

Faculdade de Engenharia da Universidade do Porto

Melhoria da Gestão de Utilidades Industriais: Desenvolvimento de um Sistema de Monitorização e Alarmística

Tomás Bernardo de Sá Magalhães



Mestrado em Engenharia Mecânica
Especialização em Engenharia e Gestão da Produção

Orientador: Prof. Américo Azevedo

17 de julho de 2026

Melhoria da Gestão de Utilidades Industriais: Desenvolvimento de um Sistema de Monitorização e Alarmística

Tomás Bernardo de Sá Magalhães

Mestrado em Engenharia Mecânica
Especialização em Engenharia e Gestão da Produção

Abstract

This dissertation was carried out in an industrial context at the production unit of the Super Bock Group, in the field of monitoring the consumption of industrial utilities. The efficient management of utilities such as water, electricity, carbon dioxide and compressed air depends on reliable data and on the ability to detect consumption deviations quickly, two requirements that the existing manual process did not adequately ensure.

The work began with a diagnosis of the manual process, which revealed data dispersed across several sources, a significant manual effort in compilation and validation, and difficulty in detecting reading errors and abnormal consumption in a timely manner. Based on this diagnosis, a set of solutions was developed for the automated processing and validation of data, for alerting, and for visualisation, applied to the consumption of all monitored utilities. The data processing and validation were automated with Power Automate and Office Scripts, integrated into the company's existing ecosystem. Reference limits were defined for each utility, abnormal readings were flagged automatically through conditional formatting, and a daily alert report was sent by email.

The monitoring dashboard was redesigned in Power BI, with key performance indicators and data filtering. The results show that the objective of the work was achieved. The automatic flagging of inconsistent readings and out-of-range consumption no longer depends exclusively on human observation, reducing the reliance on manual and visual detection of consumption anomalies. According to the operational validation and the experience reported by the staff member responsible for the process, the monthly analysis was reduced from more than twenty pages to six, and the preparation time from about one day to roughly one hour.

These figures constitute an operationally validated estimate and not a consolidated quantitative result. Automating data processing and integrating it into Business Intelligence tools therefore contributed to more reliable and responsive monitoring of industrial utility consumption, compared with the previous manual process.

Keywords: Industrial utilities, Consumption monitoring, Process automation, Business Intelligence, Power BI, Anomaly detection

Resumo

A presente dissertação foi realizada em contexto industrial na unidade produtiva do Super Bock Group, no âmbito da monitorização do consumo de utilidades industriais. A gestão eficiente de utilidades como a água, a eletricidade, o dióxido de carbono e o ar comprimido depende da fiabilidade dos dados e da capacidade de detetar desvios de consumo de forma rápida, dois requisitos que o processo manual existente não assegurava de forma adequada.

O trabalho começou pelo diagnóstico do processo manual, que revelou dados dispersos por várias fontes, um esforço manual considerável na compilação e validação e dificuldade em detetar atempadamente erros de leitura e consumos anómalos. Com base neste diagnóstico, foi desenvolvido um conjunto de soluções para o tratamento e validação automáticos dos dados, para a alarmística e para a visualização, aplicadas ao consumo de todas as utilidades monitorizadas. O tratamento e a validação dos dados foram automatizados com recurso a Power Automate e Office Scripts, integrados no ecossistema já adotado pela empresa. Foram definidos limites de referência para cada utilidade, as leituras anómalas passaram a ser sinalizadas automaticamente por formatação condicional e foi enviado um relatório diário de alertas por correio eletrónico.

O dashboard de monitorização foi reformulado em Power BI, com indicadores de desempenho e filtragem dinâmica. Os resultados mostram que o objetivo do trabalho foi cumprido. A sinalização automática de leituras inconsistentes e de consumos fora dos limites de referência deixou de depender exclusivamente da observação humana, reduzindo a dependência da deteção manual e visual de anomalias de consumo. De acordo com a validação operacional e a experiência reportada pela colaboradora responsável pelo posto de trabalho, a análise mensal reduziu-se de mais de vinte páginas para seis, e o tempo de preparação de cerca de um dia para aproximadamente uma hora.

Estes valores constituem uma estimativa validada operacionalmente e não um resultado quantitativo consolidado. A automatização do tratamento de dados e a sua integração em ferramentas de Business Intelligence contribuíram, assim, para uma monitorização mais fiável e mais reativa do consumo de utilidades industriais, face ao processo manual anterior.

Palavras-chave: Utilidades industriais, Monitorização de consumos, Automatização de processos, Business Intelligence, Power BI, Deteção de anomalias

Agradecimentos

Diz-se que uma tese se escreve sozinha à frente do computador, mas não é verdade. Por trás destas páginas há muita gente que merece um obrigado e é isso que vou fazer aqui.

Aos meus pais, obrigado por tudo (e tudo é mesmo muita coisa). Por acreditarem e me fazerem sempre acreditar, especialmente em alturas em que desistir parecia uma opção.

Ao meu irmão, pela companhia e por ser, ao seu jeito, dos meus maiores apoios mesmo quando parecia não ser.

Aos meus avós, pelo carinho sem medida e por torcerem por mim como se eu já fosse engenheiro há anos. Por todos os almoços na sua companhia e por estarem sempre muito presentes na minha vida.

À minha namorada, por pegar no complicado e torná-lo simples, por transformar as chatices em piada e por acreditar em mim tanto ou mais do que eu acredito nela.

Às senhoras do bar, que nunca me deixaram passar fome, especialmente nos dias em que o dinheiro ficava em casa e pelo carinho que sempre tiveram comigo.

Às senhoras da secretaria, pela simpatia e pela paciência com as minhas (muitas, muitas) dúvidas. Perderam a conta das vezes em que lá apareci, e mesmo assim receberam-me sempre com um sorriso.

A todos os professores que ao longo deste percurso me conheceram melhor e me fizeram acreditar que eu ia conseguir.

À Super Bock Group e a todos os colaboradores com quem trabalhei, pela forma como me acolheram e por me terem feito sentir parte da equipa desde o primeiro dia.

E, por fim, um último obrigado ao Professor Américo Azevedo por ter aceite ser meu orientador nesta última etapa e me ajudar a terminar este ciclo.

Conseguimos.

A todos, do fundo do coração: obrigado.

Tomás Magalhães

*“Life is what happens to you
while you’re busy making other plans.”*

John Lennon

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Pergunta de Investigação	2
1.3	Objetivos	2
1.4	Âmbito e Limitações do Estudo	3
1.5	Metodologia	3
1.6	Descrição da Empresa	4
1.6.1	História	4
1.6.2	Marcas	5
1.7	Estrutura	5
2	Fundamentação Teórica	7
2.1	Contextualização Histórica da Cerveja	7
2.2	Cadeia de Produção da Cerveja	8
2.2.1	Maltagem	8
2.2.2	Fabrico	9
2.2.3	Enchimento	9
2.3	Utilidades Industriais	10
2.3.1	Água	10
2.3.2	Dióxido de Carbono (CO ₂)	11
2.3.3	Energia Térmica	11
2.3.4	Elettricidade	12
2.3.5	Ar Comprimido	12
2.4	Sustentabilidade e Monitorização das Utilidades	12
2.5	Digitalização e Indústria 4.0 na Gestão de Energia	13
2.6	Monitorização de Utilidades com BI	14
3	Análise e Validação dos Dados de Consumo das Utilidades	17
3.1	Processo Atual de Recolha de Dados	17
3.1.1	Limitações do Processo Atual	17
3.2	Definição de Limites de Referência	18
3.3	Soluções Implementadas	21
3.3.1	Verificação Automática por Formatação Condicional	21
3.3.2	Automatização da Importação de Dados	22
3.3.3	Sistema de Alertas por Correio Eletrónico	25
3.4	Processo de Validação e Detecção de Anomalias	29

4	Reformulação do Dashboard de Monitorização em Power BI	31
4.1	Limitações do Dashboard Existente	31
4.2	Objetivos da Reformulação	33
4.3	Arquitetura da Solução	34
4.4	Página de Análise por Linha de Produção	35
4.5	Reformulação da Análise Mensal de KPIs	35
4.5.1	Motivação e Impacto Operacional	35
4.5.2	Processo de Desenvolvimento e Versões Intermédias	36
4.5.3	Estrutura da Solução Final	39
4.5.4	Lógica Visual dos Gráficos KPI	45
4.5.5	Indicadores de Variação Percentual com Código de Cor	45
4.5.6	Implementação Técnica — Medidas DAX	45
4.6	Discussão e Mais-valias	46
5	Conclusões e Trabalhos Futuros	49
5.1	Conclusões	49
5.2	Trabalhos Futuros	51
5.2.1	Integração de Controlo Estatístico de Processos (CEP)	51
	Referências	53
A	Automatização da Importação de Dados	55
A.1	Fase 1 — Macro VBA	55
A.2	Fase 2 — Office Scripts (Power Automate)	58
A.2.1	Leitura do ficheiro de origem (Ler Leituras)	58
A.2.2	Escrita no ficheiro de destino (Escrever Leituras)	60
B	Sistema de Alertas por Correio Eletrónico	65
B.1	Fase 1 — Script Python	65
B.2	Fase 2 — Office Scripts (Power Automate)	73
B.2.1	Análise de anomalias por utilidade	74
B.2.2	Geração do relatório HTML	79
C	Medidas DAX Implementadas no Power BI	85
C.1	Linhas de Enchimento	85
C.2	Adega	87
C.3	Produção	89
C.4	Fabrico	92

Lista de Figuras

1.1	Sequência cronológica da história do Super Bock Group	5
1.2	Marcas Super Bock Group	6
2.1	Linha de Produção da Cerveja	8
3.1	Processo de recolha e tratamento de dados de consumo das utilidades. . . .	17
3.2	Desvio máximo definido para a Água Captada (diferença entre dias conse- cutivos).	18
3.3	Limites máximos definidos para a Água Processo e Água Rede Geral (ETA). .	18
3.4	Limites máximos definidos para a Água Rede Geral no Coletor de Enchimento. .	18
3.5	Limites máximos definidos para a Água Desmineralizada e Água para Apro- veitamentos Térmicos.	19
3.6	Limites máximos definidos para a Água Industrial no Coletor ETA, com distinção entre período de verão e restante ano.	19
3.7	Limites máximos definidos para a Água Industrial no Coletor de Enchimento. .	19
3.8	Limites máximos definidos para o Ar Comprimido no Coletor de Centrais. . .	19
3.9	Limites máximos definidos para o Ar Comprimido no Coletor de Adegas. . .	19
3.10	Limites máximos definidos para o Ar Comprimido no Coletor de Enchimento. .	19
3.11	Limites máximos definidos para o Ar Comprimido no Armazém Automático e para o CO ₂ no Coletor Central, com distinção entre período de verão e restante ano.	20
3.12	Limites máximos definidos para o CO ₂ no Coletor de Adegas.	20
3.13	Limites máximos definidos para o CO ₂ no Coletor de Enchimento.	20
3.14	Desvios máximos definidos para o Biogás, Vapor Fábrica, Vapor Adegas e Vapor Enchimento.	20
3.15	Desvios máximos e limites definidos para o Vapor Centrais, Vapor ETAR/Restaurante e Aproveitamentos Térmicos.	20
3.16	Desvios máximos definidos para a Água Quente por Linhas (total e com cogeração).	21
3.17	Formatação condicional: leitura inferior ao valor do dia anterior.	21
3.18	Formatação condicional: consumo diário superior ao limite definido.	21
3.19	Formatação condicional: limite sazonal diferenciado para o período de verão. .	22
3.20	Formatação condicional: limite para o período adjacente ao verão.	22
3.21	Formatação condicional: consumo acumulado superior ao valor de referência. .	22
3.22	Fluxo de atualização diária no Power Automate.	25
3.23	Formato do relatório de alertas diário enviado por correio eletrónico.	27
3.24	Fluxo Power Automate para envio de alertas de utilidades: sete ações se- quenciais.	28

3.25	Processo de registo e validação dos dados de consumo das utilidades.	29
4.1	Dashboard original para a Água no Enchimento, distribuído por páginas independentes.	32
4.2	Apresentação Mensal apresentada em reunião da empresa para avaliação e discussão de resultados.	33
4.3	Página inicial do dashboard reformulado, com acesso direto às Análises Mensais de Enchimento, Adega e Fabrico através de botões de navegação. .	35
4.4	Versão intermédia com cartões de detalhe por baixo de cada gráfico, apresentando o valor atual, o valor do ano anterior e a variação percentual: gráficos YTD e Mês ainda separados.	36
4.5	Versão intermédia em grelha 2×3: gráficos YTD e Mês separados por linha, com percentagens de variação por baixo de cada gráfico, sem codificação de cor por utilidade.	37
4.6	Versão intermédia com fundo colorido por utilidade: introdução da codificação cromática para distinção visual, com gráficos YTD e Mês ainda apresentados separadamente.	37
4.7	Versão intermédia com Mês e YTD integrados no mesmo gráfico: aproximação à solução final, com dois pares de barras por KPI e percentagens de variação por baixo, mas sem legendagem unificada nem navegação por linha.	38
4.8	Versão pré-final: estrutura com botões de seleção de linha, filtros de Ano e Mês e gráficos Mês/YTD integrados. Nesta iteração os filtros de contadores ainda se encontravam expostos na parte inferior, sendo posteriormente ocultados na versão definitiva.	38
4.9	Separadores das páginas de Análise Mensal criadas no Power BI: AM Enchimento, AM Linhas 4 & 7, AM Linha Vini, AM Produção, AM Adega e AM Fabrico.	39
4.10	Versão final da página de análise mensal de KPIs para as Linhas de Enchimento: botões de seleção de linha no topo, filtros de Ano e Mês, botões de sub-contador (Aproveitamentos Térmicos, Captações), cinco KPIs (Água, Térmica, Eletricidade, Ar Comprimido e CO ₂) com gráficos Mês/YTD integrados e indicadores de variação percentual com código de cor.	40
4.11	Página dedicada às Linhas 4 e 7, com layout diferenciado: os KPIs de Água, Térmica, Ar Comprimido e CO ₂ são apresentados de forma agregada para o conjunto das duas linhas, enquanto o KPI de Eletricidade é desagregado individualmente para cada linha, refletindo as diferenças no perfil de consumo elétrico.	41
4.12	Página dedicada à Linha Vini, com layout simplificado: são apresentados apenas os três KPIs monitorizados nesta linha (Água, Eletricidade e CO ₂), refletindo a ausência de consumos de Energia Térmica e Ar Comprimido. A navegação de regresso às Linhas de Enchimento é assegurada pelo botão <i>Pág. Anterior</i>	42
4.13	Página de Produção da análise mensal de KPIs, com visão agregada dos consumos de Adega e Fabrico combinados, apresentando os cinco KPIs (Água, Térmica, Eletricidade, Ar Comprimido e CO ₂) e botões de navegação para a análise detalhada por área.	43
4.14	Página de Adegas da análise mensal de KPIs, com os cinco KPIs completos.	44
4.15	Página de Fabrico da análise mensal de KPIs; a ausência do KPI de CO ₂ reflete a irrelevância deste indicador para a área de fabrico.	44

Lista de Tabelas

3.1	Macros VBA desenvolvidas por ficheiro de utilidade.	23
3.2	Lógica de atualização aplicada pela macro VBA consoante a data da linha.	23
3.3	Lógica de atualização do Office Script (Power Automate): versão final.	24
3.4	Tipos de alerta emitidos pelo sistema de notificação automática.	26
4.1	Comparação entre o processo anterior e as soluções implementadas.	47

Lista de Acrónimos

BI	Business Intelligence
CEP	Controlo Estatístico de Processos
CIP	Cleaning in Place
CO₂	Dióxido de Carbono
CUFP	Companhia União Fabril Portuense
DAX	Data Analysis Expressions
EnPI	Energy Performance Indicator
ETA	Estação de Tratamento de Água
ETAR	Estação de Tratamento de Águas Residuais
FEUP	Faculdade de Engenharia da Universidade do Porto
ISO	International Organization for Standardization
KPI	Key Performance Indicator
SPLY	Same Period Last Year
YTD	Year-to-Date

Capítulo 1

Introdução

O presente trabalho é desenvolvido no âmbito da realização da Unidade Curricular de Dissertação do Mestrado em Engenharia Mecânica, na área de especialização em Engenharia e Gestão da Produção, da Faculdade de Engenharia da Universidade do Porto (FEUP). A dissertação foi realizada em contexto industrial na unidade produtiva do Super Bock Group, localizada em Leça do Balio, e incide na melhoria e automatização das ferramentas de monitorização e gestão de dados associadas às utilidades industriais.

O trabalho centra-se na análise e na melhoria dos processos de monitorização e tratamento dos dados de consumo das utilidades industriais da unidade de produção de cerveja, com vista a uma gestão mais eficiente destes recursos.

Neste capítulo é apresentado o enquadramento geral da dissertação, incluindo o contexto em que o trabalho se insere e a definição dos principais objetivos do estudo. É também realizada uma breve caracterização da empresa onde o projeto foi desenvolvido, bem como a apresentação da estrutura e organização do presente documento.

1.1 Enquadramento

A gestão eficiente das utilidades industriais é cada vez mais importante na indústria, tanto por razões económicas como de sustentabilidade e de eficiência energética. Uma monitorização adequada dos consumos permite identificar desvios operacionais, otimizar a utilização dos recursos e apoiar a tomada de decisão na gestão das instalações.

Na unidade industrial do Super Bock Group, o acompanhamento dos consumos de utilidades, como água, eletricidade, dióxido de carbono e ar comprimido, é realizado através da recolha periódica de dados provenientes de diversos contadores distribuídos ao longo da instalação. No entanto, a existência de múltiplas fontes de dados e a necessidade de realizar várias etapas de registo e tratamento de informação podem dificultar a análise eficiente dos consumos e atrasar a identificação de possíveis anomalias ou ineficiências operacionais.

Neste contexto, é relevante desenvolver abordagens que melhorem a forma como os dados de consumo são organizados, analisados e utilizados na gestão das utilidades industriais.

1.2 Pergunta de Investigação

O problema enunciado no Capítulo 3 assenta em dois eixos: a fiabilidade e a reatividade da monitorização do consumo de utilidades industriais. O enquadramento apresentado evidencia que a gestão de fiabilidade diz respeito à confiança nos dados depois de tratados e validados, sobretudo na deteção de erros de leitura e de consumos anómalos. A reatividade traduz a rapidez com que um desvio é sinalizado e disponibilizado para análise. Com base nestes dois eixos, formula-se a seguinte pergunta de investigação.

Pergunta central: «Em que medida a automatização do tratamento de dados e a integração de ferramentas de *Business Intelligence* permitem melhorar a fiabilidade e a reatividade na deteção de anomalias de consumo de utilidades industriais, face ao processo manual existente?»

Pergunta subsidiária: «Que critério de definição de limites de referência se revela adequado para a sinalização automática de consumos anómalos no contexto operacional em estudo?»

1.3 Objetivos

Este trabalho procura demonstrar de que forma a automatização do tratamento de dados e a sua integração em ferramentas de *Business Intelligence* tornam a monitorização do consumo de utilidades industriais mais fiável e mais reativa. Os objetivos estão formulados como resultados a alcançar, sendo as tarefas necessárias descritas na metodologia.

Objetivo geral: determinar de que forma a automatização do tratamento de dados e a sua integração em ferramentas de *Business Intelligence* contribuem para uma monitorização mais fiável e mais reativa do consumo de utilidades industriais.

Para concretizar este objetivo geral, definem-se os seguintes objetivos específicos:

- Caracterizar o processo existente e diagnosticar as suas limitações de fiabilidade e de reatividade;
- Estabelecer critérios de definição de limites de referência adequados ao contexto operacional para a sinalização de consumos anómalos;
- Demonstrar em que medida a automatização reduz o esforço manual e o risco de erro;
- Avaliar o contributo das ferramentas de *Business Intelligence* para a rapidez e a fiabilidade da deteção de desvios.

1.4 Âmbito e Limitações do Estudo

O presente trabalho delimita a sua área de atuação à monitorização e análise avançada dos dados operacionais das utilidades industriais da unidade de produção de cerveja, abrangendo o conjunto das utilidades monitorizadas, como a água, a eletricidade, o dióxido de carbono e o ar comprimido. Em várias destas utilidades, as ineficiências operacionais, como fugas nas redes de distribuição ou longos períodos de operação em vazio, são difíceis de identificar sem um sistema de monitorização contínuo e analítico.

Neste contexto, o âmbito da dissertação incide sobre o diagnóstico do processo manual de recolha e tratamento dos dados, a automatização da sua validação e da respetiva alarmística, bem como a reformulação das ferramentas de *Business Intelligence* utilizadas na monitorização dos consumos. O trabalho privilegia a digitalização e valorização da informação existente, com vista à identificação de anomalias e ao apoio à tomada de decisão. Ficam excluídas do âmbito desta dissertação as análises estatísticas avançadas, remetidas para trabalhos futuros, bem como quaisquer intervenções físicas ou mecânicas nos equipamentos de produção, incluindo afinações em compressores ou alterações à infraestrutura de tubagens.

1.5 Metodologia

O trabalho seguiu uma abordagem estruturada, que progride do diagnóstico para a solução. Cada fase responde a uma necessidade identificada na fase anterior, e as escolhas técnicas justificam-se pelo contexto operacional e pelo ecossistema digital da empresa.

Começou-se pelo diagnóstico do processo manual existente, uma vez que só a compreensão das suas limitações de fiabilidade e de reatividade permite dimensionar a solução e evitar automatizar um processo deficiente. Esta fase consistiu no levantamento do sistema de recolha, registo e tratamento dos dados de consumo, com identificação dos erros de leitura e dos consumos anómalos não detetados.

Em seguida, procedeu-se à recolha e organização dos dados históricos de consumo das diversas utilidades. A análise destes dados serviu de base à definição de limites de referência para cada utilidade.

Para a sinalização de consumos anómalos optou-se por limites de referência conservadores, próximos dos valores máximos observados, em vez de métodos estatísticos como percentis (P95/P99) ou múltiplos do desvio padrão. Esta opção deve-se à sua simplicidade e à sua interpretação operacional direta. Um limite próximo do máximo histórico é facilmente compreendido e validado pelas equipas, ao passo que um critério estatístico exigiria pressupostos sobre a distribuição dos dados e uma leitura menos imediata em contexto de operação [1].

A automatização do tratamento e da validação dos dados foi desenvolvida com recurso a

Power Automate e Office Scripts, em vez de soluções como VBA ou Python. Esta escolha deve-se à integração nativa no ecossistema Microsoft 365 já adotado pela empresa, à manutenção centralizada na nuvem e à dispensa de infraestrutura local e de dependências instaladas em postos individuais, fatores que favorecem a continuidade e a transferência da solução.

A estruturação e a visualização dos consumos foram realizadas em Power BI, pela sua integração no mesmo ecossistema e pela capacidade de incorporar indicadores chave de desempenho (KPIs) e filtragem dinâmica por linha de produção e por período. A sinalização automática de leituras fora dos limites de referência foi materializada através de formatação condicional por cor, que torna o desvio perceptível de imediato, sem exigir interpretação adicional por parte do utilizador.

A aplicação de Controlo Estatístico de Processos, com recurso a cartas de controlo aplicadas ao sistema de ar comprimido, não foi concretizada neste trabalho. Esta análise permitiria avaliar a estabilidade do processo e apoiar a identificação de desvios sistemáticos de consumo, encontrando-se proposta na secção de Trabalhos Futuros.

1.6 Descrição da Empresa

O Super Bock Group é uma das principais empresas portuguesas do setor das bebidas, com atividade centrada na produção e comercialização de cervejas, águas e sidras. O grupo está também presente nos mercados dos refrigerantes, vinhos e sangrias.

A empresa comercializa um vasto portefólio de produtos nos mercados nacional e internacional. Para além das bebidas, desenvolve atividades complementares como a produção e comercialização de malte e o turismo.

A atividade industrial e logística distribui-se por várias localizações em território nacional, o que assegura a capacidade produtiva e a cobertura do mercado.

1.6.1 História

A origem do atual Super Bock Group remonta a 1890, com a fundação da Companhia União Fabril Portuense das Fábricas de Cerveja e Bebidas Refrigerantes (CUFP), na cidade do Porto. Desde a sua criação, a empresa foi evoluindo e expandindo a sua atividade, acompanhando o desenvolvimento da indústria cervejeira em Portugal.

Ao longo do século XX, o grupo registou um crescimento significativo, caracterizado pela expansão das suas unidades produtivas e pelo lançamento de novas marcas e produtos no mercado. Foi também neste período que surgiu a marca Super Bock, cuja comercialização teve início na década de 1930 e que se viria a tornar uma das principais referências do setor cervejeiro em Portugal.

Na década de 1960 foi inaugurado o Centro de Produção de Leça do Balio, que atualmente constitui uma das principais unidades industriais do grupo. Posteriormente, durante a

década de 1970, ocorreu a nacionalização do setor cervejeiro em Portugal, levando à transformação da CUFP na UNICER, União Cervejeira E.P., designação pela qual a empresa ficou conhecida durante várias décadas.

Em 2017, a empresa adotou a designação de Super Bock Group, refletindo a sua presença em vários segmentos do mercado das bebidas. A Figura 1.1 sintetiza a sequência cronológica desta evolução.

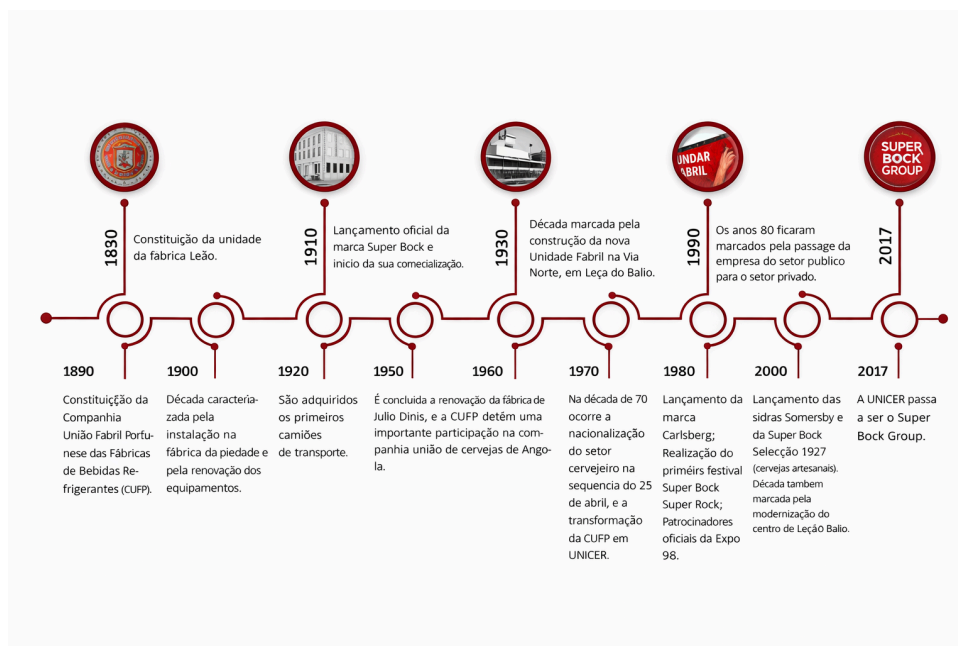


Figura 1.1: Sequência cronológica da história do Super Bock Group

1.6.2 Marcas

O Super Bock Group dispõe de um amplo portefólio, organizado em várias categorias de bebidas. A cerveja continua a ser uma das principais atividades do grupo, que comercializa ainda águas engarrafadas, sidras, refrigerantes, vinhos e sangrias.

Este portefólio resulta da estratégia multimarca da empresa, que procura responder a diferentes segmentos de mercado e perfis de consumidores.

A Figura 1.2 apresenta alguns dos principais produtos e respetivas marcas pertencentes ao portefólio do Super Bock Group.

1.7 Estrutura

A presente dissertação encontra-se organizada em vários capítulos, de forma a apresentar o enquadramento do problema, o desenvolvimento do trabalho e a análise dos resultados obtidos.



Figura 1.2: Marcas Super Bock Group

No Capítulo 1 é apresentado o enquadramento geral do trabalho, incluindo a contextualização do tema, a pergunta de investigação, os objetivos da dissertação e uma breve descrição da empresa onde o estudo foi desenvolvido.

O Capítulo 2 corresponde à revisão bibliográfica, onde são apresentados os principais conceitos teóricos relevantes para o desenvolvimento do trabalho. Neste capítulo é abordada a evolução histórica da produção de cerveja, a descrição do processo produtivo e o papel das utilidades industriais na indústria cervejeira. São ainda discutidos aspetos relacionados com a sustentabilidade e com a monitorização de consumos energéticos em ambientes industriais.

No Capítulo 3 é realizada a caracterização do sistema produtivo e das utilidades da unidade industrial em estudo. São descritas as principais áreas de consumo, bem como o sistema atualmente utilizado para a recolha e tratamento dos dados de consumo das utilidades, e diagnosticadas as suas limitações de fiabilidade e de reatividade.

O Capítulo 4 apresenta o trabalho desenvolvido na automatização do tratamento e validação dos dados, na respetiva alarmística e na reformulação do dashboard de monitorização de consumos de utilidades em Power BI. São descritas as limitações do sistema existente, os objetivos da reformulação e a solução implementada, que inclui a criação de indicadores chave de desempenho (KPIs), filtros dinâmicos por linha de produção e comparação de consumos face ao ano anterior.

Por fim, no Capítulo 5 são sintetizadas as conclusões gerais extraídas ao longo do projeto, validando o cumprimento dos objetivos de fiabilidade e reatividade propostos. Adicionalmente, identificam-se as linhas de continuidade e as perspetivas de desenvolvimento futuro com vista à expansão analítica do sistema implementado.

Capítulo 2

Fundamentação Teórica

Neste capítulo é apresentado o enquadramento teórico relevante para o desenvolvimento do presente trabalho.

2.1 Contextualização Histórica da Cerveja

A evolução histórica da cerveja acompanha o desenvolvimento tecnológico da civilização humana, transitando de um processo artesanal e biológico para uma atividade industrial altamente controlada. Embora as suas origens remontem a aproximadamente 6000 a.C. na Mesopotâmia e no Antigo Egito, onde assumia uma relevância sobretudo social, a estabilização técnica da bebida ocorreu de forma progressiva ao longo dos séculos. Durante a Idade Média, o processo foi aperfeiçoado nos mosteiros com a introdução do lúpulo e, em 1516, a implementação da Lei da Pureza alemã (*Reinheitsgebot*) padronizou os ingredientes essenciais: água, cevada, malte e lúpulo [2].

A verdadeira transformação para a escala industrial e mecanizada ocorreu com a Revolução Industrial, impulsionada por três avanços científicos determinantes no século XIX. Primeiro, os estudos de Louis Pasteur sobre a fermentação permitiram o desenvolvimento da pasteurização, técnica que conferiu estabilidade microbiológica e durabilidade à bebida, viabilizando o seu armazenamento em larga escala. Em segundo lugar, Emil Christian Hansen, no laboratório da Carlsberg, conseguiu isolar e cultivar linhagens puras de leveduras, garantindo a consistência e a repetibilidade do processo fermentativo. Por fim, o aparecimento dos sistemas mecânicos de refrigeração industrial permitiu o controlo térmico contínuo dos tanques ao longo de todo o ano. Estes marcos tecnológicos estabeleceram as bases para a operação das modernas unidades industriais de bebidas, onde o controlo rigoroso de processos e o consumo eficiente de utilidades são vetores críticos de competitividade e sustentabilidade.

2.2 Cadeia de Produção da Cerveja

A produção de cerveja, ilustrada na Figura 2.1, assenta num processo de natureza biotecnológica, podendo ser, de forma sucinta, dividida nas seguintes etapas: maltagem, produção e processamento do mosto, fermentação, maturação e acabamento da cerveja e, por fim, enchimento. No centro de produção de Leça do Balio, a única fase que não é realizada é a maltagem [2].

No contexto da empresa em análise, o processo produtivo organiza-se em três grandes áreas operacionais: Fabrico, Adega e Enchimento. Na unidade de Fabrico ocorre a preparação do mosto, compreendendo operações como a moagem, brassagem, filtração, ebulição e clarificação. A fase seguinte decorre na Adega, onde se realizam o arrefecimento e a fermentação do mosto, bem como a posterior filtração da cerveja. Finalmente, o produto acabado é encaminhado para a secção de Enchimento, onde é acondicionado para distribuição.

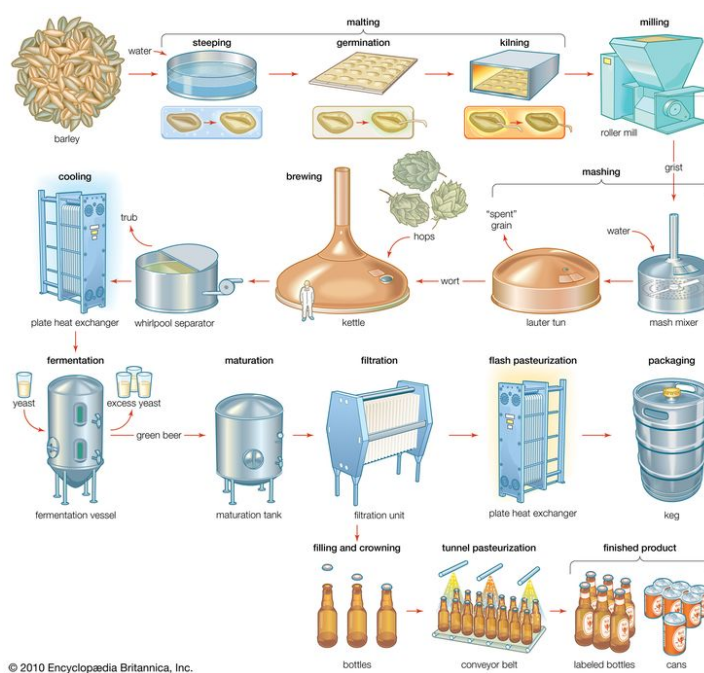


Figura 2.1: Linha de Produção da Cerveja

2.2.1 Maltagem

O malte e os cereais não maltados são fornecidos à unidade industrial e armazenados, sendo a etapa de maltagem realizada externamente. Este processo, com duração aproximada de 6 a 7 dias, compreende três fases: molha, germinação e secagem.

Na molha, o teor de humidade do grão é elevado até cerca de 42%. Segue-se a germinação, fase em que ocorre a síntese de enzimas hidrolíticas, nomeadamente α -amilase e β -amilase,

essenciais para a degradação do amido. Por fim, a secagem, realizada a temperaturas superiores a 80 °C, interrompe a germinação e estabiliza as características do malte.

A principal finalidade da maltagem é, assim, promover o desenvolvimento enzimático necessário para que, na etapa de brassagem, o amido seja convertido em açúcares fermentescíveis, como glucose, maltose e maltotriose.

2.2.2 Fabrico

Após a clarificação, o mosto mantém-se a uma temperatura elevada, sendo necessário proceder ao seu arrefecimento antes da inoculação da levedura. Este arrefecimento é realizado num permutador de placas, recorrendo a água glicolada, de modo a atingir a temperatura adequada à fermentação. Simultaneamente, promove-se o arejamento do mosto com ar estéril, assegurando as condições ideais para o desenvolvimento inicial da levedura. A inoculação ocorre durante o enchimento do fermentador.

A fermentação decorre em reatores cilíndricos, onde os açúcares fermentescíveis são convertidos em etanol e dióxido de carbono, em ambiente anaeróbio e sob temperatura controlada. Esta etapa tem uma duração aproximada de uma semana, finda a qual a levedura sedimentada no fundo do fermentador é recolhida, podendo ser reutilizada em ciclos posteriores.

Concluída a fermentação, segue-se a maturação, fase caracterizada pela redução gradual da temperatura. Este procedimento favorece a estabilização do perfil sensorial da cerveja, promovendo a remoção de compostos indesejáveis, como o diacetilo, subproduto da fermentação. Posteriormente, realiza-se a estabilização coloidal, através do arrefecimento a temperaturas próximas de 0–2 °C, com o objetivo de precipitar e remover frações proteicas.

Por fim, procede-se à filtração, etapa destinada à eliminação de partículas em suspensão, garantindo a limpidez do produto final. Nesta fase podem ainda ocorrer ajustes como a diluição, quando necessária, e a carbonatação, assegurando as características finais da cerveja antes do enchimento [3].

2.2.3 Enchimento

A etapa final do processo produtivo corresponde ao enchimento, fase em que a cerveja é acondicionada em diferentes formatos, nomeadamente garrafas, latas e barris. Antes ou após esta operação, torna-se necessária a estabilização biológica do produto, com o objetivo de assegurar a sua segurança microbiológica e prolongar o prazo de validade. Esta estabilização pode ser realizada por via térmica, através da pasteurização, ou por via física, mediante filtração esterilizante.

A pasteurização pode ocorrer antes do enchimento, designada pasteurização flash, ou após o acondicionamento, através da pasteurização em túnel. A pasteurização flash é habitualmente aplicada no enchimento de barris, uma vez que estes são previamente higienizados

e selados antes do seu preenchimento. No caso das garrafas e latas, recorre-se frequentemente à pasteurização em túnel, garantindo simultaneamente a estabilização do produto e a esterilização da embalagem. Esta técnica caracteriza-se, contudo, por um consumo significativo de água e energia.

Como alternativa, a filtração esterilizante apresenta-se como uma solução tecnológica com potencial de substituição da pasteurização convencional. Este método baseia-se na utilização de membranas de elevada eficiência, com porosidades muito reduzidas, capazes de reter microrganismos. No entanto, trata-se de um processo exigente do ponto de vista técnico e económico, uma vez que requer baixos caudais de operação e condições de enchimento asséticas rigorosas [3].

Concluída a etapa de estabilização biológica, seguem-se as operações finais de rotulagem e embalamento, preparando o produto para distribuição.

2.3 Utilidades Industriais

A produção industrial de cerveja envolve um conjunto de utilidades essenciais que suportam o funcionamento dos diversos processos acima descritos. Estas utilidades garantem as condições necessárias para a operação dos equipamentos, para o controlo dos parâmetros de processo e para a manutenção da qualidade do produto final. Entre as utilidades destacam-se a água, o dióxido de carbono (CO_2), a energia térmica, a eletricidade e o ar comprimido.

A gestão eficiente destas utilidades assume um papel fundamental na indústria cervejeira, uma vez que representam uma parcela significativa dos custos operacionais e do consumo energético global da instalação. Para além disso, a monitorização adequada destes recursos permite identificar oportunidades de melhoria ao nível da eficiência energética, da sustentabilidade e da redução de desperdícios.

Nos pontos seguintes apresenta-se uma breve descrição das principais utilidades utilizadas no processo produtivo.

2.3.1 Água

A água constitui a principal matéria-prima na produção de cerveja, representando uma parte significativa da composição final do produto. Para além da sua utilização direta no processo produtivo, a água é também utilizada em diversas operações auxiliares, como limpeza, arrefecimento e geração de vapor.

No processo cervejeiro, a água desempenha um papel fundamental nas etapas de brasagem, onde é utilizada para extrair os açúcares do malte, bem como em operações de lavagem de grãos e preparação do mosto. Para além disso, grandes volumes de água são utilizados em sistemas de limpeza CIP (Cleaning in Place), garantindo a higiene e a segurança alimentar dos equipamentos de produção.

A gestão eficiente do consumo de água é um aspeto crítico na indústria cervejeira, sendo

comum a implementação de estratégias de monitorização e reutilização de água em diferentes etapas do processo.

2.3.2 Dióxido de Carbono (CO_2)

O dióxido de carbono é um elemento essencial na produção de cerveja, estando diretamente relacionado com o processo de fermentação alcoólica. Durante esta etapa, as leveduras convertem os açúcares presentes no mosto em álcool e dióxido de carbono, sendo este último frequentemente recuperado e reutilizado em diferentes fases do processo produtivo.

O CO_2 é utilizado principalmente na carbonatação da cerveja, conferindo-lhe as características de efervescência e textura típicas da bebida. Para além disso, é utilizado em operações de pressurização de tanques, purga de oxigénio e transferência de produto, contribuindo para evitar a oxidação da cerveja e preservar a sua qualidade.

Adicionalmente, o CO_2 recuperado é armazenado em reservatórios apropriados, permitindo a sua posterior utilização em diferentes aplicações ao longo da cadeia de distribuição. Uma dessas aplicações consiste no enchimento de barris de cerveja, que posteriormente são utilizados em sistemas de extração presentes em máquinas de servir cerveja, onde o dióxido de carbono é utilizado para pressurizar o barril e impulsionar a cerveja até à torneira de serviço.

A recuperação e reutilização do CO_2 gerado na fermentação constitui uma prática comum na indústria cervejeira, permitindo reduzir custos operacionais e minimizar o impacto ambiental associado à produção.

2.3.3 Energia Térmica

A energia térmica constitui uma das utilidades mais relevantes na indústria cervejeira, estando presente em diversas etapas do processo produtivo. As principais formas de energia térmica utilizadas são o vapor, a água quente e os sistemas de recuperação de calor.

O vapor é amplamente utilizado em operações de aquecimento, pasteurização, limpeza e esterilização de equipamentos. A sua produção é normalmente assegurada por caldeiras industriais, que podem ser alimentadas a gás natural, biogás ou outros combustíveis. Na indústria cervejeira, o biogás produzido nas estações de tratamento de águas residuais (ETAR) pode ser aproveitado para alimentar parcialmente as caldeiras, contribuindo para a redução das emissões e dos custos energéticos.

A água quente é igualmente utilizada em processos de aquecimento e em sistemas de recuperação de calor, nomeadamente nos permutadores de calor presentes nas linhas de produção. A recuperação de energia térmica entre correntes quentes e frias permite reduzir significativamente o consumo de combustível e melhorar a eficiência energética global da instalação.

2.3.4 Eletricidade

A energia elétrica constitui uma utilidade transversal a todo o processo produtivo, sendo responsável pela alimentação de uma vasta gama de equipamentos industriais. Entre os principais consumidores de energia elétrica encontram-se bombas, compressores, sistemas de refrigeração, motores elétricos, linhas de enchimento e sistemas de controlo e automação.

Na indústria cervejeira, uma parcela significativa do consumo elétrico está associada aos sistemas de refrigeração, necessários para o controlo das temperaturas durante as etapas de fermentação e maturação da cerveja. Estes sistemas garantem as condições térmicas adequadas ao desenvolvimento das leveduras e à estabilização do produto final.

2.3.5 Ar Comprimido

O ar comprimido constitui uma utilidade amplamente utilizada em ambientes industriais, sendo frequentemente considerado um dos recursos energéticos mais versáteis e simultaneamente mais dispendiosos em termos de produção. Na indústria cervejeira, o ar comprimido é utilizado principalmente na operação de equipamentos pneumáticos, válvulas de controlo, sistemas de transporte e equipamentos de automação presentes nas linhas de produção e enchimento.

A produção de ar comprimido é normalmente assegurada por compressores industriais que fornecem ar a uma determinada pressão para uma rede de distribuição que alimenta os diversos pontos de consumo da instalação. No entanto, os sistemas de ar comprimido são frequentemente caracterizados por perdas significativas de eficiência, resultantes de fugas na rede, pressões de operação inadequadas ou utilização ineficiente do recurso [4].

Devido ao elevado consumo energético associado à sua produção e à dificuldade em identificar perdas de forma imediata, a monitorização adequada do sistema de ar comprimido é também importante na gestão energética das instalações industriais [4].

Neste contexto, o presente trabalho incide na análise do consumo das várias utilidades industriais da unidade de produção de cerveja, com o objetivo de estudar os padrões de consumo, identificar possíveis ineficiências e propor melhorias no processo de monitorização e análise dos dados associados a estas utilidades.

2.4 Sustentabilidade e Monitorização das Utilidades

A gestão eficiente de utilidades industriais (água, eletricidade, energia térmica, ar comprimido e dióxido de carbono) constitui atualmente um vetor estratégico para a sustentabilidade e a competitividade das operações industriais. Estes recursos representam, simultaneamente, insumos indispensáveis ao processo produtivo e uma parcela significativa dos custos operacionais e do impacto ambiental das empresas [5].

No setor cervejeiro, a intensidade de utilização destes recursos assume uma dimensão

particularmente relevante. De acordo com dados publicados para o setor cervejeiro europeu [6, 7], o consumo médio de água na indústria cervejeira europeia situa-se entre 3,5 e 5,0 hectolitros por hectolitro de cerveja produzido (hl/hl), com os melhores operadores a atingir valores próximos de 2,5 hl/hl. Para além da água incorporada no produto, grandes volumes são consumidos em operações de limpeza CIP (Clean-in-Place), arrefecimento e geração de vapor. No que respeita à energia, os sistemas de refrigeração, as caldeiras e as linhas de enchimento são os principais pontos de consumo de energia térmica e elétrica [8]. A produção de cerveja envolve ainda emissões significativas de dióxido de carbono: tanto capturado como subproduto da fermentação e reutilizado no processo, como emitido pela combustão de combustíveis nas caldeiras. A eficiência no aproveitamento do CO₂ de fermentação, bem como a monitorização das emissões diretas e indiretas (Scope 1 e Scope 2), são dimensões crescentemente exigidas pelos referenciais de sustentabilidade corporativa [9].

Em resposta às crescentes exigências regulatórias e aos compromissos de responsabilidade ambiental, as principais organizações cervejeiras mundiais têm vindo a estabelecer metas quantitativas de redução de consumo [6]. O Super Bock Group, empresa no âmbito da qual se insere o presente trabalho, integra estes princípios na sua estratégia de sustentabilidade, com compromissos públicos de redução do consumo de água e energia por hectolitro produzido. A implementação de práticas como a recuperação de calor nos permutadores, a reutilização de água de processo e o aproveitamento do biogás gerado nas estações de tratamento de águas residuais (ETAR) são exemplos concretos desta orientação estratégica.

Neste enquadramento, a monitorização sistemática e rigorosa das utilidades constitui uma ferramenta essencial para apoiar uma gestão mais eficiente dos recursos, identificar oportunidades de melhoria e demonstrar progresso face aos objetivos de sustentabilidade definidos. A norma ISO 50001:2018 (Energy Management Systems) fornece um referencial estruturado para a implementação de sistemas de gestão de energia, estabelecendo requisitos para a monitorização contínua do desempenho energético, a definição de indicadores de desempenho energético (EnPIs) e a identificação de usos significativos de energia (SEUs) [10]. A adoção deste referencial potencia a identificação proativa de ineficiências e suporta a tomada de decisão baseada em dados, premissas que norteiam igualmente o presente trabalho.

2.5 Digitalização e Indústria 4.0 na Gestão de Energia

A transição para o paradigma da Indústria 4.0 transformou profundamente a forma como os recursos energéticos são geridos nas unidades de produção industrial [11]. Historicamente, a monitorização de utilidades dependia de leituras manuais periódicas ou de análises retroativas com alcance limitado, métodos que restringiam significativamente a capacidade de resposta face a anomalias e ineficiências.

No contexto atual, a convergência entre tecnologias de informação (IT) e tecnologias operacionais (OT) é suportada por um ecossistema de ferramentas complementares que atuam em diferentes camadas da pirâmide de automação industrial. No nível de supervisão e controlo, os sistemas SCADA (Supervisory Control and Data Acquisition) permitem a recolha em tempo real de dados de processo a partir de sensores e autómatos (PLCs), centralizando a informação operacional e habilitando a deteção imediata de desvios [12]. A um nível superior, os sistemas MES (Manufacturing Execution Systems) integram dados de produção, qualidade e recursos, fornecendo uma visão transversal do desempenho operacional [13]. No topo desta hierarquia, as plataformas de Business Intelligence (BI) permitem transformar grandes volumes de dados operacionais em informação estruturada, visualizações interativas e indicadores de desempenho (KPIs), acessíveis a diferentes níveis hierárquicos da organização [14].

No âmbito deste trabalho, a plataforma Microsoft Power BI foi adotada como ferramenta central de visualização e análise. A escolha fundamenta-se na sua ampla adoção em contexto industrial e corporativo, na capacidade de integração nativa com ficheiros Microsoft Excel (formato em que os dados de consumo são atualmente armazenados), na flexibilidade para construção de medidas calculadas através da linguagem DAX (Data Analysis Expressions), e na possibilidade de partilha e atualização dinâmica dos relatórios em ambiente colaborativo [15].

O conceito de dashboard operacional é central neste trabalho. Um dashboard operacional é uma interface visual que apresenta, de forma consolidada e em tempo próximo do real, os indicadores mais relevantes para o acompanhamento de um processo, permitindo a identificação imediata de desvios e o suporte à decisão. Na conceção de dashboards eficazes, a literatura destaca a importância da redução da carga cognitiva do utilizador, da codificação visual consistente (cor, forma, posição) e da adaptação do nível de detalhe ao perfil e objetivo de cada utilizador.

Para a gestão energética corporativa alinhada com a norma ISO 50001:2018, a disponibilização de indicadores de desempenho energético (EnPIs) em dashboards interativos representa uma forma concreta de operacionalizar os requisitos de monitorização e análise contínua do desempenho energético [10].

2.6 Monitorização de Utilidades com BI

A monitorização e gestão eficiente de utilidades industriais têm sido objeto de crescente atenção na literatura académica e técnica, em particular no contexto da transição para a Indústria 4.0 e da crescente pressão para a redução do consumo energético [16]. Nesta secção, apresenta-se uma síntese dos principais trabalhos relevantes para o âmbito do presente estudo, organizados em torno da utilização de plataformas de Business Intelligence para a gestão de dados de consumo.

Monitorização e Visualização de Consumos com Business Intelligence

A integração de dados de consumo de utilidades em plataformas de BI tem sido documentada como um vetor crítico para a eficiência operacional em contextos industriais. Estudos recentes demonstraram que a centralização de dados de energia em dashboards interativos reduz o tempo de identificação de anomalias em comparação com abordagens baseadas em relatórios periódicos manuais, e que a definição de indicadores normalizados por unidade produzida (specific energy consumption) é essencial para isolar o efeito da variação do volume de produção na análise dos consumos [17].

Paralelamente, a literatura recente enfatiza que o desenho de interfaces industriais de BI deve focar-se na redução do esforço de compilação de dados, transformando tabelas estáticas em sistemas de suporte à decisão visual em tempo real [14]. No contexto de utilidades complexas, a normalização de indicadores chave de desempenho energético (EnPIs) correlacionados com o volume físico de produção em hectolitros ou toneladas constitui a base para benchmarking interno fiável [17].

No setor agroalimentar e cervejeiro especificamente, trabalhos focados no setor cervejeiro analisaram os fluxos de energia e água ao longo do processo produtivo, concluindo que a monitorização por área e por utilidade é um pré-requisito para a identificação de oportunidades de melhoria [7]. De forma complementar, outro estudo setorial propôs uma metodologia de monitorização baseada em indicadores de desempenho específicos para a indústria cervejeira, evidenciando que a disponibilização de KPIs em tempo real durante as reuniões operacionais melhora a qualidade das decisões de manutenção e de programação da produção [8].

Adicionalmente, investigações focadas na digitalização fabril demonstram que a centralização de contadores dispersos através de arquiteturas cloud de BI não só mitiga o risco de erro humano de transcrição, como dota a gestão intermédia de uma capacidade reativa imediata perante desvios de consumo latentes [16].

Deteção de Anomalias e Sinalização Visual

Para além das plataformas de BI, a literatura documenta abordagens que combinam a análise de consumos com a sinalização visual automática de desvios, tornando a deteção de anomalias acessível a utilizadores sem formação estatística avançada. A definição de limites de referência e a respetiva sinalização por código de cor constituem, neste âmbito, um mecanismo simples e de interpretação operacional direta [18].

De facto, estudos focados na usabilidade cognitiva de painéis industriais apontam que a utilização de formatação condicional baseada em limites históricos atua como um sistema de alarme visual imediato para equipas de manutenção e operação, servindo de ponte direta para métodos estatísticos mais complexos a implementar em fases posteriores. Técnicas mais avançadas de Controlo Estatístico de Processos, baseadas em cartas de controlo,

representam uma evolução possível destas abordagens e são discutidas como trabalho futuro no presente documento.

Posicionamento do Presente Trabalho

Face à literatura existente, o presente trabalho centra-se na automatização do tratamento e validação dos dados de consumo, na alarmística e na reformulação de um dashboard de monitorização em Power BI, aplicados ao conjunto das utilidades de uma unidade cervejeira real. O contributo distintivo reside, por um lado, na especificidade do contexto (indústria cervejeira, dados diários de contadores físicos, múltiplas utilidades e linhas de produção) e, por outro, no desenvolvimento de uma solução operacionalmente funcional e integrada no fluxo de trabalho existente da organização. A aplicação de Controlo Estatístico de Processos a estes dados constitui uma extensão natural desta solução, deixada como trabalho futuro.

Capítulo 3

Análise e Validação dos Dados de Consumo das Utilidades

3.1 Processo Atual de Recolha de Dados

O registo dos consumos de utilidades assenta em leituras diárias dos contadores, efetuadas pelos operadores afetos à área. Os valores obtidos, leituras acumuladas, são anotados em folha de campo e depois transcritos para ficheiros Excel, onde o consumo diário é calculado pela diferença entre registos consecutivos. Cada ficheiro agrega várias utilidades: água (captada, de processo, desmineralizada, industrial e quente), ar comprimido, dióxido de carbono, biogás, vapor e energia elétrica, com uma folha por utilidade ou subsistema. O fluxo de informação está representado na Figura 3.1.

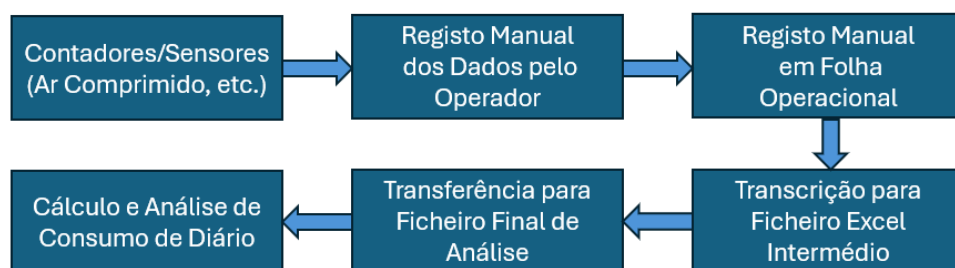


Figura 3.1: Processo de recolha e tratamento de dados de consumo das utilidades.

O processo, embora funcional, encadeia várias etapas manuais, o que abre margem a atrasos na análise e à ocorrência de erros de transcrição [19].

3.1.1 Limitações do Processo Atual

As limitações identificadas agrupam-se em dois planos. Do ponto de vista da fiabilidade, o registo e a transcrição manuais das leituras introduzem um risco permanente de erros que comprometem a integridade dos dados. Do ponto de vista da reatividade, a

ausência de mecanismos automáticos de detecção de anomalias significa que erros de leitura ou consumos anómalos só são detetados após análise manual, introduzindo atrasos que podem ser relevantes em contexto operacional. Estas duas limitações enquadraram o desenvolvimento das soluções descritas nas secções seguintes.

3.2 Definição de Limites de Referência

Para suportar a detecção automática de anomalias, foi realizada uma análise histórica dos consumos registados ao longo de 2025 para as utilidades monitorizadas, como a água, a energia elétrica, o CO₂ e o ar comprimido. A partir das séries de consumo diário, identificaram-se os valores máximos e as maiores variações entre dias consecutivos, por forma a capturar situações de operação extrema.

Com base nestes resultados, definiram-se limites de referência ligeiramente abaixo dos máximos observados. A opção por limites baseados nos máximos históricos prende-se com a sua aderência imediata à realidade da fábrica e com a facilidade de interpretação por parte das equipas no terreno, garantindo uma validação operacional direta e simplificada na fase de arranque do novo sistema de alarmística.

Para as utilidades com sazonalidade comprovada, designadamente a Água Industrial (ETA) e o CO₂ no Coletor Central, foram definidos limites diferenciados para o período de verão (maio a agosto) e para o restante ano. Os limites obtidos estão apresentados nas Figuras 3.2 a 3.16, organizados por utilidade.

Água O consumo de água é monitorizado em vários pontos da instalação, correspondendo a circuitos e utilizações distintas. Para cada coletor determinou-se o valor máximo de consumo diário ou o desvio máximo entre leituras consecutivas, conforme aplicável. Os valores definidos são apresentados nas Figuras 3.2 a 3.7.

	Água Captada (Diferença entre dias)												
	AC 02	AC 07	AC 13	AC 17	AC 19	AC 01	AC 26	AC 27	AC 28	AC 30	AC 31	AC 32	AC 33
Desvio MÁX	20	30	20	20	20	20	20	20	20	20	20	20	20

Figura 3.2: Desvio máximo definido para a Água Captada (diferença entre dias consecutivos).

	Água Processo					Água Rede Geral (ETA)									
	Mini Fábrica	Adagas	Fábrica	Inst. Piloto	Aref. Médio-Fab.	ETAl	Mini Fábrica	Centrais	Inst. Pro.	Recur. CO ₂	Edifícios	Adagas	Enchimento Total	Unid. Processos	CIP Barril/RS
LIMITE	-	-	450	-	-	100	120	150	-	-	90	750	550	110	100
MÁXIMO	-	-	608	-	-	101	141	168	-	-	94	809	676	116	106

Figura 3.3: Limites máximos definidos para a Água Processo e Água Rede Geral (ETA).

	Água Rede Geral (Enchimento)										
	Linha 1	Linha 2	Linha 3	Linha 4 (TR)	Linha 5	Linha 6	Linha 7 (ACE)	Desalcooлизаçã	Aux. Enchim.	Linha Vini	Linha Lata
LIMITE	50	50	350	80	100	55	-	40	150	30	30
MÁXIMO	50	48	471	163	250	101	-	64	231	33	30

Figura 3.4: Limites máximos definidos para a Água Rede Geral no Coletor de Enchimento.

	Água Desmineralizada						Água Aprov. Térmicos						
	Desp.pecif.	Caldeira 4	Caldeira 5	Caldeira 6	Linha Lata	Caldeira Recup.	Água Desalco	RA.D Torre Circ.	A.P. CB05→CB1	Barril→Adegas	Barril	A.P. CB15→ETA	CIP Barril /B0
LIMITE	-	-	-	-	15	-	-	-	-	400	170	-	-
MÁXIMO	-	-	-	-	21	-	-	-	-	458	234	-	-

Figura 3.5: Limites máximos definidos para a Água Desmineralizada e Água para Aproveitamentos Térmicos.

	Água Industrial (ETA)								
	Enchim. Total (F1)	Armazém 2 (F2)	Regas	Centrais	Inst. Frio	Recup. CO2	VINI	Sec. Levedura	Adegas
LIMITE	1100	200	-	P.T	P.T	P.T	15	25	80
MÁXIMO	1196	229	-	P.T	P.T	P.T	15	26	81
Período Verão				650	500	60			
Período Inverno				450	290	55			
Máximo Verão				640	485	63			
Máximo Resto do Ano				469	400	58			

Figura 3.6: Limites máximos definidos para a Água Industrial no Coletor ETA, com distinção entre período de verão e restante ano.

	Água Industrial (Enchimento)															
	Linha 1	Linha 2	Linha 3	Linha 5	Linha 6	Subst. Caldeira	Panela (Plant 5/10)	Auxiliar Enchimento	Torre de Refrigeração	Água Grátis L2	Plant. (3)	Soda	Barris	Linha Vini	Linha Lata	CIP Barril/B0
LIMITE	150	150	250	40	25	-	150	250	25	140	375	-	175	75	250	45
MÁXIMO	310	237	667	51	25	-	172	326	29	145	389	-	311	79	321	46

Figura 3.7: Limites máximos definidos para a Água Industrial no Coletor de Enchimento.

Ar Comprimido O ar comprimido é distribuído por três coletores principais (Centrais, Adegas e Enchimento), tendo sido determinado para cada um o limite máximo de consumo diário com base nos valores históricos. As Figuras 3.8 a 3.10 apresentam os limites de cada coletor; a Figura 3.11 reúne ainda o limite do Armazém Automático e o do CO₂ no Coletor Central, este último com distinção entre verão e restante ano.

	Ar Comprimido (Nm3) - Consumo Coletor Centrais										
	Fabrico	Adegas	Enchimentos	Central Águas	Barris	Mini Fábrica	Recup. CO2	Inst. Piloto	Central Térmic	Húmido Adegas	Fábrica 2
DEFINIDO	10000	10000	110000	2500	18500	2000	850	300	750	250000	3500
MÁXIMO	10480	10115	112540	3065	18837	2205	898	339	1232	294003	3578

Figura 3.8: Limites máximos definidos para o Ar Comprimido no Coletor de Centrais.

	Ar Comprimido (Nm3) - Consumo Coletor Adegas						
	TCF's	CC 1000 HL	CC 3000 HL	Prop. Levedura	Filtração Schen	Filtração L2	
DEFINIDO	-	-	500	11500	-	350	
MÁXIMO	-	-	522	11821	-	369	

Figura 3.9: Limites máximos definidos para o Ar Comprimido no Coletor de Adegas.

	Ar Comprimido (Nm3) - Consumo Coletor Enchimento									
	Linha 1	Linha 2	Linha 3	Linha 5	Linha 6	Aux. Enchim.	Linha Retorno	Desalcoolizaçã	Vini	Linha Lata
DEFINIDO	12500	20000	12500	17500	7500	900	550	1200	3000	10000
MÁXIMO	20057	24497	14379	34442	8350	955	557	1415	3215	13915

Figura 3.10: Limites máximos definidos para o Ar Comprimido no Coletor de Enchimento.

	Ar Comprimido (Nm3) - Consumo Coletor Armazém Automático				Consumo CO ₂ - Coletor Central		
	High Bay	Ofic. Carr. Bat.F2	Picking	Expedições	Adega	Enchimentos	Mini-Fábrica
DEFINIDO	200	75	150	15	P.T	10	AVARIA
MÁXIMO	259	92	168	18		24	
Período Verão					20		
Período Inverno					15		
Máximo Verão					32		
Máximo Resto Ano					21		

Figura 3.11: Limites máximos definidos para o Ar Comprimido no Armazém Automático e para o CO₂ no Coletor Central, com distinção entre período de verão e restante ano.

Dióxido de Carbono (CO₂) O CO₂ é uma utilidade crítica na indústria cervejeira (presente tanto na fermentação como no enchimento de barris), monitorizado em dois coletores: Adegas e Enchimento. Os limites correspondentes constam das Figuras 3.12 e 3.13.

	Consumo CO ₂ - Coletor Adegas						
	CC1000+Tq.Ág.D.	CC3000+Tq.Tamp.	Filtração L1+Meg.	TCF's	Carbonatação L1	Filtração L2	Anel CC3000
DEFINIDO	2000	400	7000	5000	2500	3500	1750
MÁXIMO	5316	913	8861	6444	5224	5855	3805

Figura 3.12: Limites máximos definidos para o CO₂ no Coletor de Adegas.

	Consumo CO ₂ - Coletor Enchimento							
	Linha 1	Linha 2	Linha 3	Linha 4	Linha 5	Linha 6	Linha 7	Linha Lata
DEFINIDO	2000	1750	2500	5000	2750	2500	2500	7500
MÁXIMO	2562	2314	3087	5318	2827	3318	3131	9788

Figura 3.13: Limites máximos definidos para o CO₂ no Coletor de Enchimento.

Energia Térmica Para o biogás, vapor e água quente, os limites foram expressos em termos de desvio máximo entre leituras consecutivas, critério mais representativo do comportamento destas utilidades do que o consumo diário absoluto. As Figuras 3.14 a 3.16 apresentam os desvios máximos e limites definidos.

	Biogás (Nm3)				Vapor Fábrica (ton)			Vapor Adegas (ton)		Vapor Enchimento (ton)						
	Produção	Flare (virtual)	Caldeira 4	Caldeira 5	Sala Nord	Permut. E500	Permut. E600	Adegas	Desa/correlação	Total Enchm.	Barril L1 e L2	Linha 5	JP Enchimento	Mini Fábrica	Inst. Piloto	Vini
DESVIO	-	1500	-	-	160	-	-	40	25	250	125	10	5	10	2	7750
MÁXIMO	-	5023	-	-	175	-	-	47	28	341	134	11	7	11	1	7931

Figura 3.14: Desvios máximos definidos para o Biogás, Vapor Fábrica, Vapor Adegas e Vapor Enchimento.

	Vapor Centrais (ton)				Vapor ETAR/Restaurante (ton)			Aprov. Térmicos (kWh)			
	Central CO ₂	Central Águas	Desgaseific. ICT	Secagem de Levedura	ETAR + Restaur.	ETAR	Restaurante	Barris	Barris + Adegas	Desgaseificador	Adegas (MAH)
DESVIO	15	3	40	60	35	35	5	12500	20000	7500	15
MÁXIMO	22	50	41	62	62	60	7	14413	40591	11248	29

Figura 3.15: Desvios máximos e limites definidos para o Vapor Centrais, Vapor ETAR/Restaurante e Aproveitamentos Térmicos.

	Água Quente Linhas Total (MWh)							Água Quente Linhas c/ Cogeração (MWh)					
	Total Linhas	Linha 1	Linha 2	Linha 3	Linha 5	Linha 6	Linha Lata	Total Linhas	Linha 1	Linha 2	Linha 3	Linha 5	Linha 6
DESVIO	130	15	25	15	15	12	20	100	20	20	30	10	10
MÁXIMO	131	19	26	30	24	22	20	102	20	20	30	13	11

Figura 3.16: Desvios máximos definidos para a Água Quente por Linhas (total e com cogeração).

3.3 Soluções Implementadas

Com base nos limites estabelecidos, foram desenvolvidas três soluções complementares: mecanismos de verificação automática no ficheiro Excel, automatização da importação de dados e um sistema de alertas por correio eletrónico. As duas últimas passaram por um ciclo de migração para arquitetura cloud, documentado nas subsecções respetivas.

3.3.1 Verificação Automática por Formatação Condicional

Foram implementadas regras de formatação condicional que assinalam automaticamente dois tipos de anomalia. Uma leitura inferior à do dia anterior (fisicamente impossível em contadores acumulados, salvo erro ou falha de equipamento) é destacada a vermelho (Figura 3.17). Um consumo diário acima do limite de referência é destacado a amarelo (Figura 3.18).

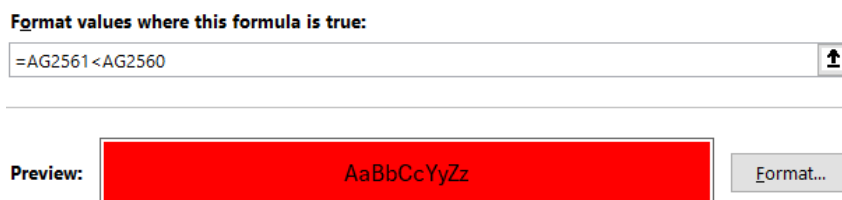


Figura 3.17: Formatação condicional: leitura inferior ao valor do dia anterior.

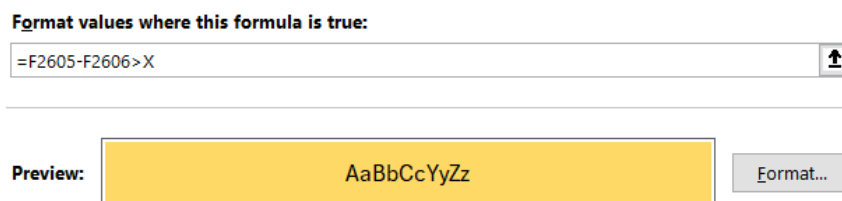


Figura 3.18: Formatação condicional: consumo diário superior ao limite definido.

Para utilidades com sazonalidade, as regras comutam automaticamente entre o limite de verão e o limite de inverno (Figura 3.19), contemplando ainda um limite específico para o período adjacente ao verão (Figura 3.20), evitando falsos positivos nos meses de maior consumo esperado.

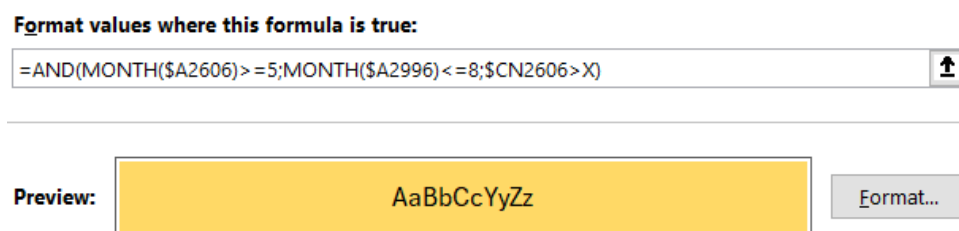


Figura 3.19: Formatação condicional: limite sazonal diferenciado para o período de verão.

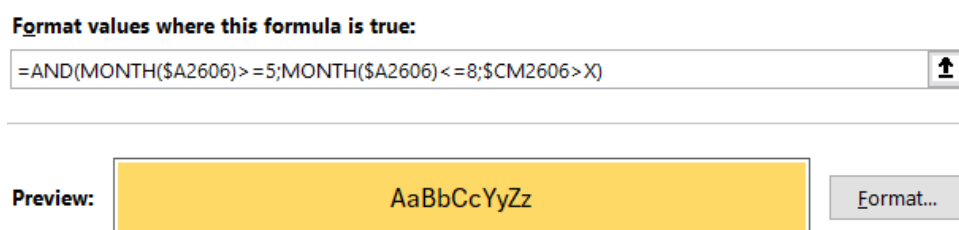


Figura 3.20: Formatação condicional: limite para o período adjacente ao verão.

Uma quarta regra monitoriza o consumo acumulado desde o início do período, complementando a análise diária na deteção de desvios que se acumulam progressivamente sem ultrapassar o limite num único dia (Figura 3.21).

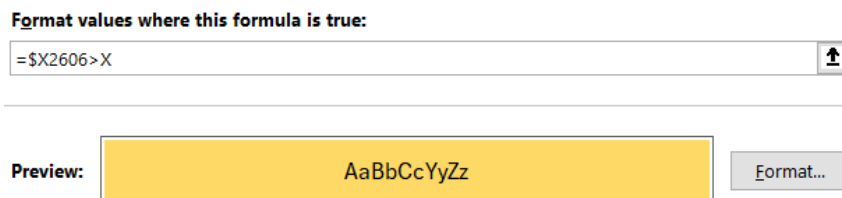


Figura 3.21: Formatação condicional: consumo acumulado superior ao valor de referência.

3.3.2 Automatização da Importação de Dados

A importação das leituras para o ficheiro da empresa era feita manualmente: a colaboradora responsável acedia ao SharePoint da subcontratada, copiava os novos valores e transcrevia-os para o ficheiro local, um processo demorado, repetitivo e vulnerável a erros. A solução desenvolvida para eliminar esta dependência passou por duas fases iterativas.

Fase 1, Macro VBA Foi desenvolvida uma macro VBA (*Visual Basic for Applications*) integrada diretamente nos ficheiros Excel, que permite atualizar os dados de cada utilidade com um simples clique, sem qualquer acesso manual ao SharePoint. O código-base encontra-se no Anexo A.1. A solução é modular: uma macro por ficheiro de utilidade, conforme a Tabela 3.1.

Tabela 3.1: Macros VBA desenvolvidas por ficheiro de utilidade.

Macro	Ficheiro de Origem
AtualizarLeituras	Água.xlsb
AtualizarArComprimidoCO2	Ar Comprimido e CO2.xlsb
AtualizarEnergia	Consumo e Produção Energia.xlsb
AtualizarEletricidade	Eletricidade.xlsb
AtualizarHorarioMaquinas	Horas Funcionamento Máquinas.xlsb

Cada macro executa sequencialmente quatro passos: forçar a sincronização do One-Drive; criar uma cópia temporária do ficheiro de origem para evitar conflitos de acesso; comparar as datas e aplicar a lógica de atualização; eliminar a cópia temporária no final. A lógica de atualização distingue quatro situações, descritas na Tabela 3.2.

Tabela 3.2: Lógica de atualização aplicada pela macro VBA consoante a data da linha.

Situação	Condição	Comportamento
Data nova	Posterior à última data registada	Copia a linha inteira do ficheiro de origem
Últimos 30 dias	Entre hoje e há 30 dias	Substitui todos os valores, incluindo células vazias
Datas anteriores	Antes dos últimos 30 dias	Preenche apenas células vazias no ficheiro destino
Datas futuras	Posteriores à data atual	Ignoradas

A janela de 30 dias garante que correções retroativas da subcontratada sejam refletidas no ficheiro da empresa; fora deste intervalo, preservam-se eventuais ajustes manuais já efetuados. A comparação de datas é feita em formato textual "dd/mm/yyyy" (e não por conversão numérica) por se ter verificado que esta última produzia erros de interpretação dia/mês sempre que ambos os valores são inferiores a 13.

Do ponto de vista técnico, a sincronização com o OneDrive é gerida por tentativas repetidas (intervalo de cinco segundos, máximo de dois minutos) caso o ficheiro esteja bloqueado. A pasta temporária é determinada via `Environ("TEMP")`, tornando a macro portátil entre computadores. O ficheiro de origem é sempre aberto em modo `ReadOnly:=True`, garantindo que os dados originais da subcontratada não são alterados.

A macro demonstrou o comportamento esperado em contexto real. Ainda assim, mantinha uma dependência estrutural: exigia um computador local com Excel ativo no momento da atualização, o que condicionava o processo à disponibilidade de hardware e à presença de um utilizador. Num contexto em que a automatização plena era um objetivo estratégico, esta limitação justificou a migração para uma arquitetura inteiramente baseada na cloud.

Fase 2, Power Automate e Office Scripts (solução final) Validada a lógica da macro VBA em contexto real, a solução foi reimplementada como um fluxo no Microsoft Power Automate, esquematizado na Figura 3.22, executado inteiramente na cloud e sem qualquer dependência de hardware local ou da presença de um utilizador. Nesta arquitetura, os ficheiros da subcontratada e os ficheiros da empresa residem no mesmo SharePoint em pastas distintas, eliminando a necessidade de sincronização via cliente OneDrive. Um trigger de *Recurrence*, configurado para execução diária às 8h00, aciona um Office Script de leitura e um Office Script de escrita, ambos apresentados nas Secções A.2.1 e A.2.2, que percorrem cada par de ficheiros (origem e destino) e aplicam a lógica de atualização descrita na Tabela 3.3.

Tabela 3.3: Lógica de atualização do Office Script (Power Automate): versão final.

Situação	Condição	Comportamento
Data nova	Posterior à última data registada	Copia a linha inteira do ficheiro de origem
Últimos 3 dias	Entre hoje e há 3 dias	Preenche apenas células vazias no ficheiro destino
Datas anteriores	Antes dos últimos 3 dias	Ignoradas
Datas futuras	Posterior à data atual	Ignoradas
Nota: células com valor existente nunca são sobrescritas, independentemente da situação.		

A migração para o Power Automate manteve o comportamento essencial da macro original e, com base na experiência de utilização diária, permitiu afiná-lo. Por um lado, a conversão dos ficheiros de origem de *.xlsb* para *.xlsm* foi necessária, dado que o runtime do Office Scripts não suporta o formato binário; a comparação de datas manteve o formato textual *dd/mm/yyyy* pela mesma razão técnica já identificada na fase anterior. Por outro lado, dois ajustes resultaram diretamente da observação do processo em uso real. Primeiro, a janela de atualização retroativa foi reduzida de 30 para 3 dias: verificou-se que um intervalo tão alargado não se justificava na prática, bastando os últimos dias para preencher as células que ficassem em branco. Segundo, reforçou-se a regra de não-sobrescrita: quando uma célula do destino já contém um valor diferente do da origem, o fluxo preserva o valor existente, garantindo que eventuais correções introduzidas manualmente pela colaboradora nunca são revertidas pela automatização. O fluxo limita-se, assim, a completar as células ainda vazias dos últimos 3 dias, respeitando integralmente o trabalho humano.

A prevenção de execuções duplicadas, antes assegurada por um ficheiro de registo local, passou a ser garantida pelo histórico de execuções nativo do Power Automate.

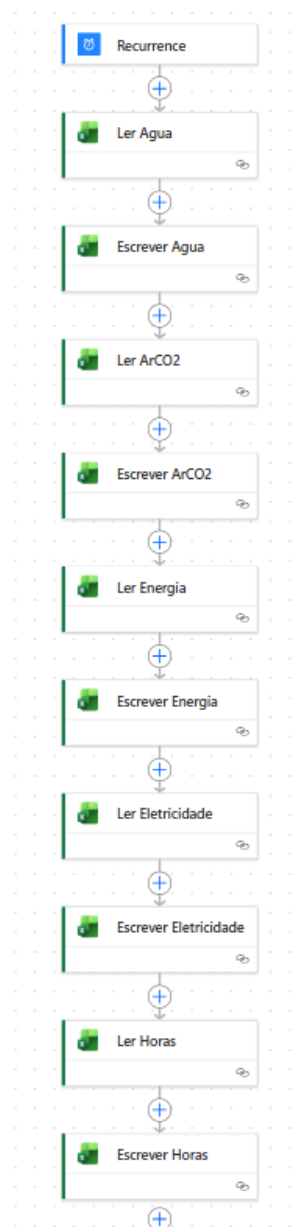


Figura 3.22: Fluxo de atualização diária no Power Automate.

3.3.3 Sistema de Alertas por Correio Eletrónico

Em paralelo com a importação automática de dados, foi desenvolvido um sistema de alarmística proativa que notifica diariamente os responsáveis sobre anomalias detetadas nos consumos, dispensando a verificação manual do ficheiro Excel. O sistema passou igualmente por duas fases de desenvolvimento.

Fase 1, Script Python Foi implementado um script Python que analisa os dados do dia anterior em cada ficheiro de utilidade e envia um relatório HTML por correio eletrónico.

O código encontra-se no Anexo B.1. O formato do relatório gerado está ilustrado na Figura 3.23. Os tipos de alerta emitidos são descritos na Tabela 3.4.

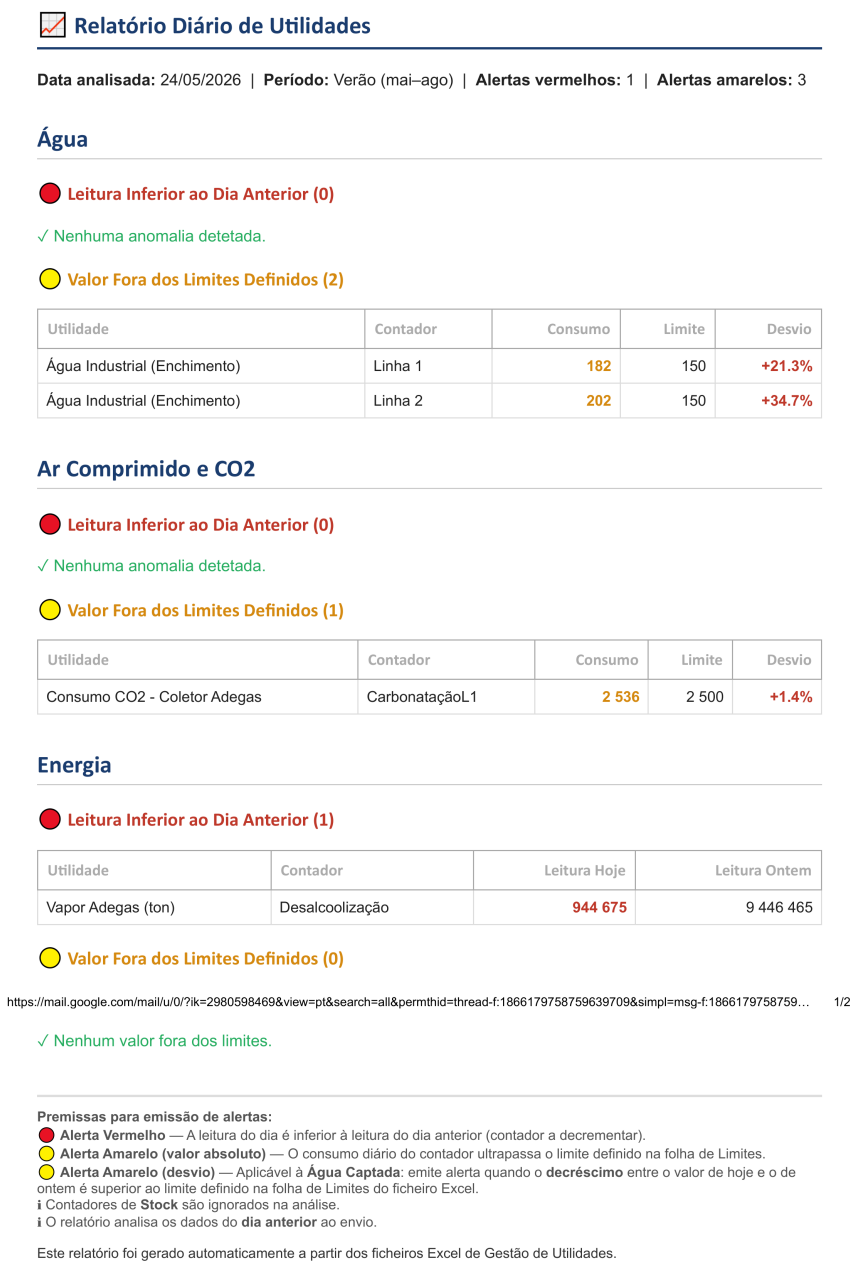
Tabela 3.4: Tipos de alerta emitidos pelo sistema de notificação automática.

Tipo	Condição	Representação
Alerta Vermelho	Leitura inferior à do dia anterior	Tabela com leitura atual e anterior
Alerta Amarelo (absoluto)	Consumo diário superior ao limite da folha Limites	Tabela com consumo, limite e desvio percentual
Alerta Amarelo (desvio)	Água Captada: decréscimo entre dias consecutivos superior ao limite	Tabela com valores consecutivos e diferença

Os limites são lidos diretamente da folha Limites de cada ficheiro Excel, com suporte a valores sazonais: qualquer atualização reflete-se automaticamente no sistema, sem alteração de código. Contadores classificados como *Stock* são excluídos da análise por não representarem consumos diretos.

Do ponto de vista técnico, o script utilizava a biblioteca `pyxlsb` para leitura de ficheiros `.xlsb` e `openpyxl` para ficheiros `.xlsx`. O envio de correio era feito via API COM do Outlook (`pywin32`), que assegura compatibilidade com o ambiente corporativo e suporte a mensagens em HTML. Um ficheiro de registo local impedia envios duplicados no mesmo dia, mesmo em caso de execuções repetidas.

A lógica de deteção e o formato do relatório foram validados localmente. A implementação em produção num servidor corporativo revelou-se, contudo, bloqueada por dois constrangimentos: o processo de aprovação de infraestrutura (incluindo acesso ao servidor, instalação das dependências Python e configuração do agendador de tarefas) era incompatível com o calendário do projeto; e a integração via `pywin32` levantou restrições de segurança no ambiente corporativo relativas ao acesso programático ao Outlook. A solução foi, por isso, migrada para o Microsoft Power Automate com Office Scripts, eliminando ambas as barreiras sem comprometer a lógica de análise desenvolvida.



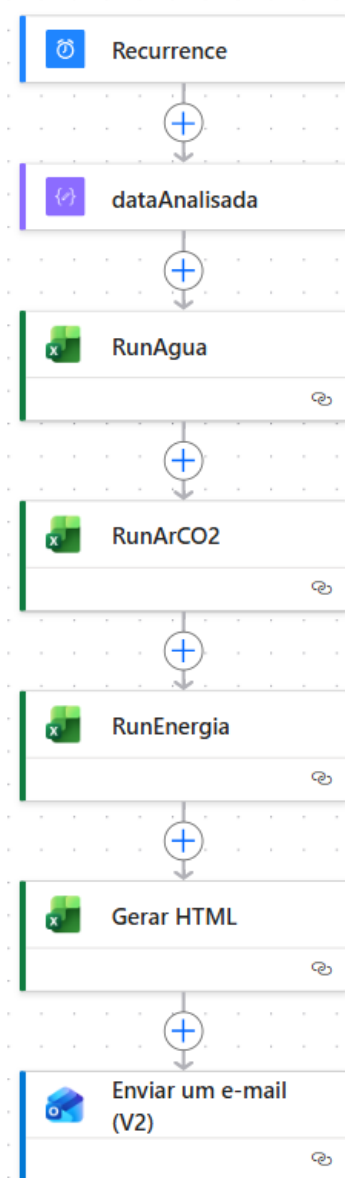


Figura 3.24: Fluxo Power Automate para envio de alertas de utilidades: sete ações sequenciais.

O fluxo inicia com um trigger de *Recurrence* configurado para as 10h00 (hora de Lisboa), hora escolhida para garantir que as leituras do dia anterior já foram registadas pelos operadores. A segunda ação executa um script auxiliar (*dataAnalizada*) que calcula a data a analisar e a disponibiliza como variável partilhada, evitando inconsistências entre os scripts seguintes.

Seguem-se três Office Scripts de análise por utilidade (*RunAgua*, *RunArCO2* e *RunEnergia*), cujo código se encontra no Anexo B.2.1. Cada script é executado sobre o ficheiro correspondente no OneDrive, percorre as leituras do dia analisado, aplica as regras de deteção de anomalias e devolve ao fluxo uma estrutura JSON com os alertas detetados.

Essa estrutura é passada ao script **Gerar HTML** (Anexo B.2.2), que produz o corpo do correio eletrónico com formatação diferenciada por tipo de alerta: vermelho para leituras inversas, amarelo para consumos acima do limite. A separação entre a lógica de análise e a lógica de apresentação permite alterar o formato do relatório sem tocar nos scripts de deteção.

O relatório é enviado pela ação *Send an email (V2)* do conector Outlook 365, operando com as credenciais organizacionais da conta associada ao fluxo, sem necessidade de servidor dedicado ou aprovação de infraestrutura.

Durante a migração foram identificados e resolvidos vários problemas técnicos. As datas nos ficheiros Excel estavam em formato DD/MM/YYYY, enquanto o runtime do Office Scripts as interpreta em ISO, discrepância que causava erros silenciosos do tipo *Invalid Date* e resultava em linhas incorretamente ignoradas; o problema foi resolvido com uma função de parsing explícito. Foi também corrigido um erro de indexação off-by-one na leitura dos limites e uma inconsistência entre os nomes das utilidades nas folhas de dados e na folha de limites, tratada com lógica de pesquisa com fallback. Por fim, verificou-se que expressões dinâmicas do Power Automate inseridas como texto literal, em vez de introduzidas pelo editor de expressões da plataforma, produziam outputs incorretos sem qualquer mensagem de erro, exigindo revisão sistemática de cada ação do fluxo.

3.4 Processo de Validação e Detecção de Anomalias

As soluções descritas na secção anterior integram-se num processo diário de registo e validação das leituras, esquematizado na Figura 3.25.

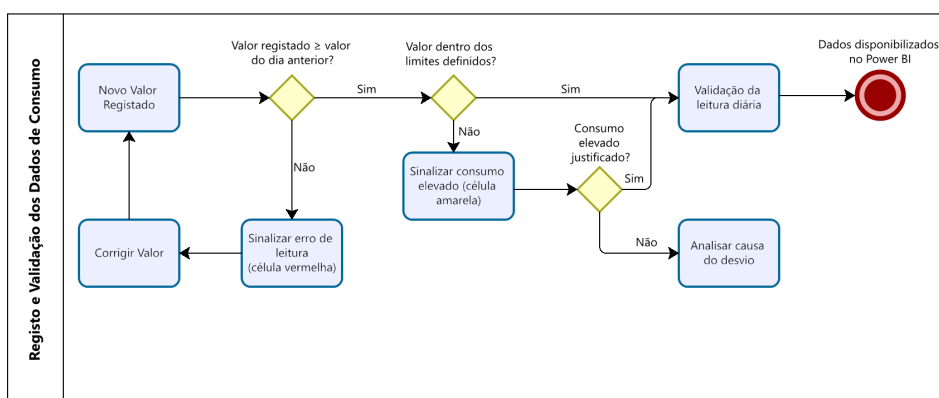


Figura 3.25: Processo de registo e validação dos dados de consumo das utilidades.

O processo inicia-se com o registo dos novos valores pelos operadores. Às 8h00, o fluxo Power Automate executa a sincronização automática das leituras entre os ficheiros Excel da subcontratada e os ficheiros da empresa. Às 8h30, é enviado o relatório diário de alertas por correio eletrónico, com base nos dados já sincronizados.

A partir deste ponto, a validação decorre em dois níveis. Primeiro, verifica-se se os valores registados são superiores ou iguais aos do dia anterior, condição obrigatória em contadores acumulados. Uma leitura inferior sinaliza um erro de registo: a célula fica destacada a vermelho, os valores são corrigidos, e a sincronização é repetida para refletir a correção. Confirmada a consistência, verifica-se se o consumo diário se mantém dentro dos limites definidos.

Um consumo acima do limite, assinalado a amarelo, desencadeia uma análise contextual. Se o desvio tiver justificação operacional (produção elevada, limpeza, manutenção ou condições climáticas), a leitura é validada. Caso contrário, procede-se à investigação da causa: verificação de fugas na rede, inspeção de equipamentos com consumos anómalos ou confirmação de erros no registo.

Concluída a validação, os dados ficam disponíveis para análise no dashboard Power BI cuja reformulação é descrita no Capítulo 4.

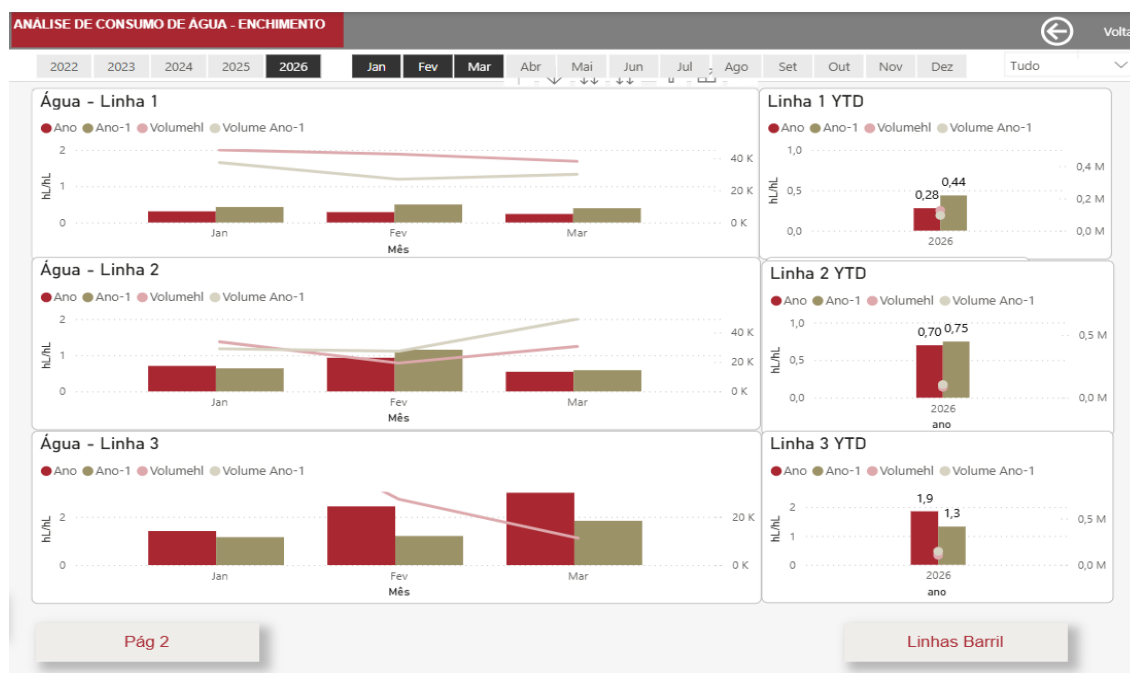
Capítulo 4

Reformulação do Dashboard de Monitorização em Power BI

4.1 Limitações do Dashboard Existente

O dashboard originalmente utilizado para a monitorização dos consumos de utilidades, desenvolvido em Power BI, apresentava limitações significativas ao nível da usabilidade, escalabilidade e capacidade analítica, comprometendo a eficiência do processo de análise. Como ilustrado na Figura 4.1, a estrutura do dashboard assentava numa lógica de organização por utilidade, distribuindo a informação por múltiplas páginas independentes, tipicamente duas ou três por utilidade, separando as linhas produtivas entre si. Considerando o total de utilidades monitorizadas e o número de linhas e áreas produtivas da instalação, este modelo resultava em mais de duas dezenas de páginas a consultar mensalmente, tornando o processo moroso e pouco eficiente do ponto de vista operacional. Esta fragmentação tinha repercussões diretas na preparação da análise mensal de KPIs apresentada em reunião de gestão. Conforme ilustrado na Figura 4.2, a análise era suportada por uma apresentação autónoma, construída manualmente através da recolha e compilação de capturas de ecrã extraídas individualmente de cada página do dashboard. Para além de constituir um processo altamente suscetível a erros de seleção e inconsistências entre os dados apresentados, o tempo despendido nesta tarefa ascendia a aproximadamente um dia de trabalho por mês, de acordo com o diagnóstico e a validação de rotina da colaboradora responsável pelo processo, o que representava um custo operacional significativo e recorrente.

Adicionalmente, ambos os suportes careciam de indicadores de variação percentual e de mecanismos de comparação direta com períodos homólogos (Same Period Last Year, SPLY), limitando a capacidade de interpretação do desempenho e obrigando os utilizadores a realizar análises manuais para identificar tendências ou desvios.



Página 1/2 — Água, Energia Térmica e Eletricidade

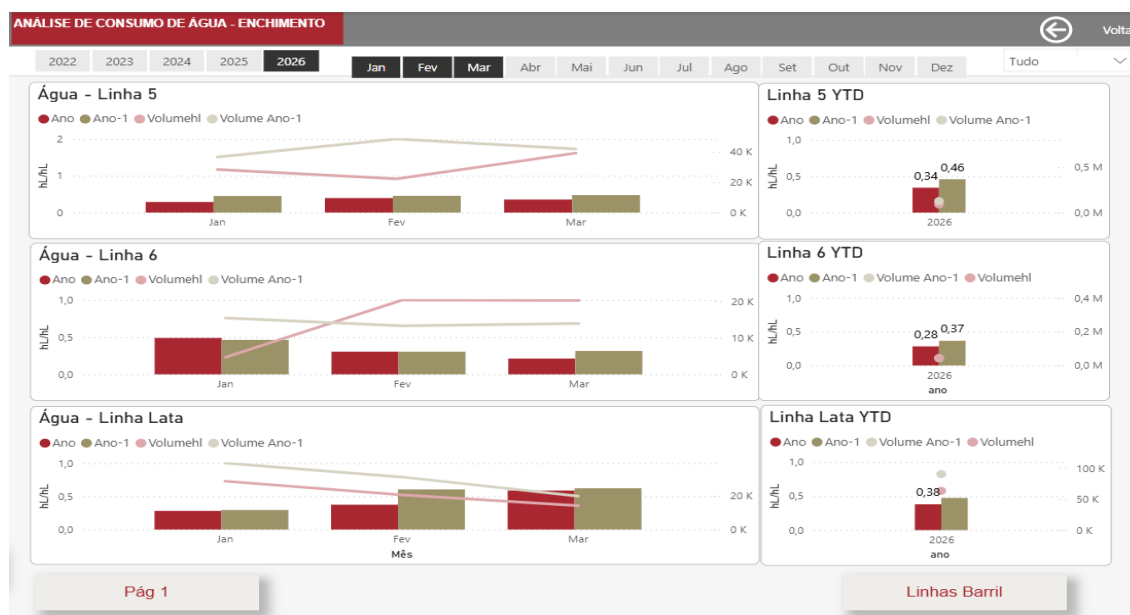
Página 2/2 — Ar Comprimido e CO₂

Figura 4.1: Dashboard original para a Água no Enchimento, distribuído por páginas independentes.



Figura 4.2: Apresentação Mensal apresentada em reunião da empresa para avaliação e discussão de resultados.

4.2 Objetivos da Reformulação

Face às limitações identificadas, a reformulação do dashboard teve como principal objetivo melhorar a eficiência da análise dos consumos e reforçar a capacidade de suporte à decisão.

De forma mais específica, definiram-se os seguintes objetivos:

- Implementar um mecanismo de seleção dinâmica de linha de produção, de modo a que todos os gráficos e indicadores se atualizem automaticamente com base na linha selecionada;
- Disponibilizar indicadores de desempenho (KPIs) que permitam comparar o consumo atual com o do ano anterior, tanto em termos mensais como acumulados (YTD – *Year-to-Date*);
- Incluir métricas de variação percentual, possibilitando identificar de forma imediata aumentos ou reduções de consumo face ao período homólogo;
- Facilitar a análise mensal e semanal, com filtros de Ano e Mês que permitam focar períodos específicos de interesse;
- Melhorar a legibilidade visual, recorrendo a indicadores de cor (verde/vermelho) para sinalizar automaticamente desvios face ao desempenho esperado.

Estes objetivos refletem uma abordagem orientada não apenas à visualização de dados, mas à sua utilização efetiva como ferramenta de apoio à gestão operacional.

4.3 Arquitetura da Solução

O modelo de dados subjacente ao dashboard foi estruturado de forma a suportar a filtragem dinâmica por linha de produção e a comparação temporal. As principais componentes da solução incluem:

- Tabelas de factos com os consumos diários de cada utilidade, organizados por linha, mês e ano;
- Tabelas de dimensão para o tempo (ano, mês, semana) e para a estrutura organizacional (linha, secção, utilidade);
- Medidas DAX (*Data Analysis Expressions*) para o cálculo dos KPIs, incluindo os valores acumulados YTD, os valores do ano anterior e as respetivas variações percentuais;
- Parâmetros de seleção de linha implementados através de segmentadores de dados (*slicers*), garantindo que a interação do utilizador propague automaticamente os filtros a todos os visuais da página.

A Figura 4.3 ilustra a página inicial do dashboard reformulado, evidenciando a navegação simplificada para as duas principais áreas de análise.



Figura 4.3: Página inicial do dashboard reformulado, com acesso direto às Análises Mensais de Enchimento, Adega e Fabrico através de botões de navegação.

4.4 Página de Análise por Linha de Produção

A principal inovação introduzida na reformulação consiste numa página dedicada à análise por linha de produção. Esta página permite ao utilizador selecionar uma linha específica através de botões de seleção, desencadeando a atualização automática de todos os visuais presentes.

Para cada linha selecionada, são disponibilizados cinco grupos de indicadores visuais, correspondentes às principais utilidades monitorizadas: Água, Térmica, Eletricidade, Ar Comprimido e CO₂, cada um com duas variantes – o valor acumulado desde o início do ano (YTD) e o valor do mês em análise.

Cada grupo de indicadores inclui um gráfico de barras combinado com uma linha de tendência, que apresenta o volume de consumo do ano atual e do ano anterior em barras sobrepostas, e o rácio de consumo por unidade produzida como linha. Abaixo de cada gráfico é apresentado um cartão de KPI com o valor atual em destaque, acompanhado do valor do ano anterior e da variação percentual correspondente. O código de cor do cartão (vermelho ou verde) é determinado automaticamente com base na comparação entre o valor atual e o do período homólogo, permitindo uma leitura imediata do desempenho.

Os filtros de Ano e Mês, disponibilizados através de segmentadores de dados, permitem restringir a análise a períodos específicos, facilitando a identificação de anomalias sazonais ou desvios pontuais.

4.5 Reformulação da Análise Mensal de KPIs

4.5.1 Motivação e Impacto Operacional

A preparação da análise mensal de KPIs de consumo constituía, na versão anterior do dashboard, uma das principais limitações operacionais identificadas, conforme descrito na Secção 4.1. A fragmentação da informação por mais de 20 páginas tornava o processo de recolha manual moroso e propenso a inconsistências.

A reformulação desta área teve como objetivo central eliminar esse processo, concentrando

toda a informação necessária num conjunto mínimo de páginas estruturadas de forma lógica, e tornando a análise realizável em tempo real, diretamente no dashboard, durante a própria reunião mensal.

4.5.2 Processo de Desenvolvimento e Versões Intermédias

O processo de desenvolvimento foi iterativo, tendo sido testadas várias abordagens antes de se chegar à configuração final. Numa primeira fase, explorou-se a manutenção de gráficos separados para a análise YTD e para o mês corrente, seguindo a lógica da versão anterior. Embora esta abordagem permitisse uma leitura isolada de cada período, a ausência de integração visual entre os dois horizontes temporais tornava a comparação entre o desempenho mensal e o acumulado pouco intuitiva, obrigando o utilizador a alternar entre gráficos para construir uma leitura global. Foram igualmente testadas outras configurações, nomeadamente cartões de variação percentual posicionados por baixo de cada gráfico KPI e layouts com fundo colorido por tipo de utilidade para facilitar a distinção visual. Explorou-se ainda a representação simultânea do mês corrente e do YTD num único gráfico, com sobreposição de pontos para os volumes de produção. Apesar de cada uma destas abordagens apresentar vantagens parciais, nenhuma reuniu de forma satisfatória os critérios de clareza, compacidade e rapidez de leitura exigidos para um contexto de reunião operacional. A solução final resultou da combinação das melhores características de cada iteração, chegando a uma estrutura que integra num único visual toda a informação relevante para a análise de cada KPI, permitindo uma leitura imediata do sentido da variação face ao ano anterior.

As Figuras 4.4 a 4.8 ilustram os protótipos testados durante o processo iterativo.

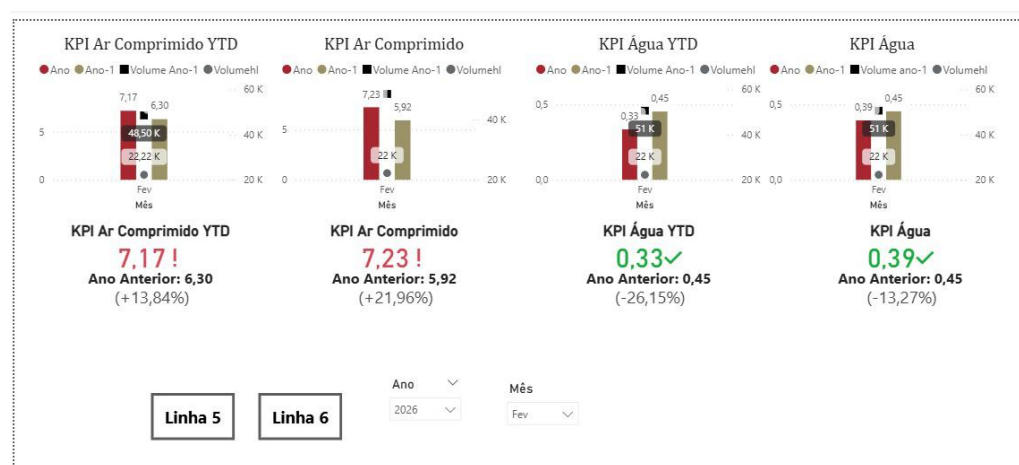


Figura 4.4: Versão intermédia com cartões de detalhe por baixo de cada gráfico, apresentando o valor atual, o valor do ano anterior e a variação percentual: gráficos YTD e Mês ainda separados.

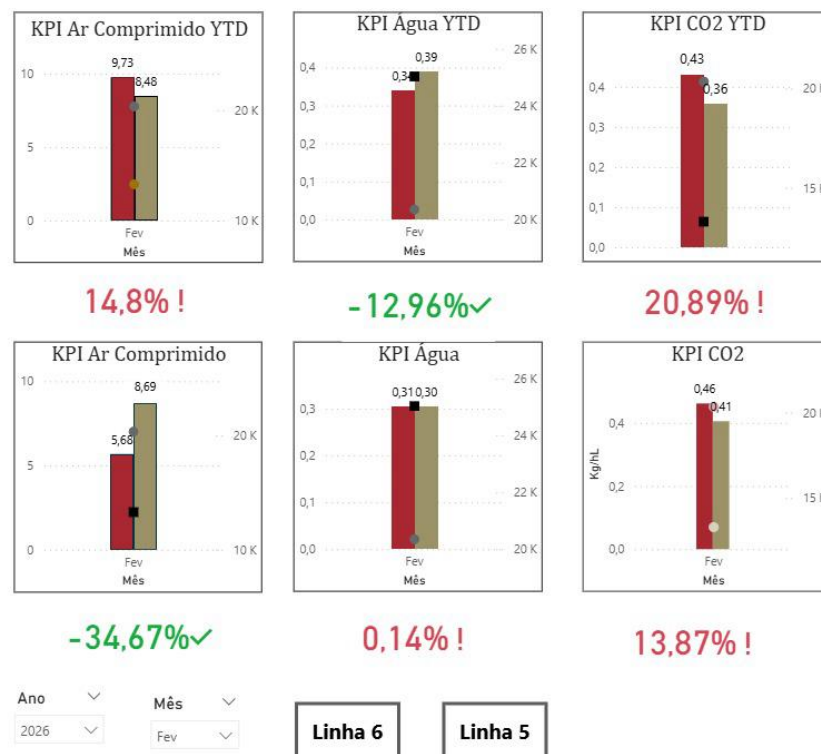


Figura 4.5: Versão intermédia em grelha 2×3: gráficos YTD e Mês separados por linha, com percentagens de variação por baixo de cada gráfico, sem codificação de cor por utilidade.

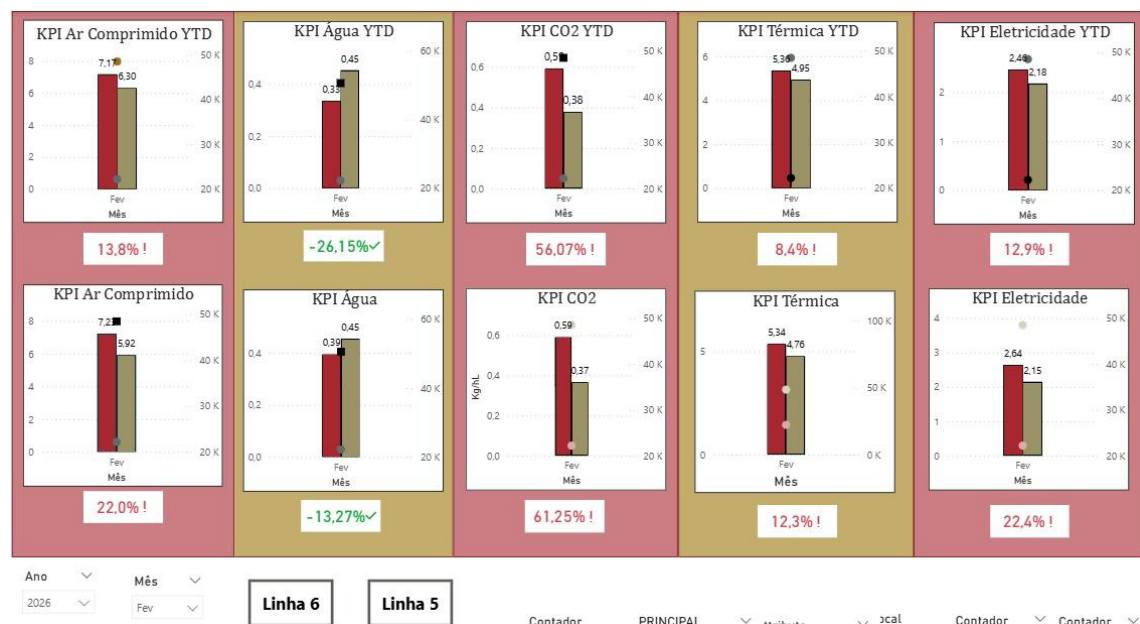


Figura 4.6: Versão intermédia com fundo colorido por utilidade: introdução da codificação cromática para distinção visual, com gráficos YTD e Mês ainda apresentados separadamente.

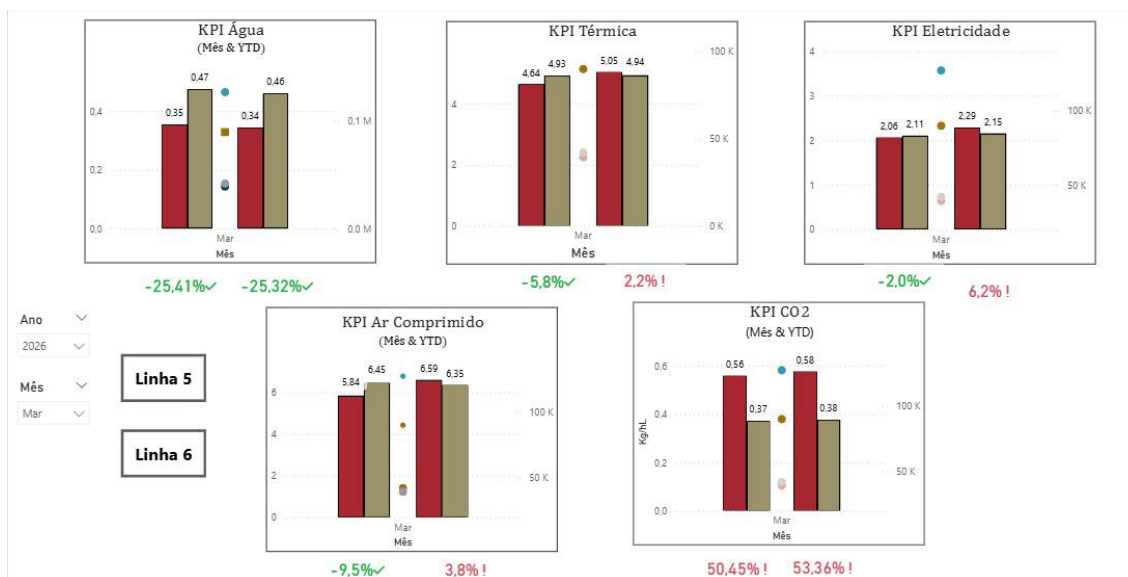


Figura 4.7: Versão intermédia com Mês e YTD integrados no mesmo gráfico: aproximação à solução final, com dois pares de barras por KPI e percentagens de variação por baixo, mas sem legendagem unificada nem navegação por linha.

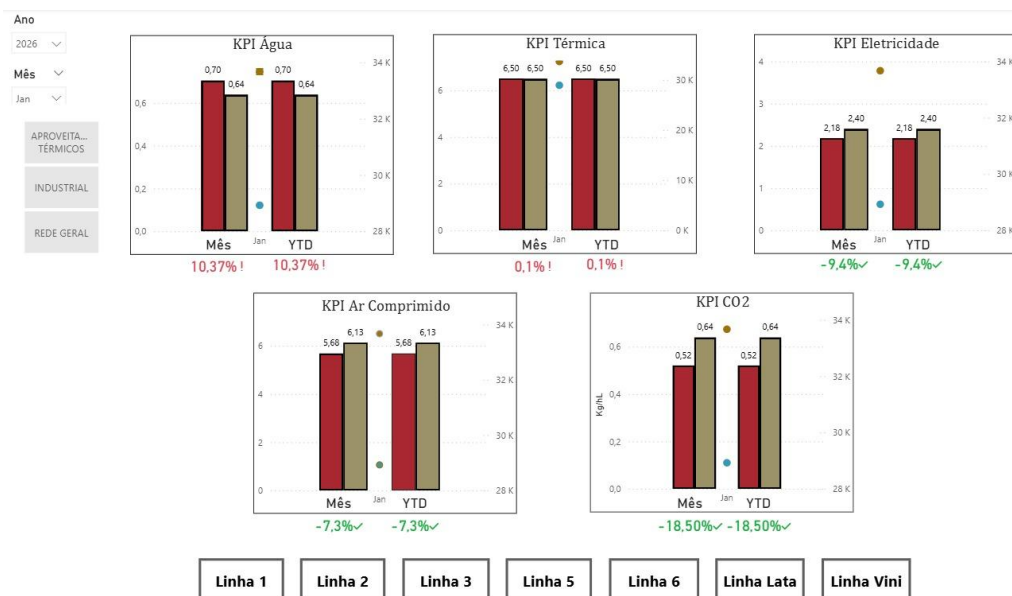


Figura 4.8: Versão pré-final: estrutura com botões de seleção de linha, filtros de Ano e Mês e gráficos Mês/YTD integrados. Nesta iteração os filtros de contadores ainda se encontravam expostos na parte inferior, sendo posteriormente ocultados na versão definitiva.

4.5.3 Estrutura da Solução Final

A Figura 4.9 apresenta os separadores de Análise Mensal criados no Power BI, correspondentes às diferentes páginas de monitorização.



Figura 4.9: Separadores das páginas de Análise Mensal criadas no Power BI: AM Enchimento, AM Linhas 4 & 7, AM Linha Vini, AM Produção, AM Adega e AM Fabrico.

A versão final da análise mensal concentra toda a informação em apenas seis páginas, cada uma dedicada a uma área produtiva distinta:

- **Página 1 — Linhas de Enchimento:** abrange as linhas L1, L2, L3, L5, L6 e Linha Lata. A navegação entre linhas é assegurada por um sistema de *bookmarks* ativados por botões, eliminando a necessidade de páginas separadas para cada linha e concentrando toda a análise de enchimento num único ecrã.
- **Página 2 — Linhas 4 e 7:** apresentadas em página própria devido às suas características produtivas distintas, que implicam uma lógica de análise diferenciada.
- **Página 3 — Linha Vini:** apresentada em página própria devido ao seu perfil de utilidades distinto, monitorizando apenas três KPIs (Água, Eletricidade e CO₂), sem monitorização de Térmica nem Ar Comprimido.
- **Página 4 — Produção:** vista agregada dos consumos das duas principais áreas produtivas (Adega e Fabrico), apresentando os cinco KPIs com botões de navegação para a análise detalhada de cada área.
- **Página 5 — Adegas:** reservada aos processos de adega, pela especificidade dos seus indicadores e dos consumos associados.
- **Página 6 — Fabrico:** dedicada à área de fabrico, que possui uma lógica de produção e de cálculo dos KPIs distinta das linhas de enchimento.

Esta estrutura reduziu o número de páginas a consultar de mais de 20 para apenas 6, representando uma simplificação radical do processo de análise e eliminando praticamente a totalidade do tempo anteriormente despendido na preparação da apresentação mensal.

4.5.3.1 Página de Linhas de Enchimento — Layout Unificado por *Bookmarks*

A solução final para as Linhas de Enchimento assenta num layout completamente uniforme entre todas as linhas. Independentemente da linha selecionada (L1, L2, L3, L5, L6 ou Linha Lata), o utilizador visualiza sempre o mesmo conjunto de cinco KPIs (Água, Térmica, Eletricidade, Ar Comprimido e CO₂), com a mesma estrutura gráfica e os mesmos indicadores de variação. Esta uniformização elimina qualquer curva de aprendizagem

e torna a análise comparativa entre linhas imediata. Cada gráfico apresenta no eixo horizontal duas categorias (Mês e YTD), cada uma com duas barras representando o ano corrente e o ano anterior, complementadas por pontos no eixo secundário que indicam os volumes de produção correspondentes.

A navegação entre linhas é implementada através de *bookmarks* do Power BI: cada botão (Linha 1, Linha 2, Linha 3, Linhas 4 & 7, Linha 5, Linha 6, Linha Lata, Linha Vini e Geral) está associado a um marcador predefinido que aplica os filtros correspondentes a todos os visuais da página de forma simultânea, preservando os filtros de Ano e Mês previamente selecionados. Os botões de sub-filtro (Aproveitamentos Térmicos, Industrial, Rede Geral e Captações) atuam exclusivamente sobre o KPI de Água, permitindo segmentar os valores por sub-contador e identificar ineficiências em subcircuitos específicos sem perder a visão global dos restantes consumos.

Note-se que as Figuras 4.10 a 4.15 foram capturadas com o mês de **março de 2026** selecionado, permitindo ilustrar de forma mais completa a lógica de apresentação dos KPIs, ao contrário de janeiro, em que os valores mensais e YTD coincidem.

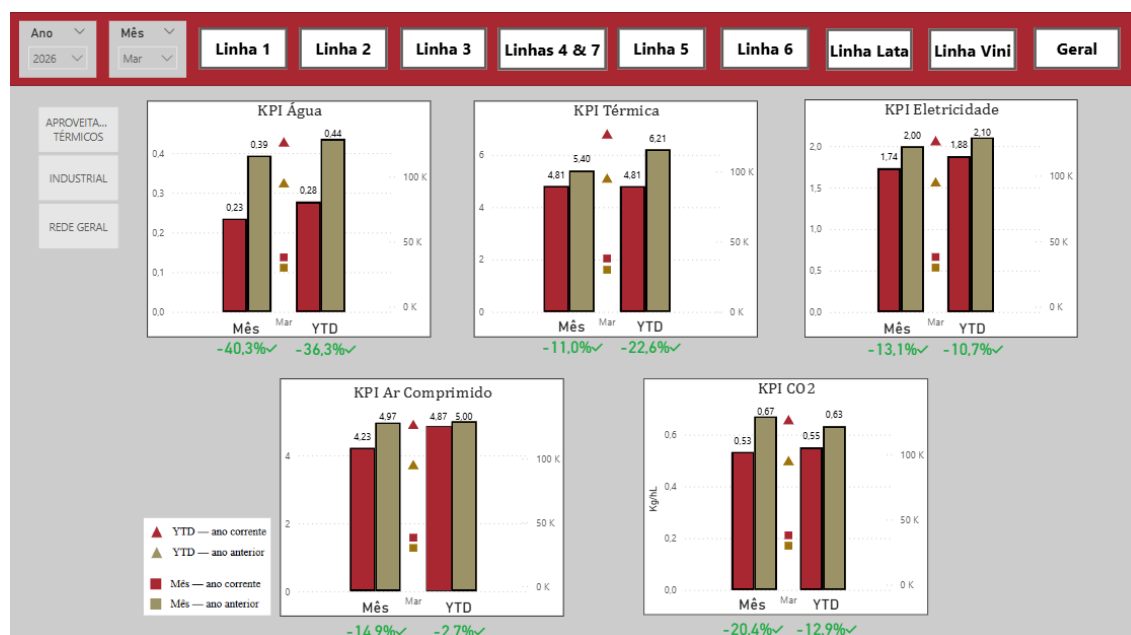


Figura 4.10: Versão final da página de análise mensal de KPIs para as Linhas de Enchimento: botões de seleção de linha no topo, filtros de Ano e Mês, botões de sub-contador (Aproveitamentos Térmicos, Captações), cinco KPIs (Água, Térmica, Eletricidade, Ar Comprimido e CO₂) com gráficos Mês/YTD integrados e indicadores de variação percentual com código de cor.

4.5.3.2 Página das Linhas 4 e 7 — Layout Diferenciado

As Linhas 4 e 7 apresentam características produtivas distintas das restantes linhas de enchimento, nomeadamente diferenças na tipologia de produto, nos consumos específicos

associados e na forma como a eletricidade é desagregada por equipamento, pelo que foi criada uma página de análise dedicada com um layout adaptado, apresentada na Figura 4.11. Ao contrário das linhas de enchimento standard, que partilham um layout uniforme com cinco KPIs idênticos, esta página apresenta os KPIs de Eletricidade de forma separada para cada linha (KPI Eletricidade Linha 4 e KPI Eletricidade Linha 7), mantendo os restantes KPIs (Água, Térmica, Ar Comprimido e CO₂) agregados para o conjunto das duas linhas. A navegação de regresso às Linhas de Enchimento é assegurada pelo botão *Pág. Anterior*.

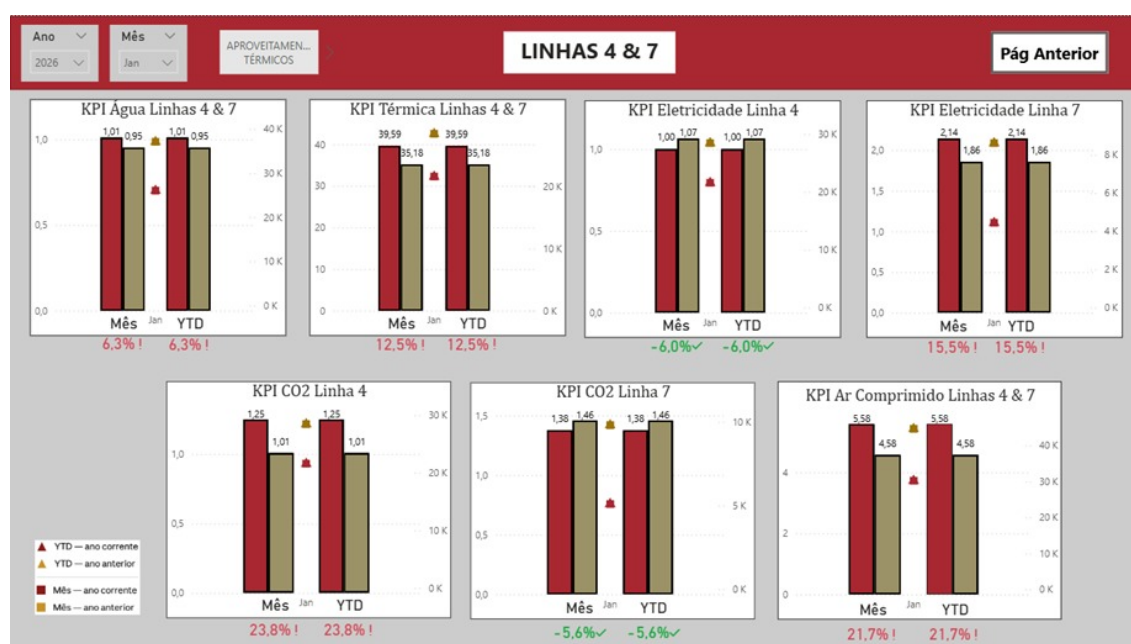


Figura 4.11: Página dedicada às Linhas 4 e 7, com layout diferenciado: os KPIs de Água, Térmica, Ar Comprimido e CO₂ são apresentados de forma agregada para o conjunto das duas linhas, enquanto o KPI de Eletricidade é desagregado individualmente para cada linha, refletindo as diferenças no perfil de consumo elétrico.

4.5.3.3 Página da Linha Vini — Layout Simplificado

A Linha Vini apresenta um perfil de utilidades distinto das restantes linhas de enchimento, não dispondo de monitorização de Energia Térmica nem de Ar Comprimido. Por este motivo, foi criada uma página de análise dedicada com um layout simplificado (Figura 4.12), adaptado às utilidades efetivamente monitorizadas nesta linha. A página apresenta apenas três KPIs (Água, Eletricidade e CO₂), mantendo a mesma lógica de visualização das restantes páginas de detalhe: cada KPI é representado por um gráfico de barras com os valores do mês em análise e do acumulado anual (YTD), acompanhados pelo valor do ano anterior para efeitos comparativos. Os botões de sub-filtro disponíveis (Aproveitamentos Térmicos, Industrial e Rede Geral) funcionam exclusivamente sobre o KPI de Água, permitindo uma granularidade adicional na análise do consumo hídrico.

desta linha. A navegação de regresso à página de Linhas de Enchimento é assegurada pelo botão *Pág. Anterior*.

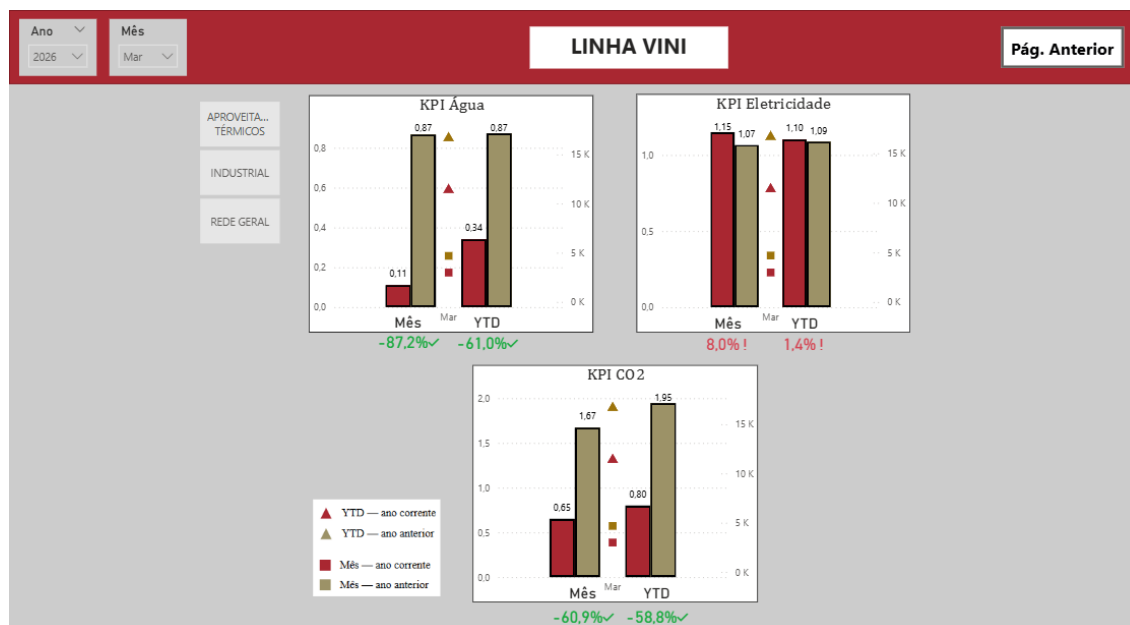


Figura 4.12: Página dedicada à Linha Vini, com layout simplificado: são apresentados apenas os três KPIs monitorizados nesta linha (Água, Eletricidade e CO₂), refletindo a ausência de consumos de Energia Térmica e Ar Comprimido. A navegação de regresso às Linhas de Enchimento é assegurada pelo botão *Pág. Anterior*.

4.5.3.4 Página de Produção — Vista Agregada

Foi ainda introduzida uma página de **Produção** que agrega, numa única vista consolidada, os consumos de utilidades das duas principais áreas produtivas da unidade: Adega e Fabrico. Esta página permite obter uma leitura imediata do desempenho energético e hídrico global da produção sem necessidade de consultar individualmente cada área, sendo possível navegar diretamente para as páginas de detalhe através dos botões *Adega* e *Fabrico* disponíveis no canto superior direito. A Figura 4.13 apresenta esta página.

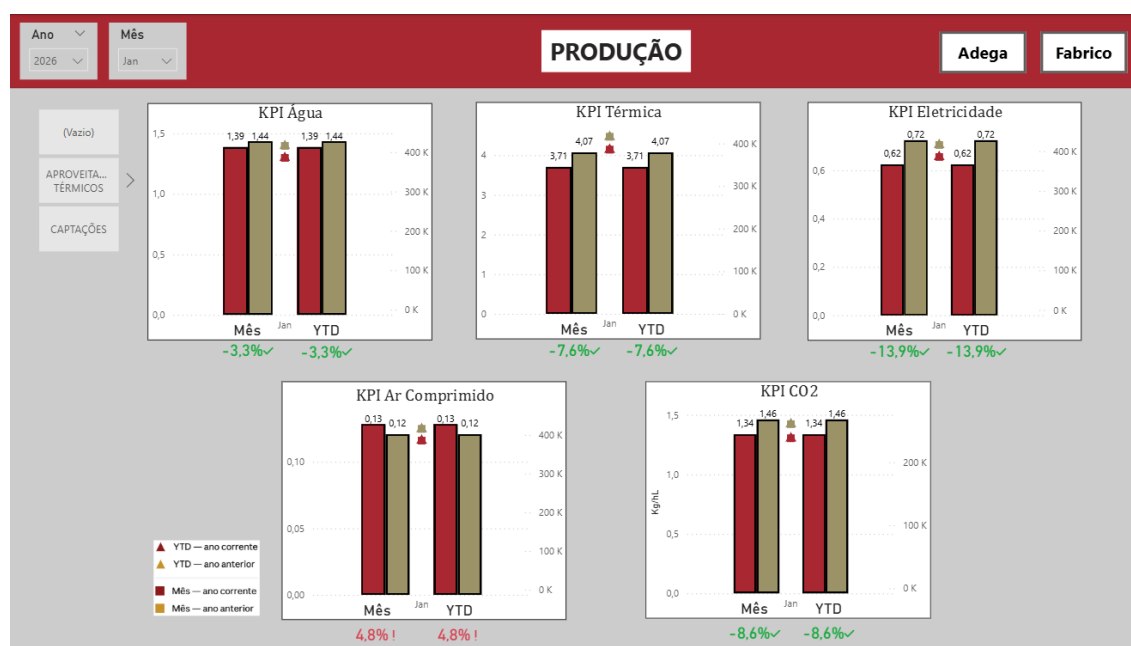


Figura 4.13: Página de Produção da análise mensal de KPIs, com visão agregada dos consumos de Adega e Fabrico combinados, apresentando os cinco KPIs (Água, Térmica, Eletricidade, Ar Comprimido e CO₂) e botões de navegação para a análise detalhada por área.

4.5.3.5 Páginas de Adega e Fabrico

As páginas de Adega e Fabrico seguem a mesma lógica de apresentação que as restantes páginas de análise mensal, com os KPIs adaptados às especificidades de cada área. A página da Adega (Figura 4.14) apresenta os cinco KPIs completos (Água, Térmica, Eletricidade, Ar Comprimido e CO₂), uma vez que os processos de fermentação e conservação tornam o CO₂ um indicador crítico para a monitorização do desempenho ambiental. A página do Fabrico (Figura 4.15) apresenta quatro KPIs (Água, Térmica, Eletricidade e Ar Comprimido), excluindo o CO₂ dado que este não é um indicador relevante nesta área. Em ambas as páginas, os sub-filtros de contexto permitem uma análise por sub-contador, mantendo a coerência metodológica com as páginas de enchimento. A navegação entre as páginas é assegurada pelos botões Pág. Fabrico e Pág. Adega, disponíveis em cada uma das páginas, como demonstram as Figuras 4.14 e 4.15.

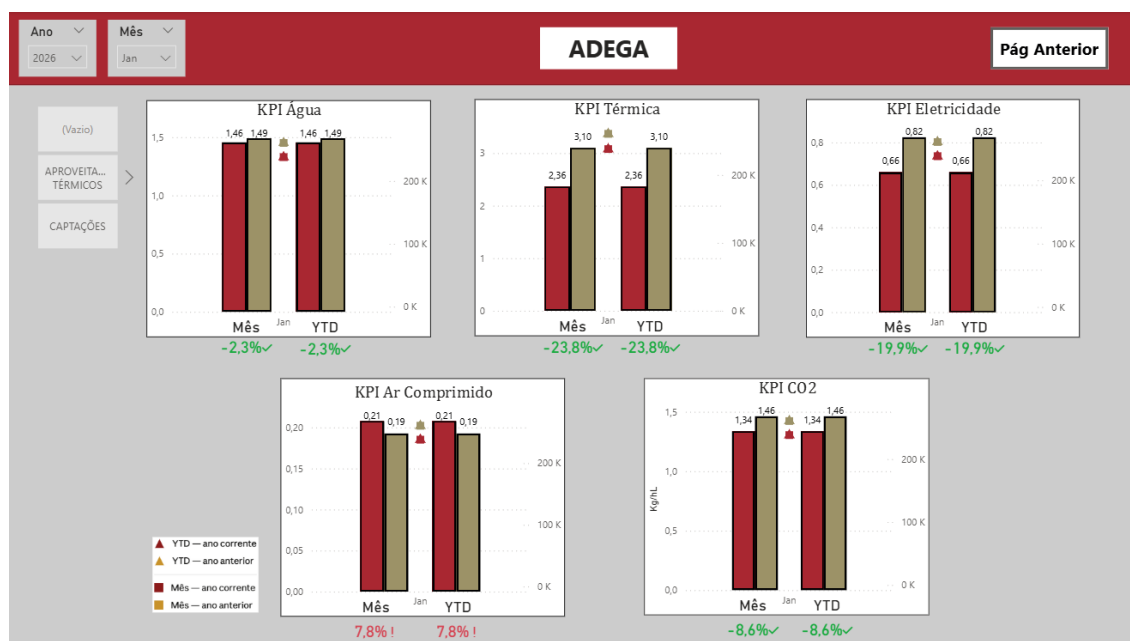


Figura 4.14: Página de Adeegas da análise mensal de KPIs, com os cinco KPIs completos.

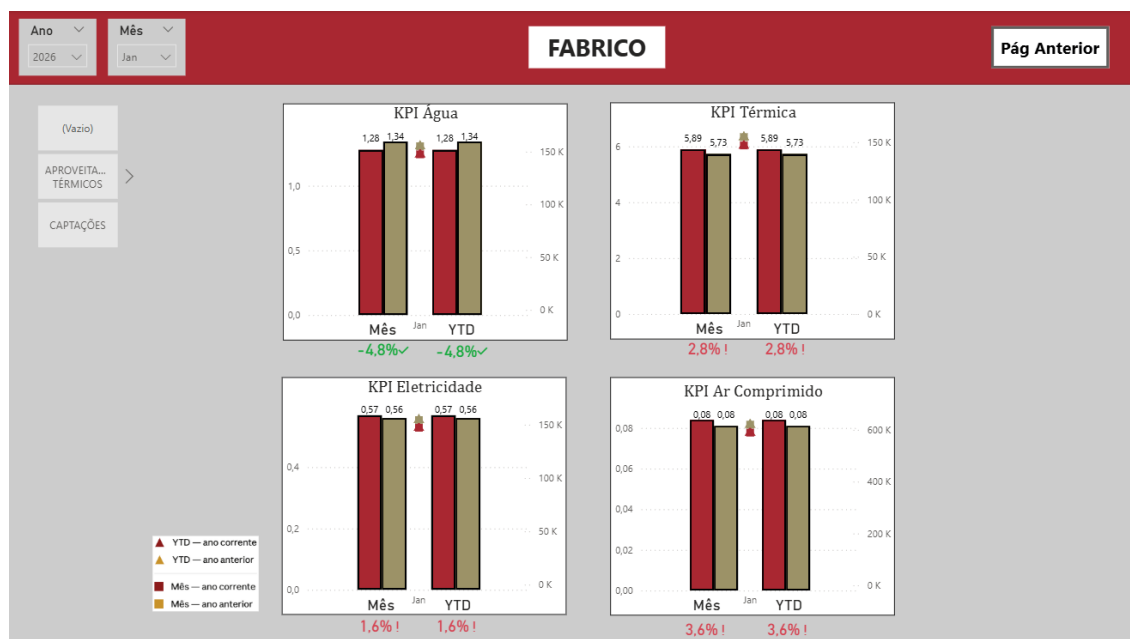


Figura 4.15: Página de Fabrico da análise mensal de KPIs; a ausência do KPI de CO₂ reflete a irrelevância deste indicador para a área de fabrico.

4.5.4 Lógica Visual dos Gráficos KPI

Cada KPI é representado por um único gráfico de barras com duas categorias no eixo horizontal: o mês corrente (Mês) e o acumulado desde o início do ano (YTD). Para cada categoria, são apresentadas duas barras (a do ano corrente, a vermelho, e a do ano anterior, a dourado/bege), permitindo a comparação direta entre períodos homólogos, tanto a nível mensal como acumulado.

O volume de produção (em hectolitros) é representado no eixo secundário sob a forma de pontos, fornecendo o contexto necessário para a interpretação dos valores de consumo específico. A legenda é unificada para todos os KPIs e todas as páginas: triângulos identificam os valores YTD, quadrados identificam os valores do mês, vermelho corresponde ao ano corrente e laranja/dourado ao ano anterior. Esta consistência visual elimina qualquer curva de aprendizagem, dado que o utilizador não necessita de reaprender a leitura de cada gráfico ao navegar entre páginas [18].

4.5.5 Indicadores de Variação Percentual com Código de Cor

Uma das inovações centrais desta reformulação foi a introdução de indicadores de variação percentual com código de cor e símbolo, posicionados por baixo de cada gráfico KPI. Estes indicadores comparam o valor atual com o do ano anterior e comunicam o sentido da variação através de dois estados visuais distintos:

- **Verde com símbolo ✓** — indica uma melhoria face ao ano anterior, ou seja, um consumo específico mais baixo e uma maior eficiência energética;
- **Vermelho com símbolo !** — indica um agravamento face ao ano anterior, correspondendo a um consumo específico mais elevado e a uma menor eficiência.

A variação é apresentada tanto para o mês corrente como para o YTD, permitindo distinguir se a variação observada é pontual ou se reflete uma tendência acumulada ao longo do ano. Esta dupla perspetiva temporal é particularmente útil em contexto de reunião, onde é frequentemente necessário contextualizar um resultado mensal no quadro do desempenho global do ano.

4.5.6 Implementação Técnica — Medidas DAX

Do ponto de vista técnico, a reformulação da análise mensal exigiu a criação de um conjunto extenso de medidas DAX, organizadas segundo um padrão sistemático e replicável para cada zona produtiva e cada tipo de consumo monitorizado.

Para cada combinação de zona (Linhas de Enchimento, Adeegas, Fabrico) e de utilidade (Água, Energia Térmica/Vapor, Eletricidade, Ar Comprimido e CO₂), foram criadas quatro categorias de medidas:

- **Medidas de variação percentual do mês**, que comparam o KPI do mês corrente com o do mesmo mês do ano anterior;
- **Medidas de variação percentual do YTD**, que comparam o KPI acumulado do ano corrente com o acumulado até ao mesmo período do ano anterior;
- **Medidas YTD do ano corrente**, calculadas com recurso à função `DATESYTD`, que acumula os valores desde o início do ano até à data selecionada;
- **Medidas YTD do ano anterior**, obtidas combinando `DATESYTD` com `SAMEPERIODLASTYEAR`, deslocando o contexto temporal para o período homólogo.

A utilização de `DATESYTD` garante que os valores acumulados são recalculados dinamicamente em função dos filtros de Ano e Mês selecionados, sem necessidade de intervenção manual [15]. A função `SAMEPERIODLASTYEAR` assegura que a comparação é sempre feita com o período exatamente equivalente do ano anterior.

Devido à extensão do código desenvolvido, as medidas DAX completas foram remetidas para o Anexo C. No corpo do capítulo mantém-se apenas a descrição da lógica de implementação, de forma a preservar a fluidez da leitura e concentrar a discussão nos objetivos funcionais da solução.

Em síntese, as medidas foram organizadas em quatro grupos principais: Linhas de Enchimento, Produção, Adega e Fabrico. Em todos os casos, a lógica combina o consumo de cada utilidade com o respetivo denominador operacional, como volume enchido ou produção da área, permitindo obter indicadores normalizados e comparáveis entre períodos. As funções `DATESYTD` e `SAMEPERIODLASTYEAR` foram utilizadas para calcular os valores acumulados e as comparações homólogas, enquanto as medidas percentuais suportam a codificação visual verde/vermelho usada nos cartões KPI do dashboard.

4.6 Discussão e Mais-valias

As reformulações descritas neste capítulo traduziram-se em melhorias significativas e mensuráveis no processo de análise e monitorização dos consumos de utilidades.

Do ponto de vista quantitativo, o número de páginas necessárias para a análise mensal foi reduzido de mais de 20 para apenas 6, e o tempo de preparação passou de cerca de um dia de trabalho para aproximadamente uma hora. É fundamental salvaguardar que estes indicadores de tempo e volume não decorrem de um estudo cronometrado formal ou de uma análise estatística de tempos e movimentos. Pelo contrário, representam uma estimativa validada operacionalmente e baseada no histórico de rotina e na experiência prática reportada pela colaboradora responsável pelo posto de trabalho¹. Este ganho de eficiência

¹Os valores de tempo e volume reportados baseiam-se na experiência prática e estimativa validada com a colaboradora responsável pelo posto de trabalho, refletindo o ganho operacional percebido após a implementação da solução.

percebido liberta recursos para atividades de maior valor acrescentado, nomeadamente a análise crítica e a interpretação dos dados, em detrimento de tarefas operacionais de compilação.

Do ponto de vista qualitativo, a filtragem dinâmica por linha de produção, a navegação baseada em *bookmarks* e a integração de indicadores de variação percentual com codificação cromática melhoraram a usabilidade e a capacidade analítica do dashboard. A análise passou a ser realizável de forma interativa e em tempo real, diretamente durante a reunião mensal, garantindo que todos os participantes acedem à mesma informação atualizada e consistente, eliminando o risco de discrepâncias associado ao processo anterior.

A consistência visual adotada ao longo de todas as páginas (legenda unificada, estrutura idêntica e código de cor rigoroso) permite que utilizadores com diferentes graus de familiaridade interpretem os resultados de forma autónoma e eficiente, facilitando igualmente a comunicação dos resultados a diferentes níveis hierárquicos da organização.

Em síntese, o dashboard reformulado constitui uma ferramenta mais eficaz, robusta e orientada à decisão para o acompanhamento operacional dos consumos de utilidades. Para além de responder diretamente às limitações identificadas na versão anterior, a solução desenvolvida contribui para uma abordagem mais integrada e eficiente da gestão de dados, alinhada com os objetivos de monitorização, otimização de recursos e suporte à decisão definidos no âmbito deste trabalho.

Tabela 4.1: Comparação entre o processo anterior e as soluções implementadas.

Indicador	Antes	Depois
Atualização das leituras	Manual	Automática
Deteção de consumos anómalos	Visual	Automática
Envio de alertas	Não existente	Diário por e-mail
Tempo de reação a anomalias	Após análise manual	No próprio dia
Dashboard Power BI	Estático e disperso	Reformulado e interativo
Número de páginas no Power BI	Mais de 20	6
Preparação da análise mensal	Cerca de 1 dia	Cerca de 1 hora
Risco de erro humano	Elevado	Reduzido

Capítulo 5

Conclusões e Trabalhos Futuros

Este capítulo sintetiza os principais resultados do trabalho desenvolvido e apresenta as direções de continuidade que dele decorrem.

5.1 Conclusões

O presente trabalho teve como objetivo determinar em que medida a automatização do tratamento de dados e a integração de ferramentas de Business Intelligence poderiam contribuir para uma monitorização mais fiável e mais reativa do consumo de utilidades industriais numa unidade de produção cervejeira.

O diagnóstico inicial identificou limitações significativas no processo existente: a recolha e o tratamento manual dos dados exigiam um esforço operacional elevado, aumentavam a probabilidade de erros de transcrição e atrasavam a deteção de consumos anómalos, comprometendo tanto a fiabilidade da informação disponível como a rapidez da resposta operacional.

Para responder a estas limitações, foi desenvolvida uma solução integrada, assente na automatização do tratamento e validação dos dados através de Power Automate e Office Scripts, complementada por mecanismos automáticos de alarmística e pela reformulação do dashboard de monitorização em Power BI. A solução foi concebida para se integrar no ecossistema Microsoft 365 já utilizado pela empresa, o que garantiu simplicidade de implementação, facilidade de manutenção e reduzida necessidade de infraestrutura adicional.

Quanto à pergunta de investigação central, centrada na deteção de anomalias, os resultados obtidos mostram que a automatização do tratamento de dados e a integração de ferramentas de Business Intelligence tornaram essa deteção mais fiável e mais reativa face ao processo manual anterior. A sinalização de leituras inconsistentes e de consumos fora dos limites de referência deixou de depender exclusivamente da observação humana, passando a assentar em regras aplicadas de forma consistente a todos os registos diários. Este efeito é corroborado pela experiência da colaboradora responsável pela validação das

leituras, que identifica diariamente desvios sinalizados automaticamente: alguns sem significado operacional, sendo descartados após análise, e outros correspondentes a erros de registo ou consumos que antes só eram detetados por observação visual direta. Não foi feito um registo sistemático do número de anomalias corretamente sinalizadas, pelo que não é possível apresentar uma taxa de deteção; ainda assim, a evidência qualitativa disponível aponta para uma redução clara da dependência da observação humana, sustentando de forma mais direta a resposta à pergunta de investigação. A reatividade foi reforçada pela implementação de alertas diários automáticos e pela disponibilização imediata da informação em dashboards interativos, permitindo identificar mais depressa os desvios já sinalizados.

A definição de limites de referência baseados na análise histórica dos consumos revelou-se também adequada ao contexto estudado. Face a metodologias estatísticas mais sofisticadas de deteção de anomalias, a solução adotada privilegiou critérios simples, transparentes e facilmente interpretáveis pelos utilizadores finais, o que facilitou a sua validação operacional e a aceitação pelas equipas responsáveis pela monitorização diária.

Os resultados percebidos pela colaboradora responsável pela análise mensal confirmam o impacto positivo da solução: o processo passou de mais de vinte páginas para seis, e o tempo de preparação de cerca de um dia de trabalho para aproximadamente uma hora. Estes valores não resultam de um estudo cronometrado nem de um registo sistemático ao longo do período de utilização; correspondem a uma estimativa validada operacionalmente, baseada na experiência prática da utilizadora final, e devem ser lidos como indicativos da magnitude do ganho de eficiência, não como um resultado quantitativo consolidado. Ainda assim, evidenciam um alívio real do esforço administrativo: a automatização eliminou tarefas repetitivas, reduziu o risco de erro humano por transcrição e disponibilizou informação de forma mais célere e consistente para apoio à decisão.

Do ponto de vista científico, esta dissertação mostra que soluções de digitalização baseadas em ferramentas amplamente disponíveis podem produzir melhorias relevantes na gestão de utilidades industriais quando desenvolvidas a partir de uma análise detalhada do processo existente e adaptadas às necessidades da organização. O principal contributo do trabalho não está na utilização isolada de tecnologias digitais, mas na integração coerente entre automatização, validação de dados, alarmística e visualização analítica, uma metodologia adaptável a outros contextos industriais com necessidades semelhantes.

Em síntese, os objetivos definidos para esta dissertação foram atingidos: desenvolveu-se e validou-se, em contexto industrial real, uma solução que tornou a deteção de anomalias e a monitorização das utilidades mais fiáveis e mais reativas, reforçando o suporte à decisão e a gestão operacional dos recursos.

5.2 Trabalhos Futuros

O trabalho desenvolvido abre várias linhas de continuidade que permitiriam alargar e aprofundar as soluções implementadas.

Uma primeira direção consiste em estender o sistema de monitorização e de alertas à análise semanal dos indicadores de desempenho (KPIs) em Power BI. Esta análise não foi concretizada no presente trabalho porque os volumes de enchimento não são atualizados diariamente, o que impedia o cálculo e a sinalização dos indicadores com a periodicidade pretendida. À data, a empresa encontra-se em fase de viabilização do acesso a estes dados de forma mais frequente, pelo que a sua disponibilização permitirá replicar a abordagem já implementada para o acompanhamento semanal dos KPIs.

Uma segunda direção passa por incorporar técnicas de machine learning na deteção de anomalias nos valores de consumo das utilidades. A aprendizagem dos padrões históricos de consumo permitiria definir limites adaptativos, em vez de limites fixos, e distinguir com maior rigor os consumos efetivamente anómalos das variações normais de operação. Esta evolução tornaria os alertas mais precisos e eficientes, reduzindo falsos positivos e melhorando a capacidade de antecipação do sistema já implementado.

5.2.1 Integração de Controlo Estatístico de Processos (CEP)

A solução desenvolvida assenta em limites de referência fixos, estabelecidos a partir de máximos históricos operacionais, os quais demonstraram eficácia na sinalização de erros de leitura e de picos de consumo evidentes. Contudo, face à variabilidade intrínseca das utilidades industriais, esta abordagem revela-se reativa e sensível a valores extremos isolados, não permitindo identificar desvios sistemáticos nem perdas graduais de eficiência. A evolução natural deste trabalho reside, por conseguinte, na transição para um paradigma preditivo e analítico, através da implementação de metodologias de Controlo Estatístico de Processos (CEP) [1].

O ecossistema digital instituído, composto pela centralização diária de dados no Share-Point e pela respetiva modelação em Power BI, reúne os pré-requisitos necessários para esta expansão. A aplicação de cartas de controlo de Shewhart, nomeadamente cartas de leituras individuais acompanhadas por cartas de amplitude móvel, focar-se-á, numa fase inicial, na utilidade de ar comprimido, dado o seu elevado impacto energético e a sua suscetibilidade à ocorrência de fugas latentes na rede de distribuição. Os limites estatísticos de controlo serão calculados dinamicamente a partir do histórico de consumo já consolidado pelo sistema atual, em substituição dos máximos absolutos hoje utilizados. Ao transpor estes limites para cartões analíticos no dashboard e para regras automatizadas no Power Automate, o sistema deixará de emitir alertas baseados apenas em violações pontuais de máximos operacionais, passando a detetar padrões estatisticamente anómalos, como sequências de sete pontos consecutivos acima ou abaixo da média ou tendências de subida contínua ao longo de vários turnos operacionais.

A concretização desta metodologia enfrenta, todavia, desafios de ordem prática que exigem uma validação prudente. O volume e a qualidade do histórico de dados constituem o primeiro obstáculo: períodos de paragem programada da fábrica ou anomalias severas de contadores devem ser previamente expurgados da amostra de calibração, para evitar a distorção dos limites calculados. Adicionalmente, devido à sazonalidade comprovada nas áreas de adegas e enchimento, os limites estatísticos não poderão ser estáticos, requerendo uma recalibração periódica ou a definição de janelas estatísticas móveis ajustadas ao perfil de produção da instalação. A transição para o CEP deverá, portanto, ser conduzida de forma faseada, garantindo a estabilização prévia do processo, de modo a evitar a proliferação de falsos alarmes que comprometam a fiabilidade operacional do sistema de alarmística perante as equipas de manutenção.

Referências

- [1] Douglas C. Montgomery. *Introduction to Statistical Quality Control*. John Wiley & Sons, Hoboken, NJ, 8.^a edição, 2020.
- [2] Wolfgang Kunze. *Technology Brewing and Malting*. VLB Berlin, Berlim, 6.^a edição, 2019.
- [3] Dennis E. Briggs, Chris A. Boulton, Peter A. Brookes, e Roger Stevens. *Brewing: Science and Practice*. Woodhead Publishing, Cambridge, 2004.
- [4] R. Saidur, N. A. Rahim, e M. Hasanuzzaman. A review on compressed-air energy use and energy savings. *Renewable and Sustainable Energy Reviews*, 14(4):1135–1153, 2010.
- [5] Ana Abreu, Paula Afonso, e Simone Zanni. Energy management and industrial symbiosis towards cleaner production. *Journal of Cleaner Production*, 213:80–92, 2019.
- [6] The Brewers of Europe. Beer statistics 2022 edition. Relatório técnico, The Brewers of Europe, Bruxelas, Bélgica, 2022.
- [7] L. Fillaudeau, P. Blanpain-Avet, e G. Daufin. Water, wastewater and waste management in brewing industries. *Journal of Cleaner Production*, 14(5):463–471, 2006.
- [8] B. Sturm, S. Hugenschmidt, S. Joyce, W. Hofacker, e A. P. Roskilly. Opportunities and barriers for efficient energy use in a medium-sized brewery. *Applied Thermal Engineering*, 53(2):397–404, 2013.
- [9] World Resources Institute e World Business Council for Sustainable Development. The greenhouse gas protocol: A corporate accounting and reporting standard. Relatório técnico, GHG Protocol, 2004.
- [10] International Organization for Standardization. *ISO 50001:2018 – Energy Management Systems – Requirements with Guidance for Use*. ISO Standard, 2018.
- [11] Michal Lom, Ondrej Pribyl, e Miroslav Svitek. Industry 4.0 as a part of smart cities. Em *Smart Cities Symposium Prague*, páginas 1–6, 2016.
- [12] Stuart A. Boyer. *SCADA: Supervisory Control and Data Acquisition*. ISA, Research Triangle Park, NC, 4.^a edição, 2010.
- [13] Jürgen Kletti. *Manufacturing Execution System – MES*. Springer, Berlim, 2007.
- [14] Wayne W. Eckerson. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. John Wiley & Sons, Hoboken, NJ, 2.^a edição, 2010.

- [15] Alberto Ferrari e Marco Russo. *Introducing Microsoft Power BI*. Microsoft Press, Redmond, WA, 2016.
- [16] Fadi Shrouf e Giovanni Miragliotta. Energy management based on Internet of Things: practices and framework for adoption in production management. *Journal of Cleaner Production*, 100:235–246, 2015.
- [17] Konstantin Vikhorev, Richard Greenough, e Neil Brown. An advanced energy management framework to promote energy awareness. *Journal of Cleaner Production*, 43:103–112, 2013.
- [18] Stephen Few. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, Sebastopol, CA, 2006.
- [19] Eoin O'Driscoll e Garret E. O'Donnell. Industrial power and energy metering — a state-of-the-art review. *Journal of Cleaner Production*, 41:53–64, 2013.

Anexo A

Automatização da Importação de Dados

Este anexo reúne o código desenvolvido para a automatização da importação das leituras dos ficheiros da subcontratada para os ficheiros da empresa, descrita na Secção 3.3.2. A solução passou por duas fases: uma primeira implementação local em VBA (Anexo A.1) e a versão final, executada inteiramente na cloud, em Office Scripts orquestrados pelo Power Automate (Anexo A.2). Ambas aplicam a mesma lógica de atualização, diferindo na arquitetura de execução.

A.1 Fase 1 — Macro VBA

O código seguinte apresenta a estrutura-base da macro VBA desenvolvida para automatizar a importação de leituras a partir dos ficheiros Excel sincronizados via OneDrive/SharePoint. A mesma lógica foi adaptada aos diferentes ficheiros de utilidade, alterando apenas o caminho do ficheiro de origem, os nomes das folhas e as colunas a copiar.

```
1 Sub AtualizarLeituras()  
2   Dim caminhoOrigem As String  
3   Dim caminhoTemp As String  
4   caminhoOrigem = "C:\Users\Tomas\Super Bock Group\Energia e Utilidades - Contagens Diarias\  
   Agua.xlsb"  
5   caminhoTemp = Environ("TEMP") & "\Agua_" & Format(Now, "hhmmss") & ".xlsb"  
6   Dim wbOrigem As Workbook  
7   Dim wsOrigem As Worksheet  
8   Dim wsDestino As Worksheet  
9   Dim ultimaData As Date  
10  Dim ultimaLinhaOrigem As Long  
11  Dim ultimaLinhaDestino As Long  
12  Dim ultimaColuna As Long  
13  Dim i As Long  
14  Dim j As Long
```

```

15 Dim k As Long
16 Dim dataLinha As Date
17 Dim linhasCopiadas As Integer
18 Dim linhasAtualizadas As Integer
19 Dim linhaDestino As Long
20 Dim tentativas As Integer
21 Application.ScreenUpdating = False
22 Application.DisplayAlerts = False
23 Application.AutomationSecurity = msoAutomationSecurityForceDisable
24 Application.EnableEvents = False
25 ' --- Forçar sincronização OneDrive ---
26 MsgBox "A forçar sincronização do OneDrive..." & Chr(10) & "Aguarde...", vbInformation, "A
    sincronizar..."
27 Shell "powershell -command ""$env:LOCALAPPDATA + '\Microsoft\OneDrive\OneDrive.exe' /
    forceSync""", vbHide
28 Application.Wait Now + TimeValue("00:00:15")
29 tentativas = 0
30 Do While tentativas < 24
31
32 If Dir(caminhoOrigem) <> "" Then
33     On Error Resume Next
34     FileCopy caminhoOrigem, caminhoTemp
35     If Err.Number = 0 Then Exit Do
36     Err.Clear
37     On Error GoTo 0
38 End If
39 tentativas = tentativas + 1
40 Application.Wait Now + TimeValue("00:00:05")
41 Loop
42 If tentativas = 24 Then
43     Application.ScreenUpdating = True
44     Application.DisplayAlerts = True
45     Application.AutomationSecurity = msoAutomationSecurityByUI
46     Application.EnableEvents = True
47     MsgBox "Não foi possível aceder ao ficheiro após 2 minutos." & Chr(10) & _
48         "Verifique a ligação ao OneDrive.", vbCritical, "Erro de sincronização"
49     Exit Sub
50 End If
51 Set wsDestino = ThisWorkbook.Sheets("Leituras")
52 ultimaLinhaDestino = wsDestino.Cells(wsDestino.Rows.Count, 1).End(xlUp).Row
53 ultimaColuna = wsDestino.Cells(1, wsDestino.Columns.Count).End(xlToLeft).Column
54 If ultimaLinhaDestino > 1 Then
55     ultimaData = CDate(wsDestino.Cells(ultimaLinhaDestino, 1).Value)
56 Else
57     ultimaData = CDate("01/01/2000")
58 End If
59 MsgBox "Última data registada: " & Format(ultimaData, "dd/mm/yyyy") & Chr(10) & "A ir buscar
    dados até " &
60     Format(Date, "dd/mm/yyyy") & "...", vbInformation, "A atualizar..."
61 On Error GoTo ErroAbertura
62 Set wbOrigem = Workbooks.Open( _
63     Filename:=caminhoTemp, _
64     ReadOnly:=True, _
65     UpdateLinks:=0, _
66     CorruptLoad:=xlNormalLoad)
67 On Error GoTo 0
68 Set wsOrigem = wbOrigem.Sheets("Leituras")

```

```

69 ultimaLinhaOrigem = wsOrigem.Cells(wsOrigem.Rows.Count, 1).End(xlUp).Row
70 linhasCopiadas = 0
71 linhasAtualizadas = 0
72 For i = 2 To ultimaLinhaOrigem
73     On Error Resume Next
74     dataLinha = CDate(wsOrigem.Cells(i, 1).Value)
75     On Error GoTo 0
76     If dataLinha <= Date Then
77         linhaDestino = 0
78         For k = 2 To ultimaLinhaDestino
79             On Error Resume Next
80             If Format(wsDestino.Cells(k, 1).Value, "dd/mm/yyyy") = Format(wsOrigem.Cells(i, 1).Value,
81                 "dd/
82                 mm/yyyy") Then
83                 linhaDestino = k
84                 Exit For
85             On Error GoTo 0
86         Next k
87         If linhaDestino = 0 And dataLinha > ultimaData Then
88             ultimaLinhaDestino = wsDestino.Cells(wsDestino.Rows.Count, 1).End(xlUp).Row + 1
89             wsOrigem.Rows(i).Copy Destination:=wsDestino.Rows(ultimaLinhaDestino)
90             ultimaData = dataLinha
91             linhasCopiadas = linhasCopiadas + 1
92         ElseIf linhaDestino > 0 Then
93             If dataLinha >= Date - 30 Then
94                 For j = 2 To ultimaColuna
95                     wsDestino.Cells(linhaDestino, j).Value = wsOrigem.Cells(i, j).Value
96                 Next j
97                 linhasAtualizadas = linhasAtualizadas + 1
98             Else
99                 Dim atualizado As Boolean
100                 atualizado = False
101                 For j = 2 To ultimaColuna
102                     If wsDestino.Cells(linhaDestino, j).Value = "" Then
103                         If wsOrigem.Cells(i, j).Value <> "" Then
104                             wsDestino.Cells(linhaDestino, j).Value = wsOrigem.Cells(i, j).Value
105                             atualizado = True
106                         End If
107                     End If
108                 Next j
109                 If atualizado Then linhasAtualizadas = linhasAtualizadas + 1
110             End If
111         End If
112     End If
113 Next i
114 wbOrigem.Close SaveChanges:=False
115 On Error Resume Next
116 Kill caminhoTemp
117 On Error GoTo 0
118 Application.ScreenUpdating = True
119 Application.DisplayAlerts = True
120 Application.AutomationSecurity = msoAutomationSecurityByUI
121 Application.EnableEvents = True
122 MsgBox "Atualização concluída!" & Chr(10) & Chr(10) & _
123     "Linhas novas adicionadas: " & linhasCopiadas & Chr(10) & _

```

```

125     "Linhas existentes atualizadas: " & linhasAtualizadas & Chr(10) & Chr(10) & _
126     "Dados atualizados até " & Format(Date, "dd/mm/yyyy"), _
127     vbInformation, "Atualização concluída"
128     Exit Sub
129 ErroAbertura:
130     Application.ScreenUpdating = True
131     Application.DisplayAlerts = True
132     Application.AutomationSecurity = msoAutomationSecurityByUI
133     Application.EnableEvents = True
134     MsgBox "Erro ao abrir o ficheiro." & Chr(10) & _
135     "Código do erro: " & Err.Number & Chr(10) & _
136     "Descrição: " & Err.Description, _
137     vbCritical, "Erro"
138 End Sub

```

Listagem A.1: Macro VBA para atualização automática de leituras em Excel.

A.2 Fase 2 — Office Scripts (Power Automate)

A versão final substitui a macro VBA por dois Office Scripts, acionados diariamente por um *trigger* de *Recurrence* do Power Automate: um script de leitura, que recolhe os dados do ficheiro de origem, e um script de escrita, que aplica a lógica de atualização no ficheiro de destino.

A.2.1 Leitura do ficheiro de origem (Ler Leituras)

```

1  function main(workbook: ExcelScript.Workbook, folhaLeituras: string): string {
2      const ws = workbook.getWorksheet(folhaLeituras);
3      if (!ws) {
4          return JSON.stringify({ erro: "Folha '" + folhaLeituras + "' não encontrada", dados: [] });
5      }
6      const range = ws.getUsedRange();
7      if (!range) {
8          return JSON.stringify({ erro: "Folha vazia", dados: [] });
9      }
10     const values = range.getValues();
11     const primeiraLinhaDados = encontrarPrimeiraLinhaDados(values);
12     // Filtrar para últimos 90 dias - evita JSON gigante
13     const DIAS_MAXIMOS = 90;
14     const hoje = new Date();
15
16     hoje.setUTCHours(0, 0, 0, 0);
17     const dataMinima = new Date(hoje.getTime() - DIAS_MAXIMOS * 86400000);
18     const dadosFiltrados: (string | number | boolean)[][] = [];
19     if (primeiraLinhaDados >= 0) {
20         for (let i = primeiraLinhaDados; i < values.length; i++) {
21             const d = parseData(values[i][0]);
22             if (d && d >= dataMinima) {
23                 dadosFiltrados.push(values[i]);

```

```

24     }
25   }
26 }
27 return JSON.stringify({
28   nome: workbook.getName(),
29   primeiraLinhaDados: primeiraLinhaDados,
30   cabecalhos: values.slice(0, Math.max(0, primeiraLinhaDados)),
31   dados: dadosFiltrados,
32   erro: ""
33 });
34 }
35 // Helpers
36 function encontrarPrimeiraLinhaDados(values: (string | number | boolean)[][]): number {
37   for (let i = 0; i < values.length; i++) {
38     if (parseData(values[i][0]) !== null) return i;
39   }
40   return -1;
41 }
42 function parseData(v: string | number | boolean): Date | null {
43   if (v === null || v === undefined) return null;
44   if (typeof v === "number") {
45     if (v > 36000 && v < 80000) {
46       return new Date(Date.UTC(1899, 11, 30) + Math.floor(v) * 86400000);
47     }
48     return null;
49   }
50   if (typeof v !== "string") return null;
51   const s = v.trim();
52   if (!s) return null;
53   let m = s.match(/^(\d{1,2})[\/\-\.](\d{1,2})[\/\-\.](\d{4})(?:[sT].*)?$/);
54   if (m) {
55     const dia = parseInt(m[1]);
56     const mes = parseInt(m[2]);
57     const ano = parseInt(m[3]);
58     if (dia >= 1 && dia <= 31 && mes >= 1 && mes <= 12 && ano >= 1900 && ano <= 2200) {
59       return new Date(Date.UTC(ano, mes - 1, dia));
60     }
61   }
62   m = s.match(/^(\d{4})[\/\-\.](\d{1,2})[\/\-\.](\d{1,2})(?:[sT].*)?$/);
63   if (m) {
64     const ano = parseInt(m[1]);
65     const mes = parseInt(m[2]);
66     const dia = parseInt(m[3]);
67     if (dia >= 1 && dia <= 31 && mes >= 1 && mes <= 12 && ano >= 1900 && ano <= 2200) {
68       return new Date(Date.UTC(ano, mes - 1, dia));
69     }
70   }
71   const parsed = new Date(s);
72   if (!isNaN(parsed.getTime())) {
73     const ano = parsed.getUTCFullYear();
74     if (ano >= 1900 && ano <= 2200) {
75       return new Date(Date.UTC(ano, parsed.getUTCMonth(), parsed.getUTCDate()));
76     }
77   }
78   return null;
79 }

```

Listagem A.2: Office Script para leitura do ficheiro de origem.

A.2.2 Escrita no ficheiro de destino (Escrever Leituras)

```

1  function main(
2      workbook: ExcelScript.Workbook,
3      folhaLeituras: string,
4      dadosFonteJSON: string
5  ): string {
6      let fonte: FonteData;
7      try {
8          fonte = JSON.parse(dadosFonteJSON) as FonteData;
9      } catch (e) {
10         return JSON.stringify({ erro: "JSON do fonte invalido: " + e, linhasNovas: 0,
11             linhasCompletadas: 0 });
12     }
13     if (fonte.erro) return JSON.stringify({ erro: "Fonte: " + fonte.erro, linhasNovas: 0,
14         linhasCompletadas: 0 });
15     ;
16     if (!fonte.dados || fonte.dados.length === 0) {
17         return JSON.stringify({ erro: "Fonte sem dados", linhasNovas: 0, linhasCompletadas: 0 });
18     }
19     const ws = workbook.getWorksheet(folhaLeituras);
20     if (!ws) return JSON.stringify({ erro: "Folha não encontrada no destino", linhasNovas: 0,
21         linhasCompletadas: 0 });
22     const range = ws.getUsedRange();
23     if (!range) return JSON.stringify({ erro: "Folha destino vazia", linhasNovas: 0,
24         linhasCompletadas: 0 });
25     const values = range.getValues();
26     let primeiraLinhaDados = encontrarPrimeiraLinhaDados(values);
27     if (primeiraLinhaDados < 0) {
28         primeiraLinhaDados = (fonte.primeiraLinhaDados !== undefined && fonte.primeiraLinhaDados
29             >= 0)
30             ? fonte.primeiraLinhaDados
31             : 2;
32     }
33     // Mapear datas do destino -> índice da linha (so linhas com pelo menos um valor)
34     const destinoIdxPorData: { [k: string]: number } = {};
35     let ultimaDataComValores: Date | null = null;
36     let ultimaLinhaComValores = primeiraLinhaDados - 1;
37     for (let i = primeiraLinhaDados; i < values.length; i++) {
38         const d = parseData(values[i][0]);
39         if (!d) continue;
40         if (linhaTemValores(values[i])) {
41             destinoIdxPorData[chaveData(d)] = i;
42             ultimaDataComValores = d;
43             ultimaLinhaComValores = i;
44         }
45     }
46     const hoje = new Date();
47     hoje.setUTCHours(0, 0, 0, 0);

```



```

44 // Janela para completar linhas existentes: últimos 3 dias antes da última com valores
45 const DIAS_JANELA_COMPLETAR = 3;
46 const dataLimiteCompletar = ultimaDataComValores
47   ? new Date(ultimaDataComValores.getTime() - DIAS_JANELA_COMPLETAR * 86400000)
48   : null;
49 let linhasCompletadas = 0;
50 const linhasParaAdicionar: (string | number | boolean)[][] = [];
51 for (const linhaFonte of fonte.dados) {
52   if (!linhaFonte || linhaFonte.length === 0) continue;
53   const dataFonte = parseData(linhaFonte[0]);
54   if (!dataFonte) continue;
55   if (dataFonte > hoje) continue;
56   if (!linhaTemValores(linhaFonte)) continue;
57   const idxDestino = destinoIdxPorData[chaveData(dataFonte)];
58   if (idxDestino !== undefined) {
59     // Data existe no destino. Só tenta completar se está dentro da janela (últimos N dias)
60     if (dataLimiteCompletar && dataFonte >= dataLimiteCompletar) {
61       const linhaDestino = valores[idxDestino];
62       let completou = false;
63       const cellsParaAtualizar: number[] = [];
64       for (let j = 1; j < linhaFonte.length && j < linhaDestino.length; j++) {
65         const cellDestino = linhaDestino[j];
66         const cellFonte = linhaFonte[j];
67         // Só preenche se destino está VAZIO e fonte tem valor
68         if ((cellDestino === "" || cellDestino === null || cellDestino === undefined)
69           && cellFonte !== "" && cellFonte !== null && cellFonte !== undefined) {
70           linhaDestino[j] = cellFonte;
71           cellsParaAtualizar.push(j);
72           completou = true;
73         }
74       }
75       if (completou) {
76         const rowRange = ws.getRangeByIndexes(idxDestino, 0, 1, linhaDestino.length);
77
78         rowRange.setValues([linhaDestino]);
79         linhasCompletadas++;
80       }
81     }
82     // Se está fora da janela, não toca
83   } else {
84     // Data NAO existe no destino - adicionar se for posterior à última
85     if (ultimaDataComValores === null || dataFonte > ultimaDataComValores) {
86       linhasParaAdicionar.push(linhaFonte);
87     }
88   }
89 }
90 let linhasNovas = 0;
91 if (linhasParaAdicionar.length > 0) {
92   const startRow = ultimaLinhaComValores + 1;
93   const cols = valores[0].length;
94   const paddedRows = linhasParaAdicionar.map(r => {
95     const padded: (string | number | boolean)[] = [];
96     for (let j = 0; j < cols; j++) {
97       padded.push(j < r.length && r[j] !== null && r[j] !== undefined ? r[j] : "");
98     }
99     return padded;
100   });

```

```

101 const novoRange = ws.getRangeByIndexes(startRow, 0, paddedRows.length, cols);
102 novoRange.setValues(paddedRows);
103 const dateRange = ws.getRangeByIndexes(startRow, 0, paddedRows.length, 1);
104 dateRange.setNumberFormat("dd/mm/yyyy");
105 linhasNovas = paddedRows.length;
106 }
107 return JSON.stringify({
108   nome: workbook.getName(),
109   linhasNovas: linhasNovas,
110   linhasCompletadas: linhasCompletadas,
111   erro: ""
112 });
113 }
114 // Helpers
115 function linhaTemValores(linha: (string | number | boolean)[]): boolean {
116   for (let j = 1; j < linha.length; j++) {
117     const cell = linha[j];
118     if (cell !== "" && cell !== null && cell !== undefined) {
119       return true;
120     }
121   }
122   return false;
123 }
124 function encontrarPrimeiraLinhaDados(values: (string | number | boolean)[][]): number {
125   for (let i = 0; i < values.length; i++) {
126     if (parseData(values[i][0]) !== null) return i;
127   }
128   return -1;
129 }
130 function chaveData(d: Date): string {
131   return d.getUTCFullYear() + "-" + (d.getUTCMonth() + 1) + "-" + d.getUTCDate();
132 }
133 function parseData(v: string | number | boolean): Date | null {
134   if (v === null || v === undefined) return null;
135   if (typeof v === "number") {
136     if (v > 36000 && v < 80000) {
137       return new Date(Date.UTC(1899, 11, 30) + Math.floor(v) * 86400000);
138     }
139     return null;
140   }
141   if (typeof v !== "string") return null;
142   const s = v.trim();
143   if (!s) return null;
144   let m = s.match(/^(\d{1,2})[\/\-\.](\d{1,2})[\/\-\.](\d{4})(?:[sT].*)?$/);
145   if (m) {
146     const dia = parseInt(m[1]);
147     const mes = parseInt(m[2]);
148     const ano = parseInt(m[3]);
149     if (dia >= 1 && dia <= 31 && mes >= 1 && mes <= 12 && ano >= 1900 && ano <= 2200) {
150
151       return new Date(Date.UTC(ano, mes - 1, dia));
152     }
153   }
154   m = s.match(/^(\d{4})[\/\-\.](\d{1,2})[\/\-\.](\d{1,2})(?:[sT].*)?$/);
155   if (m) {
156     const ano = parseInt(m[1]);
157     const mes = parseInt(m[2]);

```

```
158     const dia = parseInt(m[3]);
159     if (dia >= 1 && dia <= 31 && mes >= 1 && mes <= 12 && ano >= 1900 && ano <= 2200) {
160         return new Date(Date.UTC(ano, mes - 1, dia));
161     }
162 }
163 const parsed = new Date(s);
164 if (!isNaN(parsed.getTime())) {
165     const ano = parsed.getUTCFullYear();
166     if (ano >= 1900 && ano <= 2200) {
167         return new Date(Date.UTC(ano, parsed.getUTCMonth(), parsed.getUTCDate()));
168     }
169 }
170 return null;
171 }
172 interface FonteData {
173     nome?: string;
174     primeiraLinhaDados?: number;
175     cabecalhos?: (string | number | boolean)[] [];
176     dados: (string | number | boolean)[] [];
177     erro?: string;
178 }
```

Listagem A.3: Office Script para escrita no ficheiro de destino.

Anexo B

Sistema de Alertas por Correio Eletrónico

Este anexo reúne o código desenvolvido para a alarmística diária descrita na Secção 3.3.3. Tal como na importação de dados, a solução evoluiu em duas fases que executam o mesmo trabalho: uma primeira implementação em Python (Anexo B.1), validada localmente, e a versão final em Office Scripts orquestrados pelo Power Automate (Anexo B.2), que eliminou a dependência de servidor dedicado e as restrições de acesso programático ao Outlook.

B.1 Fase 1 — Script Python

O código seguinte corresponde ao script Python desenvolvido para execução automática num servidor da empresa. O script lê diariamente os ficheiros Excel de utilidades sincronizados via OneDrive/SharePoint, analisa os valores registados no dia anterior, identifica situações de alerta e envia um relatório por correio eletrónico através do Outlook.

```
1      # =====
2      # SCRIPT: enviar_alertas_utilidades.py
3      # DESCRIÇÃO: Lê os ficheiros Excel de Água, Ar Comprimido/CO2
4      #             e Energia diretamente do SharePoint (via OneDrive),
5      #             deteta alertas e envia um email diário via Outlook.
6      #
7      # CORREÇÕES:
8      # 1) Ignora contadores de "Stock" em qualquer utilidade
9      # 2) Água Captada: alerta só se diferença entre dias > limite
10     # 3) Limites indexados por utilidade+contador (evita confusão
11     #     entre coletores com contadores de mesmo nome)
12     #
13     # REQUISITOS:
14     # pip install pyxlsb pywin32
15     # =====
16     import sys
17     import json
```

```

18     from datetime import date, datetime, timedelta
19     from pathlib import Path
20     # =====
21     # CONFIGURAÇÃO
22     # =====
23     EMAIL_DESTINATARIO = "tomas.sa.pt@gmail.com"
24     EMAIL_CC = ""
25
26     EMAIL_ASSUNTO_BASE = "Relatório Diário de Utilidades"
27     # Ficheiro de log para evitar envios duplicados
28     FICHEIRO_LOG = Path(__file__).parent / "log_envios_utilidades.json"
29     # Pasta base do SharePoint sincronizado via OneDrive
30     PASTA_BASE = r"C:\Users\Tomas\Super Bock Group\Energia e Utilidades - Contagens Diárias"
31     FICHEIROS = [
32     {
33         "nome": "Água",
34         "caminho": rf"{PASTA_BASE}\Água.xlsx",
35         "folha_consumos": "Consumos",
36         "folha_leituras": "Leituras",
37         "folha_limites": "Limites",
38         "linha_limite": 2,
39         "linha_verao": None,
40         "linha_inverno": None,
41         "utils_desvio": ["Água Captada"],
42     },
43     {
44         "nome": "Ar Comprimido e CO2",
45         "caminho": rf"{PASTA_BASE}\Ar Comprimido e CO2.xlsx",
46         "folha_consumos": "Consumos",
47         "folha_leituras": "Leituras",
48         "folha_limites": "Limites",
49         "linha_limite": 2,
50         "linha_verao": None,
51         "linha_inverno": None,
52         "utils_desvio": [],
53     },
54     {
55         "nome": "Energia",
56         "caminho": rf"{PASTA_BASE}\Consumo e Produção Energia.xlsx",
57         "folha_consumos": "Consumos",
58         "folha_leituras": "Leituras",
59         "folha_limites": "Sheet1",
60         "linha_limite": 2,
61         "linha_verao": None,
62         "linha_inverno": None,
63         "utils_desvio": [],
64     },
65 ]
66 LINHA_CABECALHO_UTIL = 0
67 LINHA_CABECALHO_CONT = 1
68 PRIMEIRA_LINHA_DADOS = 2
69 def carregar_folha(caminho, nome_folha):
70     ext = Path(caminho).suffix.lower()
71     if ext == ".xlsx":
72         from pyxlsb import open_workbook
73         with open_workbook(caminho) as wb:
74             with wb.get_sheet(nome_folha) as sheet:

```

```

75         return [[c.v for c in row] for row in sheet.rows()]
76     else:
77         from openpyxl import load_workbook
78         wb = load_workbook(caminho, data_only=True, read_only=True)
79         ws = wb[nome_folha]
80         dados = [[c.value for c in row] for row in ws.iter_rows()]
81         wb.close()
82         return dados
83     def serial_para_date(valor):
84         if valor is None:
85             return None
86         try:
87             if isinstance(valor, (int, float)):
88                 return date(1899, 12, 30) + timedelta(days=int(valor))
89             if isinstance(valor, datetime):
90                 return valor.date()
91             if isinstance(valor, date):
92                 return valor
93         except Exception:
94             return None
95     return None
96
97     def encontrar_linha_por_data(dados, data_alvo):
98         ultima_valida = None
99         for i in range(PRIMEIRA_LINHA_DADOS, len(dados)):
100             d = serial_para_date(dados[i][0])
101             if d is not None:
102                 if d == data_alvo:
103                     return i
104             ultima_valida = i
105         return ultima_valida
106     def obter_nome_utilidade(cabecalho_util, col):
107         """Encontra o nome da utilidade para uma coluna (vai para a esquerda se vazio)."""
108         for c in range(col, -1, -1):
109             val = cabecalho_util[c] if c < len(cabecalho_util) else None
110             if val and str(val).strip() not in ("", "0", "None"):
111                 return str(val).strip()
112         return "Desconhecida"
113     def is_stock(nome_utilidade):
114         """Verifica se é uma utilidade de stock (a ignorar)."""
115         return "stock" in nome_utilidade.lower()
116     def construir_mapa_limites(dados_limites, cfg):
117         mapa = {}
118         if len(dados_limites) <= cfg["linha_limite"]:
119             return mapa
120         cab_util = dados_limites[LINHA_CABECALHO_UTIL]
121         cab_cont = dados_limites[LINHA_CABECALHO_CONT]
122         util_atual = None
123         for col in range(len(cab_cont)):
124             u = cab_util[col] if col < len(cab_util) else None
125             if u and str(u).strip() not in ("", "None"):
126                 util_atual = str(u).strip()
127             nome_cont = cab_cont[col] if col < len(cab_cont) else None
128             if not nome_cont or str(nome_cont).strip() in ("", "None"):
129                 continue
130             nome_cont = str(nome_cont).strip()
131             def val_num(linha, c=col):

```

```

132     if linha is None or linha >= len(dados_limites):
133         return None
134     try:
135         v = dados_limites[linha][c] if c < len(dados_limites[linha]) else None
136         return float(v) if v is not None and str(v).strip() not in ("-", "", "None", "P.T", "
AVARIA") else None
137     except (ValueError, TypeError):
138         return None
139     chave = (util_atual or "", nome_cont)
140     mapa[chave] = {
141         "limite": val_num(cfg["linha_limite"]),
142         "verao": val_num(cfg["linha_verao"]),
143         "inverno": val_num(cfg["linha_inverno"]),
144     }
145     return mapa
146 def is_verao(d):
147     return 5 <= d.month <= 8
148 def is_val(v):
149     if v is None:
150         return False
151     if str(v).strip() in ("", "None", "0x17"):
152         return False
153     try:
154         float(v)
155         return True
156     except (ValueError, TypeError):
157         return False
158
159 def analisar_ficheiro(cfg, data_analisada):
160     caminho = cfg["caminho"]
161     if not Path(caminho).exists():
162         print(f" AVISO: Ficheiro não encontrado: {caminho}")
163         return [], []
164     dados_consumos = carregar_folha(caminho, cfg["folha_consumos"])
165     dados_leituras = carregar_folha(caminho, cfg["folha_leituras"])
166     dados_limites = carregar_folha(caminho, cfg["folha_limites"])
167     data_ontem = data_analisada - timedelta(days=1)
168     linha_consumos_hoje = encontrar_linha_por_data(dados_consumos, data_analisada)
169     linha_consumos_ontem = encontrar_linha_por_data(dados_consumos, data_ontem)
170     linha_leituras_hoje = encontrar_linha_por_data(dados_leituras, data_analisada)
171     linha_leituras_ontem = encontrar_linha_por_data(dados_leituras, data_ontem)
172     if linha_consumos_hoje is None:
173         print(f" AVISO: Sem dados para {data_analisada} em {cfg['nome']}")
174         return [], []
175     cab_util = dados_consumos[LINHA_CABECALHO_UTIL]
176     cab_cont = dados_consumos[LINHA_CABECALHO_CONT]
177     mapa_limites = construir_mapa_limites(dados_limites, cfg)
178     verao = is_verao(data_analisada)
179     utils_desvio = cfg.get("utils_desvio", [])
180     alertas_v = []
181     alertas_a = []
182     for col in range(1, len(cab_cont)):
183         nome_cont = cab_cont[col] if col < len(cab_cont) else None
184         if not nome_cont or str(nome_cont).strip() in ("", "0", "None"):
185             continue
186         nome_cont = str(nome_cont).strip()
187         nome_util = obter_nome_utilidade(cab_util, col)

```



```

188     if is_stock(nome_util):
189         continue
190     if linha_leituras_hoje is not None and linha_leituras_ontem is not None:
191         try:
192             lh = dados_leituras[linha_leituras_hoje][col] if col < len(dados_leituras[
193                 linha_leituras_hoje]) else None
194             lo = dados_leituras[linha_leituras_ontem][col] if col < len(dados_leituras[
195                 linha_leituras_ontem]) else None
196             if is_val(lh) and is_val(lo):
197                 if float(lh) < float(lo):
198                     alertas_v.append({
199                         "utilidade": nome_util,
200                         "contador": nome_cont,
201                         "leitura_hoje": float(lh),
202                         "leitura_ontem": float(lo),
203                     })
204         except (ValueError, TypeError, IndexError):
205             pass
206     try:
207         consumo_hoje = dados_consumos[linha_consumos_hoje][col]
208         if not is_val(consumo_hoje):
209             continue
210         consumo_hoje = float(consumo_hoje)
211     except (ValueError, TypeError, IndexError):
212         continue
213     info = mapa_limite.get((nome_util, nome_cont))
214     if not info:
215         for (u, c), v in mapa_limite.items():
216             if c == nome_cont:
217                 info = v
218                 break
219     if not info:
220         continue
221     if verao and info["verao"] is not None:
222         limite = info["verao"]
223     elif not verao and info["inverno"] is not None:
224         limite = info["inverno"]
225     else:
226         limite = info["limite"]
227
228     if limite is None:
229         continue
230     if nome_util in utils_desvio:
231         if linha_consumos_ontem is None:
232             continue
233         try:
234             consumo_ontem = dados_consumos[linha_consumos_ontem][col]
235             if not is_val(consumo_ontem):
236                 continue
237             diferenca = float(consumo_ontem) - consumo_hoje
238             if diferenca <= 0:
239                 continue
240             valor_comparar = diferenca
241         except (ValueError, TypeError, IndexError):
242             continue
243     else:
244         valor_comparar = consumo_hoje

```

```

243     if valor_comparar > limite:
244         if nome_util in utils_desvio:
245             consumo_ontem_val = float(dados_consumos[linha_consumos_ontem][col])
246             diferenca = consumo_hoje - consumo_ontem_val
247             desvio_pct = (abs(diferenca) / abs(consumo_ontem_val) * 100) if consumo_ontem_val !=
0 else None
248         else:
249             consumo_ontem_val = None
250             diferenca = None
251             desvio_pct = ((valor_comparar - limite) / limite * 100) if limite > 0 else None
252         alertas_a.append({
253             "utilidade": nome_util,
254             "contador": nome_cont,
255             "consumo": consumo_hoje,
256             "consumo_ontem": consumo_ontem_val,
257             "diferenca": diferenca,
258             "limite": limite,
259             "desvio_pct": desvio_pct,
260             "is_desvio": nome_util in utils_desvio,
261         })
262     return alertas_v, alertas_a
263 def construir_html(data_email, resultados, verao):
264     periodo = "Verão (mai-ago)" if verao else "Resto do Ano"
265     def fmt(n):
266         return f"{n:,.0f}".replace(",", " ")
267     total_v = sum(len(r["alertas_v"]) for r in resultados)
268     total_a = sum(len(r["alertas_a"]) for r in resultados)
269     html = f"""
270 <html><body style="font-family:Calibri,Arial,sans-serif;font-size:14px;color:#222;">
271 <h2 style="color:#1a3c6e;border-bottom:2px solid #1a3c6e;padding-bottom:6px;">
272     &#128200; Relatório Diário de Utilidades
273 </h2>
274 <p><b>Data analisada:</b> {data_email.strftime('%d/%m/%Y')}
275     &nbsp;&nbsp;&nbsp;<b>Período:</b> {periodo}
276     &nbsp;&nbsp;&nbsp;<b>Alertas vermelhos:</b> {total_v}
277     &nbsp;&nbsp;&nbsp;<b>Alertas amarelos:</b> {total_a}
278 </p>
279 """
280     for r in resultados:
281         nome = r["nome"]
282         alertas_v = r["alertas_v"]
283         alertas_a = r["alertas_a"]
284         nv = len(alertas_v)
285         na = len(alertas_a)
286         html += f"""
287 <h2 style="color:#1a3c6e;margin-top:30px;border-bottom:1px solid #ccc;padding-bottom:4px;
">
288     {nome}
289 </h2>"""
290         html += f'<h3 style="color:#c0392b;">&#128308; Leitura Inferior ao Dia Anterior ({nv})</h3>'
291         if nv == 0:
292             html += '<p style="color:#27ae60;">&#10003; Nenhuma anomalia detetada.</p>'
293         else:
294
295         html += """
296 <table style="border-collapse:collapse;width:100%;margin-bottom:10px;">

```

```

297 <tr style="background:#c0392b;color:white;">
298 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Utilidade</th>
299 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Contador</th>
300 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Leitura Hoje</th>
301 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Leitura Ontem</th>
302 </tr>"""
303 for i, a in enumerate(alertas_v):
304     bg = "#fdecea" if i % 2 == 0 else "#ffffff"
305     html += f"""
306 <tr style="background:{bg};">
307 <td style="padding:7px;border:1px solid #ddd;">{a['utilidade']}</td>
308 <td style="padding:7px;border:1px solid #ddd;">{a['contador']}</td>
309 <td style="padding:7px;border:1px solid #ddd;text-align:right;color:#c0392b;font-weight:
bold;">{fmt(a['leitura_hoje'])}</td>
310 <td style="padding:7px;border:1px solid #ddd;text-align:right;">{fmt(a['leitura_ontem'])}
</td>
311 </tr>"""
312 html += "</table>"
313 html += f'<h3 style="color:#d68910;">&#128993; Valor Fora dos Limites Definidos ({na})</h3>'
314 if na == 0:
315     html += '<p style="color:#27ae60;">&#10003; Nenhum valor fora dos limites.</p>'
316 else:
317     alertas_desvio = [a for a in alertas_a if a.get("is_desvio")]
318     alertas_normais = [a for a in alertas_a if not a.get("is_desvio")]
319     if alertas_normais:
320         html += """
321 <table style="border-collapse:collapse;width:100%;margin-bottom:10px;">
322 <tr style="background:#d68910;color:white;">
323 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Utilidade</th>
324 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Contador</th>
325 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Consumo</th>
326 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Limite</th>
327 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Desvio</th>
328 </tr>"""
329     for i, a in enumerate(alertas_normais):
330         bg = "#fef9e7" if i % 2 == 0 else "#ffffff"
331         desvio_str = f"+{a['desvio_pct']:.1f}%" if a['desvio_pct'] is not None else "n/a"
332         html += f"""
333 <tr style="background:{bg};">
334 <td style="padding:7px;border:1px solid #ddd;">{a['utilidade']}</td>
335 <td style="padding:7px;border:1px solid #ddd;">{a['contador']}</td>
336 <td style="padding:7px;border:1px solid #ddd;text-align:right;color:#d68910;font-
weight:bold;">{fmt(a['consumo'])}</td>
337 <td style="padding:7px;border:1px solid #ddd;text-align:right;">{fmt(a['limite'])}</td>
338 <td style="padding:7px;border:1px solid #ddd;text-align:right;color:#c0392b;font-
weight:bold;">{desvio_str}</td>
339 </tr>"""
340     html += "</table>"
341     if alertas_desvio:
342         html += """
343 <table style="border-collapse:collapse;width:100%;margin-bottom:10px;">
344 <tr style="background:#d68910;color:white;">
345 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Utilidade</th>
346 <th style="padding:8px;text-align:left;border:1px solid #aaa;">Contador</th>
347 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Dia Anterior</th>
348 <th style="padding:8px;text-align:right;border:1px solid #aaa;">Valor Hoje</th>

```

```

349     <th style="padding:8px;text-align:right;border:1px solid #aaa;">Diferença</th>
350     <th style="padding:8px;text-align:right;border:1px solid #aaa;">Limite</th>
351     <th style="padding:8px;text-align:right;border:1px solid #aaa;">Desvio</th>
352 </tr>"""
353 for i, a in enumerate(alertas_desvio):
354     bg = "#fef9e7" if i % 2 == 0 else "#ffffff"
355     col_ontem = fmt(a['consumo_ontem']) if a['consumo_ontem'] is not None else "-"
356     col_diferenca = fmt(a['diferenca']) if a['diferenca'] is not None else "-"
357     desvio_str = f"{a['desvio_pct']:.1f}%" if a['desvio_pct'] is not None else "n/a"
358     html += f"""
359     <tr style="background:{bg};">
360         <td style="padding:7px;border:1px solid #ddd;">{a['utilidade']}

```

```

396     return False
397 def registrar_envio():
398     with open(FICHEIRO_LOG, "w") as f:
399         json.dump({"ultimo_envio": date.today().isoformat()}, f)
400 def enviar_email_outlook(destinatario, cc, assunto, corpo_html):
401     import win32com.client
402     outlook = win32com.client.Dispatch("Outlook.Application")
403     mail = outlook.CreateItem(0)
404     mail.To = destinatario
405     if cc:
406         mail.CC = cc
407     mail.Subject = assunto
408     mail.HTMLBody = corpo_html
409     mail.Send()
410 def main():
411     print(f"[{datetime.now().strftime('%H:%M:%S')}] A iniciar envio de alertas de Utilidades...")
412     if ja_enviado_hoje():
413         print("Email já enviado hoje. A terminar.")
414         sys.exit(0)
415     data_hoje = date.today() - timedelta(days=1)
416     resultados = []
417     for cfg in FICHEIROS:
418         print(f"\nA processar: {cfg['nome']}...")
419         alertas_v, alertas_a = analisar_ficheiro(cfg, data_hoje)
420         print(f" Alertas vermelhos: {len(alertas_v)}")
421         print(f" Alertas amarelos: {len(alertas_a)}")
422         resultados.append({
423             "nome": cfg["nome"],
424             "alertas_v": alertas_v,
425             "alertas_a": alertas_a,
426         })
427     corpo = construir_html(data_hoje, resultados, is_verao(data_hoje))
428     assunto = f"{EMAIL_ASSUNTO_BASE} -- {data_hoje.strftime('%d/%m/%Y')}"
429     print(f"\nA enviar email para {EMAIL_DESTINATARIO}...")
430
431     try:
432         enviar_email_outlook(EMAIL_DESTINATARIO, EMAIL_CC, assunto, corpo)
433         registrar_envio()
434         print("Email enviado com sucesso!")
435     except Exception as e:
436         print(f"ERRO ao enviar email: {e}")
437         sys.exit(1)
438 if __name__ == "__main__":
439     main()

```

Listagem B.1: Script Python para envio automático de alertas por correio eletrónico.

B.2 Fase 2 — Office Scripts (Power Automate)

A versão final separa a lógica de análise da lógica de apresentação: três scripts de análise (`RunAgua`, `RunArCO2` e `RunEnergia`), que partilham o mesmo código-base, devolvem ao fluxo uma estrutura JSON com os alertas detetados; um quarto script converte essa

estrutura no corpo HTML do correio eletrónico. Esta separação permite alterar o formato do relatório sem tocar nos scripts de deteção.

B.2.1 Análise de anomalias por utilidade

```

1 function main(
2   workbook: ExcelScript.Workbook,
3   folhaConsumos: string,
4   folhaLeituras: string,
5   folhaLimites: string,
6   linhaLimite: number,
7   utilsDesvio: string,
8   dataAnalisada: string,
9   linhaVerao: number,
10  linhaInverno: number,
11  utilsSazonais: string
12 ): AnaliseResultado {
13   const utilsDesvioList = utilsDesvio
14     .split(",")
15     .map(s => s.trim())
16     .filter(s => s.length > 0);
17   const utilsSazonaisList = utilsSazonais
18     .split(",")
19     .map(s => s.trim())
20     .filter(s => s.length > 0);
21   const dataAlvo = parseDataFlex(dataAnalisada);
22   if (dataAlvo === null) {
23     return {
24       nome: workbook.getName(),
25       alertas_v: [],
26       alertas_a: [],
27       aviso: "Formato de data invalido: '" + dataAnalisada
28         + "'. Use YYYY-MM-DD ou DD/MM/YYYY."
29     };
30   }
31   const dataOntem = new Date(dataAlvo.getTime() - 86400000);
32
33   const dadosConsumos = lerFolha(workbook, folhaConsumos);
34   const dadosLeituras = lerFolha(workbook, folhaLeituras);
35   const dadosLimites = lerFolha(workbook, folhaLimites);
36   if (!dadosConsumos || !dadosLeituras || !dadosLimites) {
37     return {
38       nome: workbook.getName(),
39       alertas_v: [],
40       alertas_a: [],
41       aviso: "Folha não encontrada"
42     };
43   }
44   const linhaConsumosHoje = encontrarLinhaPorData(dadosConsumos, dataAlvo);
45   ;
46   const linhaConsumosOntem = encontrarLinhaPorData(dadosConsumos, dataOntem);
47   ;
48   const linhaLeiturasHoje = encontrarLinhaPorData(dadosLeituras, dataAlvo);
49   ;
50   const linhaLeiturasOntem = encontrarLinhaPorData(dadosLeituras, dataOntem);
51   ;
52   if (linhaConsumosHoje === -1) {
53     return {
54       nome: workbook.getName(),
55       alertas_v: [],
56       alertas_a: [],
57       aviso: "Sem dados para a data"
58     };
59   }
60 }

```

```

50     const cabUtil = dadosConsumos[0];
51     const cabCont = dadosConsumos[1];
52     const mapaLimites = construirMapaLimites(
53         dadosLimites , linhaLimite , linhaVerao , linhaInverno
54     );
55     const verao = isVerao(dataAlvo);
56     const alertasV: AlertaVermelho[] = [];
57     const alertasA: AlertaAmarelo[] = [];
58     for (let col = 1; col < cabCont.length; col++) {
59         const nomeContRaw = cabCont[col];
60         if (nomeContRaw === null || nomeContRaw === undefined) continue;
61         const nomeCont = String(nomeContRaw).trim();
62         if (nomeCont === "" || nomeCont === "0" || nomeCont === "None")
63             continue;
64         const nomeUtil = obterNomeUtilidade(cabUtil, col);
65         if (nomeUtil.toLowerCase().indexOf("stock") >= 0) continue;
66         // Alerta vermelho: leitura de hoje < leitura de ontem
67         if (linhaLeiturasHoje !== -1 && linhaLeiturasOntem !== -1) {
68
69             const lh = numOrNull(getCell(dadosLeituras , linhaLeiturasHoje , col))
70             ;
71             const lo = numOrNull(getCell(dadosLeituras , linhaLeiturasOntem , col))
72             ;
73             if (lh !== null && lo !== null && lh < lo) {
74                 alertasV.push({
75                     utilidade: nomeUtil, contador: nomeCont,
76                     leitura_hoje: lh, leitura_ontem: lo
77                 });
78             }
79         }
80         const consumoHoje = numOrNull(getCell(dadosConsumos , linhaConsumosHoje ,
81             col));
82         if (consumoHoje === null) continue;
83         // Obter limite (com fallback por nome de contador)
84         let info = mapaLimites[chave(nomeUtil, nomeCont)];
85         if (!info) {
86             for (const k of Object.keys(mapaLimites)) {
87                 if (k.endsWith("||" + nomeCont)) { info = mapaLimites[k]; break; }
88             }
89         }
90         if (!info) continue;
91         // Seleção de limite sazonal
92         let limite: number | null = info.limite;
93         const isSazonal = utilsSazonaisList.indexOf(nomeUtil) >= 0;
94         if (isSazonal) {
95             if (verao && info.verao !== null) limite = info.verao;
96             if (!verao && info.inverno !== null) limite = info.inverno;
97         }
98         if (limite === null) continue;
99         // Alerta amarelo: consumo acima do limite (absoluto ou por desvio)
100         const isDesvio = utilsDesvioList.indexOf(nomeUtil) >= 0;
101         let valorComparar: number;
102         let consumoOntemVal: number | null = null;
103         let diferenca: number | null = null;
104         if (isDesvio) {
105             if (linhaConsumosOntem === -1) continue;
106             const co = numOrNull(getCell(dadosConsumos , linhaConsumosOntem , col))

```

```

107     ;
108     if (co === null) continue;
109     const dif = co - consumoHoje;
110     if (dif <= 0) continue;
111     valorComparar = dif;
112     consumoOntemVal = co;
113
114     diferenca = consumoHoje - co;
115   } else {
116     valorComparar = consumoHoje;
117   }
118   if (valorComparar > limite) {
119     let desvioPct: number | null = null;
120     if (isDesvio && consumoOntemVal !== null && consumoOntemVal !== 0) {
121       desvioPct = Math.abs(diferenca!) / Math.abs(consumoOntemVal) * 100;
122     } else if (!isDesvio && limite > 0) {
123       desvioPct = (valorComparar - limite) / limite * 100;
124     }
125     alertasA.push({
126       utilidade: nomeUtil, contador: nomeCont,
127       consumo: consumoHoje, consumo_ontem: consumoOntemVal,
128       diferenca: diferenca, limite: limite,
129       desvio_pct: desvioPct, is_desvio: isDesvio
130     });
131   }
132 }
133 return {
134   nome: workbook.getName(),
135   alertas_v: alertasV, alertas_a: alertasA, aviso: ""
136 };
137 }
138 // -- Helpers
139 -----
140 function parseDataFlex(s: string): Date | null {
141   if (!s) return null;
142   const t = s.trim();
143   let m = t.match(/^(\\d{4})-(\\d{1,2})-(\\d{1,2})$/);
144   if (m) return new Date(
145     Date.UTC(parseInt(m[1]), parseInt(m[2]) - 1, parseInt(m[3]))
146   );
147   m = t.match(/^(\\d{1,2})[\\/-](\\d{1,2})[\\/-](\\d{4})$/);
148   if (m) return new Date(
149     Date.UTC(parseInt(m[3]), parseInt(m[2]) - 1, parseInt(m[1]))
150   );
151   return null;
152 }
153 function lerFolha(
154   wb: ExcelScript.Workbook, nome: string
155 ): (string | number | boolean)[][] | null {
156   const ws = wb.getWorksheet(nome);
157   if (!ws) return null;
158
159   const r = ws.getUsedRange();
160   return r ? r.getValues() as (string | number | boolean)[][] : [];
161 }
162 function getCell(
163   dados: (string | number | boolean)[][], lin: number, col: number

```



```

164 ) {
165   if (lin < 0 || lin >= dados.length) return null;
166   const linha = dados[lin];
167   return col < linha.length ? linha[col] : null;
168 }
169 function isVerao(d: Date): boolean {
170   const m = d.getUTCMonth() + 1;
171   return m >= 5 && m <= 8;
172 }
173 function chave(util: string, cont: string): string {
174   return util + "||" + cont;
175 }
176 function numOrNull(
177   v: string | number | boolean | null | undefined
178 ): number | null {
179   if (v === null || v === undefined) return null;
180   const s = String(v).trim();
181   if (["", "None", "-", "P.T", "AVARIA", "0x17"].includes(s)) return null;
182   const n = Number(s);
183   return isNaN(n) ? null : n;
184 }
185 function serialParaDate(v: string | number | boolean): Date | null {
186   if (v === null || v === undefined) return null;
187   if (typeof v === "number")
188     return new Date(Date.UTC(1899, 11, 30) + v * 86400000);
189   if (typeof v === "string") {
190     const d = new Date(v);
191     if (!isNaN(d.getTime())) return d;
192   }
193   return null;
194 }
195 function mesmaData(a: Date, b: Date): boolean {
196   return a.getUTCFullYear() === b.getUTCFullYear()
197     && a.getUTCMonth() === b.getUTCMonth()
198     && a.getUTCDate() === b.getUTCDate();
199 }
200 function encontrarLinhaPorData(
201   dados: (string | number | boolean)[][], alvo: Date
202 ): number {
203   const alvoSerial =
204     (alvo.getTime() - Date.UTC(1899, 11, 30)) / 86400000;
205   let ultima = -1;
206   for (let i = 2; i < dados.length; i++) {
207     const v = dados[i][0];
208     if (v === null || v === undefined) continue;
209     if (typeof v === "number") {
210       if (Math.abs(v - alvoSerial) < 0.5) return i;
211       ultima = i;
212     } else {
213       const d = serialParaDate(v);
214       if (d !== null) {
215         if (mesmaData(d, alvo)) return i;
216         ultima = i;
217       }
218     }
219   }
220   return ultima;

```

```

221     return ultima;
222 }
223 function obterNomeUtilidade(
224     cabUtil: (string | number | boolean)[], col: number
225 ): string {
226     for (let c = col; c >= 0; c--) {
227         const v = c < cabUtil.length ? cabUtil[c] : null;
228         if (v !== null && v !== undefined) {
229             const s = String(v).trim();
230             if (s !== "" && s !== "0" && s !== "None") return s;
231         }
232     }
233     return "Desconhecida";
234 }
235 function construirMapaLimites(
236     dadosLim: (string | number | boolean)[][],
237     linhaLim: number, linhaVer: number, linhaInv: number
238 ): { [k: string]: LimiteInfo } {
239     const mapa: { [k: string]: LimiteInfo } = {};
240     if (dadosLim.length <= linhaLim) return mapa;
241     const cabUtil = dadosLim[0];
242     const cabCont = dadosLim[1];
243     let utilAtual = "";
244     const usarVerao = linhaVer >= 0 && linhaVer < dadosLim.length;
245     const usarInverno = linhaInv >= 0 && linhaInv < dadosLim.length;
246     for (let col = 0; col < cabCont.length; col++) {
247         const u = col < cabUtil.length ? cabUtil[col] : null;
248         if (u !== null && u !== undefined) {
249
250             const us = String(u).trim();
251             if (us !== "" && us !== "None") utilAtual = us;
252         }
253         const nc = col < cabCont.length ? cabCont[col] : null;
254         if (nc === null || nc === undefined) continue;
255         const nomeCont = String(nc).trim();
256         if (nomeCont === "" || nomeCont === "None") continue;
257         mapa[chave(utilAtual, nomeCont)] = {
258             limite: numOrNull(getCell(dadosLim, linhaLim, col)),
259             verao: usarVerao ? numOrNull(getCell(dadosLim, linhaVer, col)) :
260                 null,
261             inverno: usarInverno ? numOrNull(getCell(dadosLim, linhaInv, col)) :
262                 null
263         };
264     }
265     return mapa;
266 }
267 // -- Tipos
268 -----
269 interface AlertaVermelho {
270     utilidade: string; contador: string;
271     leitura_hoje: number; leitura_ontem: number;
272 }
273 interface AlertaAmarelo {
274     utilidade: string; contador: string;
275     consumo: number; consumo_ontem: number | null;
276     diferenca: number | null; limite: number | null;
277     desvio_pct: number | null; is_desvio: boolean;

```

```

278 }
279 interface LimiteInfo {
280     limite: number | null; verao: number | null; inverno: number | null;
281 }
282 interface AnaliseResultado {
283     nome: string; alertas_v: AlertaVermelho[];
284     alertas_a: AlertaAmarelo[]; aviso: string;
285 }

```

Listagem B.2: Office Script para análise de anomalias por utilidade (RunAgua / RunArCO2 / RunEnergia).

B.2.2 Geração do relatório HTML

```

1      function main(
2          workbook: ExcelScript.Workbook,
3          resultadoAgua: string,
4          resultadoArCO2: string,
5          resultadoEnergia: string,
6          dataAnalizada: string
7      ): string {
8          const rAgua = JSON.parse(resultadoAgua) as Resultado;
9          const rArCO2 = JSON.parse(resultadoArCO2) as Resultado;
10         const rEnergia = JSON.parse(resultadoEnergia) as Resultado;
11         const blocos: Bloco[] = [
12             { nome: "Agua", res: rAgua },
13             { nome: "Ar Comprimido e CO2", res: rArCO2 },
14             { nome: "Energia", res: rEnergia }
15         ];
16         const data = parseDataFlex(dataAnalizada);
17         if (data === null) {
18             return "<p>Formato de data invalido: " + dataAnalizada + "</p>";
19         }
20         const verao = (data.getUTCMonth() + 1) >= 5
21             && (data.getUTCMonth() + 1) <= 8;
22         const periodo = verao ? "Verao (mai-ago)" : "Resto do Ano";
23         const dataFmt = formatarData(data);
24         let totalV = 0, totalA = 0;
25         for (const b of blocos) {
26             totalV += b.res.alertas_v.length;
27             totalA += b.res.alertas_a.length;
28         }
29         let html = `<html><body style="font-family:Calibri,Arial,sans-serif;"
30             + `font-size:14px;color:#222;">`;
31         html += `<h2 style="color:#1a3c6e;border-bottom:2px solid #1a3c6e;"
32             + `padding-bottom:6px;">Relatorio Diario de Utilidades </h2>`;
33         html += `<p><b>Data analisada:</b> ${dataFmt} &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp; `
34             + `<b>Periodo:</b> ${periodo} &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp; `
35             + `<b>Alertas vermelhos:</b> ${totalV} &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp; `
36             + `<b>Alertas amarelos:</b> ${totalA}</p>`;
37         for (const b of blocos) {
38             html += renderSeccao(b);
39         }

```

```

40 }
41 html += `  
<hr style="border:1px solid #ddd;">`;
42 html += `

Premissas para emissao de alertas:</b><br>`
44   + `Alerta Vermelho -- leitura do dia inferior a leitura do dia `
45   + `anterior (contador a decrementar).<br>`
46   + `Alerta Amarelo (valor absoluto) -- consumo diario superior ao `
47   + ` `
48   + `limite definido na folha de Limites.<br>`
49   + `Alerta Amarelo (desvio) -- aplicavel a Agua Captada: `
50   + `decrescimo `
51   + `entre hoje e ontem superior ao limite.<br>`
52   + `Contadores de Stock sao ignorados. O relatorio analisa os `
53   + `dados `
54   + `do dia anterior ao envio.</p>`;
55 if (totalV === 0 && totalA === 0) {
56   html += `


```

```

97         + `text-align:right;color:#c0392b;font-weight:bold;">`
98         + `${fmt(a.leitura_hoje)}</td>`
99         + `  |
```

```

154         </th>`
155     + `  |
```

```

211     }
212     function fmt(n: number | null): string {
213         if (n === null || n === undefined) return "-";
214         return Math.round(n).toString()
215             .replace(/\B(?=(\d{3})+(?!\d))/g, " ");
216     }
217     function formatarData(d: Date): string {
218         const dd = String(d.getUTCDate()).padStart(2, "0");
219         const mm = String(d.getUTCMonth() + 1).padStart(2, "0");
220         return `${dd}/${mm}/${d.getUTCFullYear()}`;
221     }
222     // -- Tipos
223     -----
224
225     interface AV {
226         utilidade: string; contador: string;
227         leitura_hoje: number; leitura_ontem: number;
228     }
229     interface AA {
230         utilidade: string; contador: string;
231         consumo: number; consumo_ontem: number | null;
232         diferenca: number | null; limite: number | null;
233         desvio_pct: number | null; is_desvio: boolean;
234     }
235     interface Resultado {
236         nome: string; alertas_v: AV[]; alertas_a: AA[]; aviso: string;
237     }
238     interface Bloco { nome: string; res: Resultado; }

```

Listagem B.3: Office Script para geração do relatório HTML (Gerar HTML).

Anexo C

Medidas DAX Implementadas no Power BI

Este anexo apresenta as principais medidas DAX desenvolvidas no Power BI para suportar a análise mensal de KPIs descrita no Capítulo 4, incluindo as medidas de variação percentual, os indicadores acumulados YTD e as comparações com o período homólogo do ano anterior. As medidas encontram-se organizadas pelas três zonas produtivas monitorizadas, seguindo em todas o mesmo padrão: o consumo de cada utilidade é combinado com o respetivo denominador operacional (volume enchido ou produção da área), obtendo-se indicadores normalizados e comparáveis entre períodos.

C.1 Linhas de Enchimento

```
1  /* ----- Ar Comprimido ----- */
2  PercentT_ar = ARCOMPIMIDO[Ano] / ARCOMPIMIDO[Ano-1] - 1
3
4  PercentT_ar_YTD = [Consumo_YTD_Ar] / [Consumo_YTD_Ar_Ano1] - 1
5
6  Consumo_YTD_Ar =
7  VAR UltimaDataSelecionada = MAX('calendario'[DATE])
8  RETURN
9  IF([Volumehl] <= 0, 0,
10     CALCULATE(SUM(ARCOMPIMIDO[Consumo]) / [Volumehl],
11                DATESYTD('calendario'[DATE])))
12
13  Consumo_YTD_Ar_Ano1 =
14  VAR UltimaDataSelecionada = MAX('Calendario'[DATE])
15  RETURN
16  IF([Volumehl] <= 0, 0,
17     CALCULATE(SUM(ARCOMPIMIDO[Consumo]) / [Volumehl],
18                DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE]))))
19
20  /* ----- Água ----- */
21  PercentT_agua_YTD = [INDICADOR_AGUA] / [Indicador_agua_ano-1] - 1
```

```

22
23 PercentT_agua_ano-1 = [Consumo_YTD_AGUA] / [Consumo_YTD_AGUA_Ano1] - 1
24
25 Consumo_YTD_AGUA =
26 CALCULATE(IF(SUM('Links Enchimento'[Volume]) = 0, 0,
27     SUM('AGUA'[Consumo]) * 1000 / SUM('Links Enchimento'[Volume])),
28     DATESYTD('Calendario'[DATE])))
29
30 Consumo_YTD_AGUA_Ano1 =
31 CALCULATE(IF(SUM('Links Enchimento'[Volume]) = 0, 0,
32     SUM('AGUA'[Consumo]) * 1000 / SUM('Links Enchimento'[Volume])),
33     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
34
35 /* ----- CO2 ----- */
36 PercentT_co2_ano = CO2[Co2_esp] / CO2[CO2_Enchimento SPly] - 1
37
38 PercentT_co2_ano-1 = [Consumo_YTD_CO2] / [Consumo_YTD_CO2_Ano1] - 1
39
40 Consumo_YTD_CO2 =
41 CALCULATE(IF(CALCULATE(SUM('Links Enchimento'[Volume]),
42     DATESYTD('calendario'[DATE])) = 0, 0,
43     CALCULATE(SUM(CO2[Consumo]), DATESYTD('calendario'[DATE]))
44     / CALCULATE(SUM('Links Enchimento'[Volume]),
45     DATESYTD('calendario'[DATE])) / 100),
46     DATESYTD('calendario'[DATE])))
47
48 Consumo_YTD_CO2_Ano1 =
49 CALCULATE([Consumo_YTD_CO2], SAMEPERIODLASTYEAR(calendario[DATE]))
50
51 /* ----- Energia Térmica ----- */
52 PercentT_Termica = [ET_LINHAS] / [ET_LINHAS-1] - 1
53
54 PercentT_Termica_YTD = [Consumo_YTD_Vapor] / [Consumo_YTD_Vapor_Ano1] - 1
55
56 Consumo_YTD_Vapor =
57 VAR Consumo_AQUENTE_YTD =
58     CALCULATE(SUM(AQUENTE[Consumo]), DATESYTD('Calendario'[DATE]))
59 VAR Consumo_VAPOR_YTD =
60     CALCULATE(SUM(VAPOR[Consumo]), DATESYTD('Calendario'[DATE]))
61 VAR Volume_YTD =
62     CALCULATE([Volumehl], DATESYTD('Calendario'[DATE]))
63 RETURN
64 IF(Volume_YTD = 0, 0,
65     (Consumo_AQUENTE_YTD + (Consumo_VAPOR_YTD * 668)) / Volume_YTD)
66
67 Consumo_YTD_Vapor_Ano1 =
68 VAR Consumo_AQUENTE_YTD =
69     CALCULATE(SUM(AQUENTE[Consumo]),
70     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
71 VAR Consumo_VAPOR_YTD =
72     CALCULATE(SUM(VAPOR[Consumo]),
73     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
74 VAR Volume_YTD =
75     CALCULATE([Volumehl],
76     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
77 RETURN
78 IF(Volume_YTD = 0, 0,

```

```

79      (Consumo_AQUENTE_YTD + (Consumo_VAPOR_YTD * 668)) / Volume_YTD)
80
81  /* ----- Eletricidade ----- */
82  PercentT_Elet = [Consumo_Esp_Ano] / [Consumo_Esp_SPLY] - 1
83
84  Percent_Elet_YTD = [Consumo_YTD_ELET] / [Consumo_YTD_ELET_Ano1] - 1
85
86  Consumo_YTD_ELET =
87  VAR Consumo_YTD =
88      CALCULATE(SUM(ELEC[Consumo]), DATESYTD('Calendario'[DATE]))
89  VAR Volume_YTD =
90      CALCULATE([Volumehl], DATESYTD('Calendario'[DATE]))
91  RETURN IF(Volume_YTD = 0, 0, Consumo_YTD / Volume_YTD)
92
93  Consumo_YTD_ELET_Ano1 =
94  VAR Consumo_YTD =
95      CALCULATE(SUM(ELEC[Consumo]),
96          DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
97  VAR Volume_YTD =
98      CALCULATE([Volumehl],
99          DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
100 RETURN IF(Volume_YTD = 0, 0, Consumo_YTD / Volume_YTD)

```

Listagem C.1: Medidas DAX das Linhas de Enchimento.

C.2 Adega

As medidas da Adega utilizam a tabela LINKSADEGAS como referência de volume de produção e incluem ainda o CO₂ como indicador de desempenho ambiental.

```

1  /* ----- Água ----- */
2  Percent_Adega_Agua = [Adega_ano] / [Adega_ano-1] - 1
3
4  Percent_Adega_YTD = [Adega_Agua_YTD] / [Adega_Agua_YTD_Ano1] - 1
5
6  Adega_Agua_YTD =
7  VAR Consumo_YTD =
8      CALCULATE(SUM('AGUA'[Consumo]) * 1000, DATESYTD('Calendario'[DATE]))
9  VAR Producao_YTD =
10     CALCULATE(SUM(LINKSADEGAS[Producao]), DATESYTD('Calendario'[DATE]))
11  RETURN IF(Producao_YTD = 0, 0, Consumo_YTD / Producao_YTD)
12
13  Adega_Agua_YTD_Ano1 =
14  VAR Consumo_YTD =
15      CALCULATE(SUM('AGUA'[Consumo]) * 1000,
16          DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
17  VAR Producao_YTD =
18      CALCULATE(SUM(LINKSADEGAS[Producao]),
19          DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
20  RETURN IF(Producao_YTD = 0, 0, Consumo_YTD / Producao_YTD)
21
22  /* ----- Energia Térmica ----- */
23  Percent_Adega_Termica = [ET_ADEGA] / [ET_Adegas_Ano1] - 1

```

```

24
25 Percent_Adega_Termica_YTD = [ET_Adega_YTD] / [ET_Adega_YTD_Ano1] - 1
26
27 ET_Adega_YTD =
28 VAR Vapor_YTD =
29     CALCULATE(SUM(VAPOR[Consumo]) * 668, DATESYTD('Calendario'[DATE]))
30 VAR Aquente_YTD =
31     CALCULATE(SUM(AQUENTE[Consumo]), DATESYTD('Calendario'[DATE]))
32 VAR Producao_YTD =
33     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100, DATESYTD('Calendario'[DATE]))
34 RETURN IF(Producao_YTD = 0, 0, (Vapor_YTD + Aquente_YTD) / Producao_YTD)
35
36 ET_Adega_YTD_Ano1 =
37 VAR Vapor_YTD =
38     CALCULATE(SUM(VAPOR[Consumo]) * 668,
39         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
40 VAR Aquente_YTD =
41     CALCULATE(SUM(AQUENTE[Consumo]),
42         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
43 VAR Producao_YTD =
44     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100,
45         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
46 RETURN IF(Producao_YTD = 0, 0, (Vapor_YTD + Aquente_YTD) / Producao_YTD)
47
48 /* ----- Eletricidade ----- */
49 Percent_Adega_Elect = [ELEC_Adegas] / [Elec_ano-1_Adegas] - 1
50
51 Percent_Adega_Elect_YTD = [ELEC_Adegas_YTD] / [ELEC_Adegas_YTD_Ano1] / -1
52
53 ELEC_Adegas_YTD =
54 VAR Elec_YTD =
55     CALCULATE(SUM(ELEC[Consumo]), DATESYTD('Calendario'[DATE]))
56 VAR Producao_YTD =
57     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100, DATESYTD('Calendario'[DATE]))
58 RETURN IF(Producao_YTD = 0, 0, Elec_YTD / Producao_YTD)
59
60 ELEC_Adegas_YTD_Ano1 =
61 VAR Elec_YTD =
62     CALCULATE(SUM(ELEC[Consumo]),
63         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
64 VAR Producao_YTD =
65     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100,
66         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
67 RETURN IF(Producao_YTD = 0, 0, Elec_YTD / Producao_YTD)
68
69 /* ----- Ar Comprimido ----- */
70 Percent_Adega_Ar = [Adegas] / [Adegas_ano-1] - 1
71
72 Percent_Adega_Ar_YTD = [Adegas_Ar_YTD] / [Adegas_Ar_YTD_Ano1] - 1
73
74 Adegas_Ar_YTD =
75 VAR ArComp_YTD =
76     CALCULATE(SUM(ARCOMPIMIDO[Consumo]), DATESYTD('Calendario'[DATE]))
77 VAR Producao_YTD =
78     CALCULATE(SUM(LINKSADEGAS[Producao]), DATESYTD('Calendario'[DATE]))
79 RETURN IF(Producao_YTD = 0, 0, ArComp_YTD / Producao_YTD)
80

```

```

81 Adegas_Ar_YTD_Ano1 =
82 VAR ArComp_YTD =
83     CALCULATE(SUM(ARCOMPRIADO[Consumo]),
84         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
85 VAR Producao_YTD =
86     CALCULATE(SUM(LINKSADEGAS[Producao]),
87         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
88 RETURN IF(Producao_YTD = 0, 0, ArComp_YTD / Producao_YTD)
89
90 /* ----- CO2 ----- */
91 Percent_Adega_CO2 = CO2[Co2_Adega] / CO2[CO2_ADEGA_-1] - 1
92
93 Percent_Adega_CO2_YTD = CO2[Co2_Adega_YTD] / CO2[Co2_Adega_YTD_Ano1] - 1
94
95 Co2_Adega_YTD =
96 VAR Co2_YTD =
97     CALCULATE(SUM(CO2[Consumo]), DATESYTD('Calendario'[DATE]))
98 VAR Producao_YTD =
99     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100, DATESYTD('Calendario'[DATE]))
100 RETURN IF(Producao_YTD = 0, 0, Co2_YTD / Producao_YTD)
101
102 Co2_Adega_YTD_Ano1 =
103 VAR Co2_YTD =
104     CALCULATE(SUM(CO2[Consumo]),
105         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
106 VAR Producao_YTD =
107     CALCULATE(SUM(LINKSADEGAS[Producao]) / 100,
108         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
109 RETURN IF(Producao_YTD = 0, 0, Co2_YTD / Producao_YTD)

```

Listagem C.2: Medidas DAX da Adega.

C.3 Produção

As medidas da Produção agregam os consumos de Adega e Fabrico, utilizando como referência de volume a soma das tabelas LINKSADEGAS e Links Fabrico. O KPI de CO₂ não é calculado para a Produção, uma vez que só é relevante na Adega.

```

1  /* ----- Água ----- */
2  Agua_Producao =
3  VAR Consumo = SUM('AGUA'[Consumo]) * 1000
4  VAR Volume = SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])
5  RETURN IF(Volume = 0, 0, Consumo / Volume)
6
7  Agua_Producao_Ano1 =
8  CALCULATE([Agua_Producao], SAMEPERIODLASTYEAR('Calendario'[DATE]))
9
10 Agua_Producao_YTD =
11 VAR Consumo = CALCULATE(SUM('AGUA'[Consumo]) * 1000, DATESYTD('Calendario'[DATE]))
12 VAR Volume = CALCULATE(SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
13     DATESYTD('Calendario'[DATE]))
14 RETURN IF(Volume = 0, 0, Consumo / Volume)

```

```

15
16 Agua_Producao_YTD_Ano1 =
17 VAR Consumo = CALCULATE(SUM('AGUA'[Consumo]) * 1000,
18     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
19 VAR Volume = CALCULATE(SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
20     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
21 RETURN IF(Volume = 0, 0, Consumo / Volume)
22
23 PercentT_Producao_Agua = [Agua_Producao] / [Agua_Producao_Ano1] - 1
24 PercentT_Producao_Agua_YTD = [Agua_Producao_YTD] / [Agua_Producao_YTD_Ano1] - 1
25
26 /* ----- Térmica ----- */
27 Termica_Producao =
28 VAR Consumo = (SUM(VAPOR[Consumo]) * 668) + SUM(AQUENTE[Consumo])
29 VAR Volume = (SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100
30 RETURN IF(Volume = 0, 0, Consumo / Volume)
31
32 Termica_Producao_Ano1 =
33 CALCULATE([Termica_Producao], SAMEPERIODLASTYEAR('Calendario'[DATE]))
34
35 Termica_Producao_YTD =
36 VAR Consumo = CALCULATE((SUM(VAPOR[Consumo]) * 668) + SUM(AQUENTE[Consumo]),
37     DATESYTD('Calendario'[DATE]))
38 VAR Volume = CALCULATE((SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100,
39     DATESYTD('Calendario'[DATE]))
40 RETURN IF(Volume = 0, 0, Consumo / Volume)
41
42 Termica_Producao_YTD_Ano1 =
43 VAR Consumo = CALCULATE((SUM(VAPOR[Consumo]) * 668) + SUM(AQUENTE[Consumo]),
44     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
45 VAR Volume = CALCULATE((SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100,
46     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
47 RETURN IF(Volume = 0, 0, Consumo / Volume)
48
49 PercentT_Producao_Termica = [Termica_Producao] / [Termica_Producao_Ano1] - 1
50 PercentT_Producao_Termica_YTD = [Termica_Producao_YTD] / [Termica_Producao_YTD_Ano1] - 1
51
52 /* ----- Eletricidade ----- */
53 Elet_Producao =
54 VAR Consumo = SUM(ELEC[Consumo])
55 VAR Volume = (SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100
56 RETURN IF(Volume = 0, 0, Consumo / Volume)
57
58 Elet_Producao_Ano1 =
59 CALCULATE([Elet_Producao], SAMEPERIODLASTYEAR('Calendario'[DATE]))
60
61 Elet_Producao_YTD =
62 VAR Consumo = CALCULATE(SUM(ELEC[Consumo]), DATESYTD('Calendario'[DATE]))
63 VAR Volume = CALCULATE((SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100,
64     DATESYTD('Calendario'[DATE]))
65 RETURN IF(Volume = 0, 0, Consumo / Volume)
66
67 Elet_Producao_YTD_Ano1 =
68 VAR Consumo = CALCULATE(SUM(ELEC[Consumo]),
69     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
70 VAR Volume = CALCULATE((SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])) / 100,
71     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))

```

```

72 RETURN IF(Volume = 0, 0, Consumo / Volume)
73
74 PercentT_Producao_Elet      = [Elet_Producao]      / [Elet_Producao_Ano1]      - 1
75 PercentT_Producao_Elet_YTD = [Elet_Producao_YTD] / [Elet_Producao_YTD_Ano1] - 1
76
77 /* ----- Ar Comprimido ----- */
78 Ar_Producao =
79 VAR Consumo = SUM(ARCOMPRESSO[Consumo])
80 VAR Volume = SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])
81 RETURN IF(Volume = 0, 0, Consumo / Volume)
82
83 Ar_Producao_Ano1 =
84 CALCULATE([Ar_Producao], SAMEPERIODLASTYEAR('Calendario'[DATE]))
85
86 Ar_Producao_YTD =
87 VAR Consumo = CALCULATE(SUM(ARCOMPRESSO[Consumo]), DATESYTD('Calendario'[DATE]))
88 VAR Volume = CALCULATE(SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
89     DATESYTD('Calendario'[DATE]))
90 RETURN IF(Volume = 0, 0, Consumo / Volume)
91
92 Ar_Producao_YTD_Ano1 =
93 VAR Consumo = CALCULATE(SUM(ARCOMPRESSO[Consumo]),
94     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
95 VAR Volume = CALCULATE(SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
96     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
97 RETURN IF(Volume = 0, 0, Consumo / Volume)
98
99 PercentT_Producao_Ar      = [Ar_Producao]      / [Ar_Producao_Ano1]      - 1
100 PercentT_Producao_Ar_YTD = [Ar_Producao_YTD] / [Ar_Producao_YTD_Ano1] - 1
101
102 /* ----- Volume de Produção ----- */
103 Vol_Producao =
104 SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao])
105
106 Vol_Producao_Ano1 =
107 CALCULATE([Vol_Producao], SAMEPERIODLASTYEAR('Calendario'[DATE]))
108
109 Vol_Producao_YTD =
110 CALCULATE(
111     SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
112     DATESYTD('Calendario'[DATE])
113 )
114
115 Vol_Producao_YTD_Ano1 =
116 CALCULATE(
117     SUM(LINKSADEGAS[Producao]) + SUM('Links Fabrico'[Producao]),
118     DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE]))
119 )

```

Listagem C.3: Medidas DAX da Produção (vista agregada).

C.4 Fabrico

As medidas do Fabrico seguem o mesmo padrão das Linhas de Enchimento, mas utilizam a tabela `Links Fabrico` e o volume de produção próprio desta área.

```

1  /* ----- Água ----- */
2  Percent_Fabrico_Agua = [Fabrico_mosto] / [Fabrico_Ano-1] - 1
3
4  Percent_Fabrico_Agua_YTD = [Fabrico_mosto_YTD] / [Fabrico_mosto_YTD_Ano1] - 1
5
6  Fabrico_mosto_YTD =
7  VAR Consumo_YTD =
8      CALCULATE(SUM('AGUA'[Consumo]) * 1000, DATESYTD('Calendario'[DATE]))
9  VAR Producao_YTD =
10     CALCULATE(SUM('Links Fabrico'[Producao]), DATESYTD('Calendario'[DATE]))
11 RETURN IF(Producao_YTD <= 0, 0, Consumo_YTD / Producao_YTD)
12
13 Fabrico_mosto_YTD_Ano1 =
14 VAR Consumo_YTD =
15     CALCULATE(SUM('AGUA'[Consumo]) * 1000,
16         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
17 VAR Producao_YTD =
18     CALCULATE(SUM('Links Fabrico'[Producao]),
19         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
20 RETURN IF(Producao_YTD <= 0, 0, Consumo_YTD / Producao_YTD)
21
22 /* ----- Energia Térmica ----- */
23 Percent_Fabrico_Termica = [Fabrico_kwh/hl] / [Fabrico_kwh/hl-1] - 1
24
25 Percent_Fabrico_Termica_YTD = [Fabrico_kwh_hl_YTD] / [Fabrico_kwh_hl_YTD_Ano1] - 1
26
27 Fabrico_kwh_hl_YTD =
28 VAR Vapor_YTD =
29     CALCULATE(SUM(VAPOR[Consumo]) * 668, DATESYTD('Calendario'[DATE]))
30 VAR Producao_YTD =
31     CALCULATE(SUM('Links Fabrico'[Producao]) / 100, DATESYTD('Calendario'[DATE]))
32 RETURN IF(Producao_YTD = 0, 0, Vapor_YTD / Producao_YTD)
33
34 Fabrico_kwh_hl_YTD_Ano1 =
35 VAR Vapor_YTD =
36     CALCULATE(SUM(VAPOR[Consumo]) * 668,
37         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
38 VAR Producao_YTD =
39     CALCULATE(SUM('Links Fabrico'[Producao]) / 100,
40         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
41 RETURN IF(Producao_YTD = 0, 0, Vapor_YTD / Producao_YTD)
42
43 /* ----- Eletricidade ----- */
44 Percent_Fabrico_Elect = ELEC[ELEC_Fabrico] / ELEC[Elec_Fabrico-1] - 1
45
46 Percent_Fabrico_Elect_YTD = ELEC[ELEC_Fabrico_YTD] / ELEC[ELEC_Fabrico_YTD_Ano1] - 1
47
48 ELEC_Fabrico_YTD =
49 VAR Elec_YTD =
50     CALCULATE(SUM(ELEC[Consumo]), DATESYTD('Calendario'[DATE]))

```



```

51 VAR Producao_YTD =
52     CALCULATE(SUM('Links Fabrico'[Producao]) / 100, DATESYTD('Calendario'[DATE]))
53 RETURN IF(Producao_YTD = 0, 0, Elec_YTD / Producao_YTD)
54
55 ELEC_Fabrico_YTD_Ano1 =
56 VAR Elec_YTD =
57     CALCULATE(SUM(ELEC[Consumo]),
58         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
59 VAR Producao_YTD =
60     CALCULATE(SUM('Links Fabrico'[Producao]) / 100,
61         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
62 RETURN IF(Producao_YTD = 0, 0, Elec_YTD / Producao_YTD)
63
64 /* ----- Ar Comprimido ----- */
65 Percent_Fabrico_Ar = [Fabrico] / [Fabrico Ano-1] - 1
66
67 Fabrico_YTD =
68 VAR ArComp_YTD =
69     CALCULATE(SUM(ARCOMPRESSO[Consumo]), DATESYTD('Calendario'[DATE]))
70 VAR Producao_YTD =
71     CALCULATE(SUM('Links Fabrico'[Producao]), DATESYTD('Calendario'[DATE]))
72 RETURN IF(Producao_YTD = 0, 0, ArComp_YTD / Producao_YTD)
73
74 Fabrico_YTD_Ano1 =
75 VAR ArComp_YTD =
76     CALCULATE(SUM(ARCOMPRESSO[Consumo]),
77         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
78 VAR Producao_YTD =
79     CALCULATE(SUM('Links Fabrico'[Producao]),
80         DATESYTD(SAMEPERIODLASTYEAR('Calendario'[DATE])))
81 RETURN IF(Producao_YTD = 0, 0, ArComp_YTD / Producao_YTD)

```

Listagem C.4: Medidas DAX do Fabrico.