao)`.

Esta função vai prevê um cenário de temperatura com base em um indivíduo (uma sequência de passos) e uma lista de modelos SARIMAX.

A função recebe uma lista de passos a serem previstos, uma lista de modelos SARIMAX treinados, um DataFrame contendo dados históricos e um índice de janela de previsão como entrada. A função usa um loop `for` para iterar sobre os passos do indivíduo, faz previsões para cada passo usando o modelo SARIMAX correspondente e atualiza a série de temperatura endog com as previsões. A função retorna a série endog final contendo as previsões de temperatura para todo o cenário. A função assume que a lista `individuo` contém valores válidos para os passos de tempo, que a lista, `lista_modelo` contém modelos SARIMAX

treinados para cada passo e que o DataFrame contém dados históricos e variáveis exógenas no formato correto.

Em segundo lugar, foi desenvolvido um código em Python conforme o apêndice B.3 Apêndice III, contendo quatro funções: 1º - calcular fitness da população, 2º - selecionar os melhores indivíduos da população, 3º - cruzamento dos indivíduos(crossover) e 4º - fazer mutuação.

1 - `cal_pop_fitness (models_list, pop, setpoints, tabela, janela_previsao);`

A função recebe cinco argumentos:

- `models_list`: Uma lista contendo modelos SARIMAX para diferentes horizontes de previsão.
- `pop`: Uma matriz NumPy que representa a população (cada linha é um indivíduo, cada elemento é o passo para aquele indivíduo).
- `setpoints`: Uma lista de setpoints (temperaturas desejadas).
- `tabela`: Um DataFrame contendo dados históricos de temperatura e variáveis exógenas.
- `janela_previsao`: Índice da janela de previsão na tabela.

A função obtém o tamanho da população e o número de passos por indivíduo usando `pop.shape`. A função inicializa um vetor fitness com zeros para armazenar os valores de fitness.

A função itera sobre cada indivíduo na população (linha da matriz `pop`).

- Seleciona o indivíduo atual usando `pop[i]`.
- Chama a função `preve_cenario` da biblioteca `previsao_library` para prever a temperatura com base no indivíduo, modelos, tabela e janela de previsão.
- Chama a função `avaliar_cenario` da biblioteca `previsao_library` para calcular o fitness do cenário previsto, utilizando os últimos elementos da previsão (`temp_prevista.iloc[-len(individuo):]`) e os setpoints. O valor de fitness é armazenado em `fitness[i]`.

Normalização do fitness (opcional):

Este bloco verifica se todos os valores de fitness são iguais ao máximo. Se não forem, a função normaliza os valores de fitness calculando a distância absoluta ao valor máximo e dividindo pelo somatório dessas distâncias. Esta normalização torna os valores de fitness relativos e comparáveis.

A função imprime o fitness da população e retorna o vetor fitness.

2 – `select_mating_pool (pop, fitness, num_parents);`

A função `select_mating_pool` seleciona os melhores indivíduos da geração atual para serem pais da próxima geração. ela recebe três argumentos:

- `pop`: Uma matriz NumPy que representa a população (cada linha é um indivíduo, cada elemento é o passo para aquele indivíduo).
- `fitness`: Um vetor NumPy contendo o valor de fitness de cada indivíduo.
- `num_parents`: O número de indivíduos a serem selecionados como pais.

A função inicializa uma matriz `parents` com zeros para armazenar a população de pais. A função usa um loop `for` para selecionar `num_parents` indivíduos da população.

A função encontra o índice do indivíduo com o maior valor de fitness usando `np.argmax(fitness)`.

A função copia o indivíduo selecionado para a população de pais utilizando o índice na matriz `pop` y a slice `[:]` para copiar todos os passos.

Evitar seleção repetida (opcional):

A função define o valor de fitness do indivíduo selecionado como um valor muito negativo (por exemplo, `-float('inf')`) para evitar que ele seja selecionado novamente no loop.

3 – `crossover (parents, offspring_size)`;

Aplica o crossover de dois pontos para gerar descendentes. Essa função recebe dois argumentos:

- `parents`: Uma matriz NumPy que representa a população de pais (pais x passos).
- `offspring_size`: Uma tupla contendo o tamanho desejado da população de descendentes (número de indivíduos, número de passos por indivíduo).

A função gera `offspring_size[0]` descendentes utilizando crossover de dois pontos.

A função escolhe aleatoriamente dois pontos de crossover no cromossomo dos pais.

A função cria um novo descendente combinando a primeira parte do cromossomo do primeiro pai com a parte entre os pontos de crossover do segundo pai.

A função repete o processo para gerar todos os descendentes.

4 – `mutation (offspring, n_models, num_mutations=1)`.

Esta função introduz diversidade genética na população de descendentes aplicando mutações aleatórias. Ela recebe 3 argumentos:

- `offspring (np.ndarray)`: Uma matriz NumPy representando a população de descendentes (cada linha é um indivíduo, cada elemento é o passo para aquele indivíduo).
- `n_models (int)`: O número de modelos SARIMAX disponíveis (usado para selecionar novos valores durante a mutação).

- `num_mutations` (int, optional): O número de genes a serem mutados em cada descendente (padrão: 1).

Seleção de Índices de Mutação: Para cada descendente a função `np.random.choice` seleciona aleatoriamente `num_mutations` índices distintos dentro do intervalo válido do número de passos do descendente para aplicar a mutação.

Aplicação da Mutação: Um loop itera sobre os índices de mutação selecionados e para cada gene selecionado: O valor do gene no descendente é substituído por um valor aleatório entre 1 e `n_models` (inclusive) usando `np.random.randint`.

Retorno: A função retorna a matriz `offspring` modificada, contendo a população de descendentes mutada.

Em terceiro e último, foi desenvolvido um código em python conforme o apêndice 5.5.4, contendo apenas uma função: `GA_main` (`n_ind`, `n_time_slots`, `num_generations`, `models_list`, `setpoints`, `tabela`, `janela_previsao`).

A função principal `GA_main` define o algoritmo genético (AG) para otimizar a sequência de modelos de controle de temperatura em uma piscina.

O código dentro do bloco `if __name__ == "__main__":` é executado apenas quando o script é executado diretamente, não quando importado como módulo. Aqui está uma análise detalhada do código:

- 1) Definição de parâmetros:
 - `steps`: Define o número de intervalos de tempo por sistema (1 neste caso).
 - `comp_cenario`: Calcula o número total de sistemas ($12 * 1 = 12$).
 - `setpoint`: Define a temperatura desejada para cada sistema (13°C).
 - `setpoints`: Cria uma lista de valores `setpoint` para todos os sistemas usando compreensão de lista.

- 2) Inicialização dos parâmetros do AG:
 - `n_ind`: Define o tamanho da população (número de soluções candidatas) para 100.
 - `n_time_slots`: Especifica o número de intervalos de tempo (12 neste caso).
 - `num_generations`: Define o número de gerações para o AG ser executado (15).
- 3) Carregamento de Dados e Modelos:
 - `diretorio`: Define o caminho para o arquivo Excel contendo dados históricos ('../files/info_30.xlsx').
 - `tabela`: Lê os dados históricos do arquivo Excel para um DataFrame Pandas.
 - `janela_previsao`: Define o tamanho da janela de previsão (300 intervalos de tempo).

- `modelo_piscina_path`, `modelo_coletor_path`, `modelo_pavimento_path` e `modelo_geotermia_path`: Definem os caminhos para os arquivos pickle contendo os modelos SARIMAX treinados para cada sistema.

- `modelo_piscina`, `modelo_coletor`, `modelo_pavimento` e `modelo_geotermia`: Carrega os modelos SARIMAX treinados dos arquivos pickle.

- `modelos_dict`: Cria um dicionário que mapeia IDs de modelo (1, 2, 3, 4) aos respectivos modelos SARIMAX.

- `lista_modelos`: Cria um objeto Models e atribui o `modelos_dict` ao seu atributo `models`.

4) Execução do Algoritmo Genético:

- `best_individual`: Chama a função `GA_main` para executar o algoritmo genético, passando os seguintes argumentos:

`n_ind`: Tamanho da população (100)

`n_time_slots`: Número de intervalos de tempo (12)

`num_generations`: Número de gerações (15)

`lista_modelos`: Objeto contendo os modelos SARIMAX

`setpoints`: Lista de valores de temperatura desejados

`tabela`: Dados históricos de temperatura

`janela_previsao`: Tamanho da janela de previsão (300).

5) Impressão do melhor indivíduo:

- O código imprime o array `best_individual`, que representa a sequência de IDs de modelo (1, 2, 3, 4) que atingiu a maior aptidão durante o processo de otimização do AG.

Aqui está uma análise detalhada do código dentro da função `GA_main()`:

➤ Parâmetros da função:

- `n_ind`: Número de indivíduos (soluções candidatas) na população.

- `n_time_slots`: Número de intervalos de tempo para os quais a temperatura precisa ser controlada.

- `num_generations`: Número de gerações para executar o AG.

- `models_list`: Lista contendo os modelos SARIMAX treinados para cada sistema de controle de temperatura.

- `setpoints`: Valores de temperatura desejados para cada intervalo de tempo.

- `tabela`: (Provavelmente) Dados históricos de temperatura usados para previsão do modelo.

- `janela_previsao`: Tamanho da janela de previsão (número de etapas de tempo a prever).

- Variáveis locais:
 - `sol_per_pop`: Número de indivíduos por população (igual a `n_ind`).
 - `num_parents_mating`: Número de indivíduos selecionados como pais para a próxima geração (metade do tamanho da população).
 - `best_outputs`: Lista para armazenar o valor máximo de aptidão de cada geração (para acompanhar o progresso).
 - `thermo_sys_models`: Lista de modelos de sistema de controle de temperatura disponíveis em `models_list`.
 - `n_models`: Número total de modelos de sistema de controle de temperatura disponíveis.
 - `elite_on`: Flag booleana para ativar o elitismo (manter o melhor indivíduo em cada geração).
 - `pop_size`: Tupla que define o tamanho da população (número de indivíduos, número de genes por indivíduo).
 - `new_population`: Array NumPy que representa a população inicial de soluções candidatas (indivíduos).

- Loop principal (Gerações):

- Inicialização:

Cria a população inicial usando a função `create_pop`.

A seção `models_performance` provavelmente destina-se a armazenar o valor de aptidão de cada indivíduo, mas não é preenchida.

- Iteração pelas Gerações:

Loops para `num_generations`:

1° - Cálculo de Aptidão: Calcula a aptidão de cada indivíduo na população atual usando `ga.cal_pop_fitness`. Esta função provavelmente avalia o quão bem a sequência de modelos de um indivíduo prevê a temperatura desejada enquanto considera a eficiência energética.

`best_outputs` armazena a aptidão máxima para cada geração.

`best_ind` identifica o indivíduo com a maior aptidão na geração atual.

2° - Seleção de Pais: Seleciona os melhores indivíduos como pais para a próxima geração usando `ga.select_mating_pool`.

3° - Crossover: Gera descendentes usando crossover entre pais com `ga.crossover`.

4° - Mutação: Introduz variações aleatórias usando `ga.mutation`.

5° - Criação de Nova População: Combina pais para formar a nova população.

B.3 APÊNDICES III

IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO – CÓDIGO 2

Este código implementa um algoritmo genético para otimizar a seleção de modelos para prever temperaturas em sistemas térmicos. O algoritmo envolve várias etapas, incluindo cálculo de fitness, seleção de pais, crossover e mutação.

```
import numpy
import previsao_library

#Essa função calcula o valor de fitness (adaptação) de cada indivíduo na população atual.
def cal_pop_fitness(models_list, pop, setpoints, tabela, janela_previsao):
    # Calculating the fitness value of each solution in the current population.
    # Input arguments:
    # - models_list, a vector where each element is a thermo system model representative
    # - pop, a two dimensional array defining the chromosomes for the individuals in the population

    n_ind, n_time_slots = (numpy.size(pop, axis = 0), numpy.size(pop, axis = 1))

    #define fitness according to thermo system scheduling over the time_slots
    fitness = numpy.zeros(n_ind)

    for i in range(n_ind):
        temp_previsata=previsao_library.preve_cenario(pop[i], models_list.models, tabela, janela_previsao)
        fitness[i] = previsao_library.avaluar_cenario(temp_previsata.iloc[-len(pop[i]):], setpoints) #this function returns the fitness value for the
        candidate

    if all(fitness==fitness.max()):
        pass
    else:
        A = abs(fitness - fitness.max())
        fitness = abs(fitness - fitness.max()) / A.sum()
    print(f'Fitness: {fitness}')
    return fitness

#Essa função seleciona os melhores indivíduos da geração atual para serem pais da próxima geração.
def select_mating_pool(pop, fitness, num_parents):
    # Selecting the best individuals in the current generation as parents for producing the offspring of the next generation.
    parents = numpy.zeros((num_parents, pop.shape[1]))
    for parent_num in range(num_parents):
        max_fitness_idx = numpy.where(fitness == numpy.max(fitness))
        max_fitness_idx = max_fitness_idx[0][0]
        parents[parent_num, :] = pop[max_fitness_idx, :]
        fitness[max_fitness_idx] = -99999999999
    return parents

#Essa função aplica o crossover de dois pontos para gerar descendentes.
def crossover(parents, offspring_size):
    offspring = numpy.zeros(offspring_size)
    # Choose two points to apply 2 point crossover. For each point swap position for the gene in
    # parent 1 with relation to the position of the same gene in parent 2, if applicable
    #crossover_point = numpy.uint8(numpy.random.random_integers(1)/2)
    crossover_point = [0,0]
    crossover_point[0] = numpy.random.randint(offspring_size[1])
    if crossover_point[0] == 0: crossover_point[0] += 1
    crossover_point[1] = numpy.random.randint(offspring_size[1])
    if crossover_point[1] == 0: crossover_point[1] += 1
    crossover_point = numpy.sort(crossover_point)

    for k in range(offspring_size[0]):
        # Index of the first parent to mate.
        parent1_idx = k%parents.shape[0]
        # Index of the second parent to mate.
        parent2_idx = (k+1)%parents.shape[0]
        # The new offspring will have its first half of its genes taken from the first parent.
        val1 = parents[parent1_idx, crossover_point[0]]
        #val2= parents[parent2_idx, crossover_point[1]]
        offspring[k, :] = parents[parent1_idx]
```

```

    offspring[k,crossover_point[0]:crossover_point[1]]=parents[parent2_idx][crossover_point[0]:crossover_point[1]]

    return offspring

#Essa função aplica mutação aleatória em alguns genes dos descendentes.
def mutation(offspring,n_models, num_mutations=1):
    # Mutation replaces a number of num_mutations genes (in a random selected time_slot)

    for idx in range(offspring.shape[0]):

        gene_idx = numpy.random.permutation(offspring.shape[1])[0:num_mutations] #location to change the gene
        for j in range(num_mutations):
            #print("OFFSPRING:", offspring)
            offspring[idx, gene_idx[j]] = numpy.random.permutation(n_models)[0] + 1

    return offspring

```

B.4 APÊNDICES III

IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO – CÓDIGO 3

Este código implementa um algoritmo genético para otimizar a programação de um sistema térmico, utilizando uma biblioteca de modelos de previsão (previsao_library) e funções específicas para a operação do algoritmo (GA_V1.ga_v1_ecopool).

```

import numpy
import pandas as pd
import GA_V1.ga_v1_ecopool as ga
import previsao_library
from GA_V1.models_library_v1 import Models
from GA_V1.models_library_v1 import create_pop

def GA_main(n_ind, n_time_slots, num_generations, models_list, setpoints, tabela, janela_previsao):
    """
    Genetic algorithm parameters:
        Mating pool size
        Population size
    """
    sol_per_pop = n_ind
    num_parents_mating = int(sol_per_pop * 0.5)
    best_outputs = []

    #define the number of existing thermo systems
    #thermo_sys_models = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # the list of thermo systems
    thermo_sys_models = models_list.models
    n_models = len(thermo_sys_models)
    elite_on = True

    #ler lista de modelos do arquivo
    *****usar funcao de abertura do arquivo pickle correspondente*****

    # Defining the population size.
    pop_size = (sol_per_pop,
                n_time_slots) # The population will have sol_per_pop chromosome where each chromosome has num_slots genes.
                # each position has an index to the model to fit at that time slot

    # Creating the initial population.

    new_population = create_pop(n_models, n_time_slots, n_ind)

    for generation in range(num_generations):
        #print("Generation : ", generation)
        # Measuring the fitness of each chromosome in the population.

        fitness = ga.cal_pop_fitness(models_list, new_population, setpoints, tabela, janela_previsao)

        best_outputs.append(numpy.max(fitness))

```

```

best_ind = new_population[numpy.where(max(fitness)==fitness)]

# Selecting the best parents in the population for mating.
parents = ga.select_mating_pool(new_population, fitness,
                               num_parents_mating)

# Generating next generation using crossover.
offspring_crossover = ga.crossover(parents,
                                   offspring_size=(pop_size[0] - parents.shape[0], n_time_slots))
#offspring_crossover = new_population
# Adding some variations to the offspring using mutation.
offspring_mutation = ga.mutation(offspring_crossover, n_models, num_mutations=1)

# Creating the new population based on the parents and offspring.
new_population[0:parents.shape[0], :] = parents
new_population[parents.shape[0]:, :] = offspring_mutation

if elite_on: #reproduction with elitism selection
    new_population[0, :] = best_ind[0]

if False:
    print("Fitness")
    print(fitness)

# The best result in the current iteration.
print("Best result : ", numpy.max(fitness))

print(best_ind)
input("ESpera")

print("Parents")
print(parents)

print("Crossover")
print(offspring_crossover)

print("Mutation")
print(offspring_mutation)

# Getting the best solution after iterating through all generations.
# At first, the fitness is calculated for each solution in the final generation.
fitness = ga.cal_pop_fitness(models_list, new_population, setpoints, tabela, janela_previsao)

# Then return the index of that solution corresponding to the best fitness.
best_match_idx = numpy.where(fitness == numpy.max(fitness))
best_individual = new_population[best_match_idx[0], :][0]

#print("Best solution : ", best_individual)
#print("Best solution fitness : ", fitness[best_match_idx[0]][0])

return best_individual

if False:

    matplotlib.pyplot.plot(best_outputs)
    matplotlib.pyplot.xlabel("Iteration")
    matplotlib.pyplot.ylabel("Fitness")
    matplotlib.pyplot.show()

if __name__ == "__main__":
    steps = 1
    comp_cenario = 12 * steps # Defina o número de sistemas
    setpoint = 13 # Defina o valor do setpoint para cada sistema
    # Crie a lista de setpoints usando uma lista por compreensão
    setpoints = [setpoint] * comp_cenario # Lista de setpoints para cada sistema

    n_ind = 100
    n_time_slots = comp_cenario # specifies the number of 3 hours slots considered
    num_generations = 15

# Diretório onde está o arquivo com os dados
diretorio = r'./files/info_30.xlsx'
tabela = pd.read_excel(diretorio)

```

```

janela_previsao = 300
# Diretório onde estão os arquivos pickle com os modelos
modelo_piscina_path = r'../files/modelo_piscina.pkl'
modelo_coletor_path = r'../files/modelo_coletor.pkl'
modelo_pavimento_path = r'../files/modelo_pavimento.pkl'
modelo_geotermia_path = r'../files/modelo_geotermia.pkl'
indivíduo = [1, 3, 3, 2, 4]

# Carregar os modelos
modelo_piscina = previsao_library.carregar_modelo(modelo_piscina_path)
modelo_coletor = previsao_library.carregar_modelo(modelo_coletor_path)
modelo_pavimento = previsao_library.carregar_modelo(modelo_pavimento_path)
modelo_geotermia = previsao_library.carregar_modelo(modelo_geotermia_path)
# lista de modelos
modelos_dict = {1: modelo_piscina, 2: modelo_coletor, 3: modelo_pavimento, 4: modelo_geotermia}
lista_modelos=Models()
lista_modelos.models= modelos_dict
best_individual = GA_main(n_ind, n_time_slots, num_generations, lista_modelos, setpoints, tabela, janela_previsao)
#print(type(best_individual[0]))
print("Best individual : ", best_individual)

```

B.5 APÊNDICES IV

REGISTRAR UTILIZADOR

```

@app.route("/register", methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        firstname = request.form.get('firstname');lastname = request.form.get('lastname');email = request.form.get('email')
        password = request.form.get('password');username = request.form.get('username');phone = request.form.get('phone')
        country = request.form.get('country');city = request.form.get('city')
        lista_user = str(username) + "," + str(password) + "," + str(firstname) + "," + str(lastname) + "," + str(email) + "," + str(phone) + "," + str(country) + "," + str(city) + "\n"

        #Criar um ficheiro txt com lista de utilizadores
        utilizadores = r"files/user.txt"
        with open(utilizadores, "a") as arquivo:
            arquivo.write(lista_user)
            arquivo.close()

    return render_template("register.html")

```

B.6 APÊNDICES V

VALIDAÇÃO DO UTILIZADOR

```

@app.route('/', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        session.pop('user_id', None)
        username = request.form['username']
        password = request.form['password']

        user = [x for x in users if x.username == username]
        if user != []:
            user = user[0]

        if user and user.password == password:
            session['user_id'] = user.id
            return redirect(url_for('profile'))
        else:
            return render_template('login_e.html')

    return redirect(url_for('login'))

return render_template('login.html')

```

B.7 APÊNDICES VI

READ AUTÔMATO

```

# Função utilizada para ler parametros no automata
@app.route("/read_plc", methods=['GET', 'POST'])
def read_plc():
    # S representa a palavra Sonda e T representa a temperatura
    #lista = {0: "S1_Nivel_cobertura_piscina", 4: "S2_T1_retorno_piscina", 5: "S2_T1_retorno_piscina_Lim_Min", 6: "S2_T1_retorno_piscina_Lim_Max", 8: "S3_T2_ida_piscina"}
    lista = {0: "Nivel_cobertura_piscina", 4: "T1_retorno_piscina", 8: "T2_ida_piscina", 16: "T4_Pav_secundario",
            20: "T5_primario_PCM", 24: "T6_secundario_PCM", 28: 'PH', 32: "T7_primario_coletores",
            36: "T8_secundario_coletores", 40: "T9_Ida_Retorno Dissipador de calor", 44: "T10_Retorno_Ida Dissipador de calor",
            48: "T12_piscina", 52: "nivel_tanque_compensacao", 68: "Temp_exterior", 72: "Humidade_relativa", 80: "Vel_vento",
            84: "Pressao_atmosferica", 88: "Radiacao_solar", 97: "Taxa_pluviosidade", 126: "Cobertura_pos_atual", 143: "Caudal_coletores",
            148: "Caudal_PCM", 153: "Caudal_pavimento", 158: "Caudal_agua_rede", 163: "Caudal_lavagem_filtros",
            206: "Setpoint", 207: "Coletor Solar Hora Inicio", 205: "Cenario", 229: "Duracao"}

    fator = 10
    valor = []
    #print(request.method)
    if request.method == 'GET':
        for i in lista.keys():
            read = client.read_holding_registers(address=i, port=502, count=2, unit=1)
            valor.append(':{:1f}'.format(int(read.registers[0]) / fator)) # dados de todas as leituras
        temp_retorn_piscina = valor[1]; temp_ida_piscina = valor[2]; radiacao_solar = valor[17]
        t12_piscina = valor[11]; setpoint = valor[25]; T7_primario_coletores = valor[7]; T8_secundario_coletores=valor[8]
        T6_secundario_PCM = valor[5]; Radiacao_solar = valor[17]; Vel_vento = valor[15]
        Temp_exterior = valor[13]; Humidade_relativa = valor[14]; cenario = valor[27]
        if cenario == '0.0':
            cenario = 'Stall (no water circulation)'
        elif cenario == '3.0':
            cenario = 'Solar => PCM'
        elif cenario == '9.0':
            cenario = 'Solar => Pool'
        elif cenario == '10.0':
            cenario = 'PCM => Pool'
        elif cenario == '11.0':
            cenario = 'Solar and PCM => Pool'
        elif cenario == '12.0':
            cenario = 'Collecting floors => Pool'
        elif cenario == '13.0':
            cenario = 'Solar and Collecting floors => Pool'
        elif cenario == '14.0':
            cenario = 'PCM and Collecting floors => Pool'
        elif cenario == '15.0':
            cenario = 'Solar and PCM e Collecting floors => Pool'
        elif cenario == '17.0':
            cenario = 'Solar => Geo'

        elif cenario == '24.0':
            cenario = 'Geo => Pool'
        elif cenario == '25.0':
            cenario = 'Solar and Geo => Pool'
        elif cenario == '26.0':
            cenario = 'PCM and Geo => Pool'
        elif cenario == '27.0':
            cenario = 'Solar and PCM e Geo => Pool'
        elif cenario == '28.0':
            cenario = 'Collecting floors and Geo => Pool'
        elif cenario == '29.0':
            cenario = 'Solar and Collecting floors and Geo => Pool'
        elif cenario == '30.0':
            cenario = 'PCM and Collecting floors and Geo=> Pool'
        elif cenario == '31.0':
            cenario = 'Solar and PCM and Collecting floors and Geo => Pool'

    return render_template("read_plc.html", username = g.user.username, msg22=cenario, msg8 = setpoint, msg10 = t12_piscina, msg14 = temp_retorn_piscina, msg13 = temp_ida_piscina

```

B.8 APÊNDICES VII

WRITE AUTÓMATO

```

@app.route("/write_plc", methods=['GET', 'POST'])
def write_plc():
    # 206 corresponde ao registro 207 no plc. 205 corresponde ao registro 206 no plc. 227 corresponde ao registro 228 no plc
    lista_registos = {206: "setpoint", 205: "cenario", 228:"duration"} #to apply for the final PLC

    if request.method == 'POST':
        setpoint = request.form.get('setpoint')
        cenario = request.form.get('cenario')
        duration = request.form.get('duration')
        #print("SET" + str(len(setpoint)))

        #if setpoint is not None and diferencial is not None and cenario is not None and duration is not None :
        if len(setpoint) > 0 and len(cenario) > 0 and len(duration) > 0:
            if setpoint.find('.') != -1:
                inteiro = setpoint[0:setpoint.find('.')]
                decimal = setpoint[setpoint.find('.') + 1:]
                setpoint = int(inteiro) * fator + int(decimal)
                ccenario = int(cenario) * fator
                duration = int(duration) * fator
            else:
                setpoint = int(setpoint) * fator
                ccenario = int(cenario) * fator
                dduration = int(duration) * fator

        lista_chaves = list(lista_registos.keys())
        client.write_register(lista_chaves[0], setpoint)
        client.write_register(lista_chaves[1], ccenario)
        client.write_register(lista_chaves[2], dduration)

        data_hora_atual = datetime.now()
        data_unixtime= int(data_hora_atual.timestamp())
        # Guardar os valores das variaveis: inicio_t, cenario e duration em csv
        variaveis = [data_unixtime, cenario, duration]
        registro_variaveis = r'files/registro_variaveis.csv'
        registro_var = pd.read_csv(r'files/registro_variaveis.csv', header=None, names=['data_unixtime', 'Cenario', 'Duration'])
        lista_cenarios = registro_var.values[:, 1]
        cenario = int(cenario)
        if list(lista_cenarios[:]).count(cenario) == 0: # trata do append
            with open(registro_variaveis, 'a', newline='') as registro_variaveis_csv:
                writer_object = csv.writer(registro_variaveis_csv)
                writer_object.writerow(variaveis)
        else:
            # trata de escrever onde ja existe
            pos = list(lista_cenarios[:]).index(cenario)
            registro_var.values[pos, 0] = data_unixtime
            registro_var.values[pos, 2] = duration

            with open(registro_variaveis, 'r') as registro_variaveis_csv:
                reader = csv.reader(registro_variaveis_csv)
                lines = list(reader)
                lines[pos] = [data_unixtime, cenario, duration]
            with open(registro_variaveis, 'w', newline='') as registro_variaveis_csv:
                writer_object = csv.writer(registro_variaveis_csv)
                writer_object.writerows(lines)
    else:
        print("MISSING VALUES")
        return render_template('modo_funcionamento.html', username=g.user.username, error = True)
    return render_template('write_plc.html', username = g.user.username)

```


C.

HTML

C.1 APÊNDICES I

PÁGINA DE REGISTO DE UTILIZADOR

```
<!DOCTYPE html>
<html lang="en">
<html lang="pt-PT"></html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="refresh" content="60"><!-- atualizar a pagina depois de 60 segundos-->
  <title>Register</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/styles.css') }}">
</head>
<body>
  <nav>
    <ul>
      <li><a href="/logout">Return</a></li>
    </ul>
  </nav>
  <div class="container mt-4" style="margin-bottom:15%">
    <div class="row align-items-center" style="margin-top:15%">
      <div class="col-md-5 mx-auto col-lg-5">
        <div class="login">
          <form action="" method="post">
            <div class="login_1">
              <input type="text" size="14" class="btn btn-outline-dark" name="firstname" placeholder="First name">
              <label for="firtsname"></label>
              <input type="text" size="14" class="btn btn-outline-dark" name="lastname" placeholder="Last name">
              <label for="lastname"></label>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
```

```

<div class="login_1">
  <input type="text" size="14" class="btn btn-outline-dark" name="email" placeholder="E-mail">
  <label for="email"></label>
  <input type="password" size="14" class="btn btn-outline-dark" name="password" placeholder="Password">
  <label for="floatingPassword"></label>
</div>
<div class="login_1">
  <input type="text" size="14" class="btn btn-outline-dark" name="username" placeholder="Username">
  <label for="username"></label>
  <input type="phone" size="14" class="btn btn-outline-dark" name="phone" placeholder="Phone">
  <label for="phone"></label>
</div>
<div class="login_1">
  <input type="text" size="14" class="btn btn-outline-dark" name="country" placeholder="Country">
  <label for="country"></label>
  <input type="text" size="14" class="btn btn-outline-dark" name="city" placeholder="City">
  <label for="city"></label>
  <br><br>
  <button class="btn btn-outline-dark" type="submit">Register</button>
</div>
</form>
</div>
</div>
</div>
</body>
</html>

```

C.2 APÊNDICES II

PÁGINA DE LOGIN

```

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<html lang="pt-PT">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="refresh" content="10"><!-- atualizar a página depois de 10 segundos -->
  <title>Login</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/styles.css') }}">
</head>
<body>
  <div class="container mt-4" style="...">
    <div class="row align-items-center" style="...">
      <div class="col-md-5 mx-auto col-lg-5">
        <div class="login">
          <form action="" method="post">
            <div class="login_1">
              <input type="text" size="14" class="btn btn-outline-dark" name="username" placeholder="Username">
              <label for="username"></label>
            </div>
            <br>
            <div class="login_1">
              <input type="password" size="14" class="btn btn-outline-dark" name="password" placeholder="Senha">
              <label for="floatingPassword"></label>
            <br><br>
            <button class="btn btn-outline-dark" type="submit">Login</button>
            <button class="btn btn-outline-dark"><a style="text-decoration:none; color:black;" href="register">Registrar</a> </button>
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>

```

C.3 APÊNDICES II

PÁGINA READ AUTÔMATO(PLC)

```
<!DOCTYPE html>
<html lang="en">
<html lang="pt-PT">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="refresh" content="60">
<title>READ PLC</title>
<link rel="stylesheet" href="{ url_for('static', filename='css/bootstrap.min.css') }">
<link rel="stylesheet" type="text/css" href="{ url_for('static', filename='css/stylas.css') }">
</head>
<body id="read_plc">
<ul id="nav">
<li><a href="menu" id="ho">Smart menu</a></li>
<li><a href="read_plc" id="p1">Real-time data in the pool</a></li>
<li><a href="read_emcs" id="p2">Real-time data on the platform</a></li>
<li><a href="dashboard" id="p3">Dashboard</a></li>
{% if username=="admin": %}
<li><a href="modo_funcionamento" id="p4">Operating mode</a></li>
{% endif %}
<li><a href="alarm_history" id="p5">Alarm history</a></li>
<li><a href="register" id="p6">Registro</a></li>
<li><a href="/logout">Logout </a></li>
</ul>
```

```
<div class="container mt-4" style="margin-bottom:12%">
<div class="row align-items-center">
<div class="col-md-10 mx-auto col-lg-5">
<div class="read">
<form action="/read_plc" method="get">
<div class="esquerda">
<h4 class="titulo_pisc" style="background-color:white">Piscina - Coletor solar - PCM - Coletor solar de pavimento</h4>
<input class="btn btn-outline-dark" name="setpoint" placeholder="Temperatura da piscina: {{msg10}}°C" size="45"
style="background-color:white" type="text">
<label for="temperatura_piscina"></label>

<input class="btn btn-outline-dark" name="setpoint" placeholder="Temperatura de ida na piscina: {{msg13}}°C" size="45"
style="background-color:white" type="text">
<label for="temperatura_piscina_ida"></label>

<input class="btn btn-outline-dark" name="setpoint" placeholder="Temperatura de retorno na piscina: {{msg14}}°C" size="45"
style="background-color:white" type="text">
<label for="temperatura_piscina_retorno"></label>

<input class="btn btn-outline-dark" name="diferencial" placeholder="Temperatura de ida para o coletor solar: {{msg11}}°C" size="45"
style="background-color:white" type="text">
<label for="temperatura_coletor_ida"></label>

<input class="btn btn-outline-dark" name="setpoint" placeholder="Temperatura de retorno no coletor solar: {{msg15}}°C" size="45"
style="background-color:white" type="text">
```

```

</div>
<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="temperatura_coletor_retorno"></Label>

<input class="btn btn-outline-dark" name="cenario" placeholder="Ponto de regulação da temperatura da piscina: {{msg8}}°C" size="45"
style="background-color:white" type="text">
<Label for="setpoint_piscina"></Label>
</div>
<br>
<div class="direita">
<h4 class="titulo" style="background-color:white">Estação meteorológica</h4>
<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="humidade_relativa"
placeholder="Humidade relativa: {{msg12}}%">
<Label for="humidade_relativa"></Label>

<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="S18_estacao_mete_temp"
placeholder="Temperatura exterior: {{msg16}}°C">
<Label for="S18_estacao_mete_temp"></Label>

<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="Veloc_vent_estacao_mete_temp"
placeholder="Velocidade do vento: {{msg17}}m/s">
<Label for="Veloc_vent_estacao_mete_temp"></Label>

<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="S23_estacao_meteo_radiacao_solar"
placeholder="Radiação solar: {{msg21}}w/m²">

<Label for="S23_estacao_meteo_radiacao_solar"></Label>

<input type="text" style="background-color:white" size="45" class="btn btn-outline-dark" name="cenario"
placeholder="Scenario : {{msg22}} ">
<Label for="cenario"></Label>
</div>
</form>
</div>
</div>
</div>
</body>
</html>

```

C.4 APÊNDICES III

PÁGINA WRITE AUTÓMATO

```

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<html lang="pt-PT">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="refresh" content="60">
<title>CHANGE PARAMETERS</title>
<link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/styles.css') }}">
</head>
<body id="modo_funcionamento">
<ul id="nav">
<li><a href="menu" id="ho">Home</a></li>
<li><a href="read_plc" id="p1">Real-time data in the pool</a></li>
<li><a href="read_emcms" id="p2">Real-time data on the platform</a></li>
<li><a href="dashboard" id="p3">Dashboard</a></li>
{% if username=="admin": %}
<li><a href="modo_funcionamento" id="p4">0Operating mode</a></li>
{% endif %}
<li><a href="alarm_history" id="p5">Alarm history</a></li>
<li><a href="register" id="p6">Registro</a></li>
<li><a href="/Logout">Logout </a></li>
</ul>
{% if error != True: %}

```

```

<div class="container mt-4" style="margin-bottom:15%">
<div class="row align-items-center" style="margin-top:15%">
  <div class="col-md-20 mx-auto col-lg-5">
    <div class="write">
      <form action="/write_plc" method="post">
        <div class="form-floating mb-3">
          <input type="text" size="50" class="btn btn-outline-dark" name="setpoint" placeholder="Set-point [°C]">
          <label for="setpoint"></label>
        </div>
        <div class="form-floating mb-3">
          <select name="cenario" class="btn btn-outline-dark">
            <option value="" disabled selected hidden>Escolha o cenário de funcionamento</option>
            <option value="0">Stall (no water circulation)</option>
            <option value="3">Solar => PCM</option>
            <option value="9">Solar => Pool</option>
            <option value="10">PCM => Pool</option>
            <option value="11">Solar and PCM => Pool</option>
            <option value="12">Collecting floors => Pool</option>
            <option value="13">Solar and Collecting floors => Pool</option>
            <option value="14">PCM and Collecting floors => Pool</option>
            <option value="15">Solar and PCM and Collecting floors => Pool</option>
            <option value="17">Solar=> Geo</option>
            <option value="24">Geo => Pool</option>
            <option value="25">Solar and Geo => Pool</option>
            <option value="26">PCM and Geo => Pool</option>
            <option value="27">Solar and PCM and Geo => Pool</option>
            <option value="28">Collecting floors and Geo => Pool</option>
            <option value="29">Solar and Collecting floors and Geo => Pool</option>
            <option value="30">PCM and Collecting floors and Geo=> Pool</option>
            <option value="31">Solar and PCM and Collecting floors and Geo => Pool</option>
          </select>
        </div>
        <div class="form-floating mb-3">
          <input type="text" size="50" class="btn btn-outline-dark" name="duration" placeholder="Duration (max: 3270) [min]">
          <label for="duration"></label>
        </div>
        <button class="btn btn-outline-dark" type="submit">alterar o cenário</button>
        <button class="btn btn-outline-dark" <a style="text-decoration:none; color:black;" href="sistem">Alterar o sistema</a></button>
      </form>
    </div>
  </div>
</div>
{% else: %}
<div class="container mt-4" style="margin-bottom:15%">
  <div class="row lign-items-center" style="margin-top:15%">
    <div class="col-md-1 mx-auto col-lg-5">
      <div class="error">
        <form action="/menu" method="post">
          <h1><label for="Missing Values"> Missing Values </label>
          <button class="btn btn-outline-dark" type="submit"></button></h1>
        </form>
      </div>
    </div>
  </div>
</div>
{% endif %}
</body>
</html>

```

