Um Sistema de *Business Intelligence* Aplicado à Análise de Assiduidade

Mestrado em Ciência de Dados

Rodrigo Silva Nazaré

Leiria, março de 2025

Um Sistema de *Business Intelligence* aplicado à Análise de Assiduidade

Mestrado em Ciência de Dados

Rodrigo Silva Nazaré

Trabalho de Projeto realizado sob a orientação da Professora Doutora Maria Beatriz Piedade e da Professora Doutora Rosa Matias.

Leiria, março de 2025

Originalidade e Direitos de Autor

O presente relatório de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Agradecimentos

Ao longo deste projeto, em momento algum estive sozinho. Assim, gostaria de expressar a minha sincera gratidão a todas as pessoas que contribuíram de forma significativa para a sua concretização:

À minha esposa, por tornar esse caminho mais leve, pelo apoio incondicional, por ser meu porto seguro e por nunca me deixar pensar em desistir;

Aos meus filhos, pelos sorrisos graciosos e por compreenderem a minha ausência;

Aos meus pais, irmãos e demais familiares, pelo incentivo e por estarem sempre presentes, apesar da distância física;

Às professoras orientadoras Profª Drª Rosa Matias e Profª Drª Maria Beatriz Piedade, pela paciência, disponibilidade e pelos ensinamentos, sem os quais a concretização deste trabalho não teria sido possível;

Ao corpo docente do Mestrado em Ciência de Dados pela generosidade em partilhar conhecimentos valiosos ao longo do curso;

A todos aqueles que, de forma direta ou indireta, contribuíram para a concretização deste trabalho, muito obrigado!

Resumo

Num contexto de crescente complexidade institucional e exigências regulatórias, a gestão eficiente dos recursos materiais e humanos torna-se essencial para o bom funcionamento de uma instituição de ensino superior. Entre os aspetos críticos da organização académica, o controlo da assiduidade dos estudantes às aulas é fundamental para a alocação eficaz de recursos e o cumprimento de regulamentos internos.

Para enfrentar estes desafios, a utilização de *Business Intelligence* assume um papel relevante no apoio à tomada de decisão, fornecendo suporte orientado por dados e permitindo análises mais rigorosas e fundamentadas.

Neste contexto, este trabalho apresenta um estudo sobre *Business Intelligence* e descreve as etapas da conceção e do desenvolvimento de um sistema aplicado à análise de assiduidade dos estudantes às aulas para a Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

A solução desenvolvida segue uma abordagem modular, incluindo a criação de um *data warehouse,* o processo de extração-transformação-carga, além de *scripts* para atualização periódica dos dados.

O sistema final inclui *dashboards* com um conjunto de recursos que permitem a monitorização eficaz da assiduidade, possibilitando que as decisões sobre o encerramento ou desdobramento dos turnos sejam tomadas atempadamente, com base em dados disponíveis.

Além dos *insights* sobre a assiduidade, o sistema disponibiliza informação sobre os docentes, sobre os ciclos de estudo e oferece ainda um recurso adicional para auxiliar a composição dos grupos disciplinares.

Ao proporcionar uma visão abrangente da assiduidade, o sistema torna-se uma ferramenta estratégica para a gestão académica, promovendo maior eficiência no planeamento e na tomada de decisões institucionais.

Palavras-chave: *Business Intelligence,* instituição de ensino superior, controlo da assiduidade.

Abstract

In the context of increasing institutional complexity and regulatory demands, the efficient management of material and human resources becomes essential for the smooth operation of a higher education institution. Among the critical aspects of academic organization, monitoring student attendance is fundamental for effective resource allocation and compliance with internal regulations.

To address these challenges, the use of Business Intelligence plays a key role in supporting decision-making, providing data-driven insights and enabling rigorous, evidence-based analyses. In this context, this study presents an exploration of Business Intelligence and outlines the design and development phases of a system applied to attendance analysis for the School of Technology and Management of the Polytechnic University of Leiria.

The developed solution follows a modular approach, including the creation of a data warehouse, an extract-transform-load process, and scripts for periodic data updates. The final system incorporates dashboards with analytical tools that enable effective attendance monitoring, allowing decisions regarding the closure or splitting of class shifts to be made in a timely manner, based on available data.

Beyond attendance insights, the system provides information on teaching staff and study cycles and offers an additional resource to assist in organizing disciplinary groups.

By delivering a comprehensive view of attendance, the system serves as a strategic tool for academic management, enhancing efficiency in institutional planning and decision-making.

Keywords: Business Intelligence, higher education institution, attendance monitoring.

Índice

Lista de figuras

Lista de tabelas

Lista de siglas e acrónimos

AED Análise exploratória de dados.

API *Application Programming Interface.*

BI *Business Intelligence.*

BIAA *Business Intelligence and analytical applications.*

CDCS *Change Data Capture System.*

CSF Fatores críticos de sucesso.

DAX *Data Analysis Expressional.*

DW *Data Warehouse.*

ESTG Escola Superior de Tecnologia e Gestão.

ETL Extração, transformação e carga.

IES Instituições de Ensino Superior.

IPL Instituto Politécnico de Leiria.

KPI Indicadores de Desempenho.

MER Modelo entidade-relacionamento.

SCD *Slowly Changing Dimensions.*

SGBD Sistema de Gestão de Bases de Dados.

SQL *Structured Query Language.*

SSIS *SQL Server Integration Services.*

UC Unidade curricular.

1. Introdução

As Instituições de Ensino Superior (IES) enfrentam desafios cada vez mais complexos. A necessidade contínua de melhorar a eficiência operacional, aperfeiçoar a gestão de recursos, cumprir normativas internas e externas e elevar a qualidade do ensino e da investigação exige que estas organizações se adaptem rapidamente às mudanças e adotem uma abordagem estratégica, assente em dados, para a tomada de decisão.

Nesse contexto, *Business Intelligence* (BI) apresenta-se como uma solução para transformar dados brutos em informação estruturada e relevante. Ao integrar, processar e analisar grandes volumes de dados, o uso de BI possibilita uma visão abrangente e detalhada das operações institucionais, possibilitando que as IES tomem decisões estratégicas mais assertivas, aprimorando processos, reduzindo custos e aumentando a sua produtividade.

O presente trabalho propõe-se a abordar uma questão fundamental para a Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria (ESTG/IPL): a gestão e monitorização das presenças em sala de aula (assiduidade) dos estudantes nos diversos turnos em que estão inscritos.

Reconhecendo a importância que uma gestão académica eficiente tem para o sucesso institucional, este estudo pretende avaliar como o uso de BI pode contribuir para melhorar a organização e o acompanhamento do número de estudantes que estão a frequentar os turnos de unidades curriculares que estão a funcionar ao longo dos semestres na instituição.

A análise do número total de presenças nas unidades curriculares e nos seus turnos permite identificar os que precisam de ser desdobrados, por terem um elevado número de estudantes, e turnos que precisam ser encerrados, por terem um número reduzido de estudantes. Desta forma, garante-se uma maior eficiência no processo de ensino-aprendizagem e no aproveitamento dos recursos disponíveis.

O acompanhamento contínuo da evolução da assiduidade contribui para a melhoria da organização dos turnos e a otimiza a distribuição dos recursos educativos.

Além de um estudo prévio sobre o tema BI, o presente trabalho objetiva a conceção e o desenvolvimento/implementação de um sistema que, efetivamente, possa ser utilizado pela ESTG/IPL, auxiliando a instituição na análise e suporte a decisões estratégicas.

1.1. Enquadramento

O Instituto Politécnico de Leiria (IPL) é uma IES pública, localizada na região central de Portugal. Atualmente, o IPL possui cinco escolas superiores e mais de treze mil alunos, distribuídos por cerca de cento e setenta cursos.

Com ofertas de cursos nas áreas de Engenharia, Gestão, Ciências Empresarias e Ciências Jurídicas a Escola Superior de Tecnologia e Gestão (ESTG) é a maior das escolas superiores do IPL, com aproximadamente seis mil alunos.

Por exemplo, no último ano letivo (2023/2024) a ESTG ofereceu mais de 70 cursos e aproximadamente mil e duzentas unidades curriculares, distribuídas em mais de três mil e duzentos turnos.

Neste contexto, a gestão eficaz de turnos desempenha um papel essencial na administração de recursos humanos e materiais. Além da distribuição inicial, é fundamental acompanhar e realizar os ajustes necessários nos momentos oportunos, para garantir o cumprimento das normativas internas e o adequado ambiente de ensino-aprendizagem.

O IPL utiliza uma aplicação de gestão científica e pedagógica para controlar a assiduidade dos alunos. Esse sistema armazena o número total de estudantes presentes nas aulas, sendo esse número registado pelos docentes, no sumário da aula.

Contudo, não obstante a existência e obrigatoriedade de registo, não existe à data qualquer tipo de estrutura analítica ou consolidação de dados para apoio a tomada de decisão relativa a esta gestão.

Inúmeras vezes a análise das presenças em sala de aula é realizada de forma manual e não sistematizada pelos responsáveis pela gestão dos turnos, o que implica alguma suscetibilidade a erros e consumo de tempo e recursos.

Desta forma, a aplicação de técnicas e o desenvolvimento de uma solução de BI, pretende automatizar as tarefas relativas à monitorização das presenças e proporcionar uma tomada de decisão informada, através de análise de tendências e identificação de padrões.

1.2. Problemática

A dinâmica da assiduidade nas aulas configura-se como um elemento essencial para a implementação de uma administração eficaz dos recursos humanos e físicos em uma instituição de ensino. Nesse contexto, a gestão de turnos constitui uma etapa complexa e de extrema relevância, pois possibilita à organização e distribuição dos alunos de maneira a otimizar os recursos disponíveis.

Este processo é fundamental para garantir a conformidade com as diretrizes institucionais, assim como para atender às necessidades pedagógicas, assegurando a eficácia da estruturação curricular e a maximização do aproveitamento dos recursos educacionais.

Antes de avançarmos, é importante a apresentação de uma breve explicação sobre os conceitos que são essenciais para a compreensão da organização curricular: a unidade curricular, a componente e o turno. Estes três conceitos estão interligados e desempenham um papel fundamental na estruturação da oferta formativa:

- **Unidade Curricular (UC)**: é a unidade de ensino com objetivos e conteúdos de formação próprios, objetos de inscrição administrativa e de avaliação, em alguns contextos também é chamada de disciplina (Universidade de Coimbra, 2025). No IPL, cada unidade curricular possui um código único e um nome, e pode abranger diferentes componentes;
- **Componente:** define o tipo de atividade académica que será realizada ao longo da unidade curricular. No IPL, estas componentes podem ser teóricas, práticas, teórico-práticas, laboratoriais, seminário, entre outras;
- **Turno:** refere-se ao horário específico em que uma componente da unidade curricular é ministrada. Pode existir mais de um turno para a mesma componente de uma unidade curricular, com diferentes horários, permitindo que os estudantes sejam associados ao turno que melhor se adapta à sua disponibilidade.

A gestão das atividades académicas e dos horários escolares dos cursos envolve a organização e distribuição das unidades curriculares, suas componentes e os respetivos turnos, garantindo que os estudantes sejam associados de forma eficiente aos turnos das unidades curriculares em horários que lhes permitam frequentar as aulas.

No IPL, em geral, a cada componente da unidade curricular, está associado um turno, mas podem existir vários turnos, face ao número de estudantes inscritos nas unidades curriculares de cada curso.

Podem, também, existir grupos disciplinares de unidades curriculares, que envolvem o agrupamento de turnos de unidades curriculares pertencentes a cursos distintos, mas com conteúdos programáticos iguais. Normalmente, esta situação ocorre quando existe um número reduzido de estudantes inscritos por unidade curricular num determinado curso, justificando-se o agrupamento no mesmo turno de estudantes de vários cursos.

A definição do número inicial de turnos é um processo complexo que envolve o Conselho Técnico-Científico (CTC), os Coordenadores de Departamentos e os Coordenadores de Cursos.

Inicialmente, com base no histórico da distribuição de turnos dos semestres anteriores, nos níveis de assiduidade dos estudantes e na previsão de inscrições para o ano letivo em análise, o CTC elabora uma proposta preliminar. Esta proposta é submetida à análise das Coordenações de Departamento que, em articulação com as Coordenações de Curso, podem solicitar ajustes. Caso existam solicitações de alteração, o CTC procede à sua avaliação, podendo aceitá-las ou rejeitá-las, e, subsequentemente, efetiva a abertura dos turnos.

No entanto, o processo de gestão de turnos não se encerra nesse momento, pois a procura inicialmente prevista nem sempre se concretiza com o início das aulas. É frequente que, ao longo do semestre, alguns estudantes deixem de comparecer às atividades letivas.

Sendo assim, torna-se imprescindível realizar um acompanhamento contínuo da assiduidade, de modo a assegurar a continuidade do cumprimento das normativas internas do IPL relativas ao número mínimo de estudantes que frequentam os turnos, melhorando a eficiência da utilização dos recursos.

As semanas letivas cinco e dez são consideradas períodos críticos, uma vez que, nesses momentos, os coordenadores de departamento são informados sobre a assiduidade média de cada turno e, em conjunto com os coordenadores de curso, identificam a eventual necessidade de ajustes, nomeadamente a criação e o encerramento de turnos.

Atualmente, essa informação é disponibilizada através de ficheiros *Excel*, onde os dados estão somente no formato tabular, o que dificulta a análise.

Além disso, como essa informação é apresentada aos coordenadores apenas nas semanas críticas, que são as associadas aos encerramentos de turnos, esta limitação impede a identificação atempada de situações problemáticas, comprometendo que sejam realizados ajustes antecipadamente.

Este atraso acaba também por prolongar o processo de encerramento ou a criação de turnos, que seria desejável que ocorresse nas semanas iniciais do semestre.

1.3. Objetivos

Diante desse cenário, o presente estudo propõe a conceção e o desenvolvimento de um sistema de BI para auxiliar o monitoramento da evolução da assiduidade dos estudantes ao longo dos semestres na ESTG/IPL.

O intuito é analisar o comportamento da assiduidade às aulas em diferentes cursos e unidades curriculares/componentes/turnos ao longo do tempo.

O sistema deve integrar dados provenientes de diferentes aplicações de gestão científica e pedagógica da instituição para fornecer informação e *insights* valiosos. Os coordenadores terão a possibilidade de monitorar a evolução da assiduidade sempre que considerarem necessário, podendo intervir de forma imediata quando identificarem alguma anormalidade.

Além de informação sobre a assiduidade dos estudantes, o sistema disponibiliza informação pertinente sobre docentes, os diferentes ciclos de estudos (TeSP, Licenciatura, Mestrado) e as respetivas unidades curriculares. Adicionalmente, associado às semanas críticas, são disponibilizados indicadores específicos, para permitir a monitorização mais detalhada da assiduidade.

O sistema analisa o histórico de presença dos alunos e deve estar preparado para atualizações numa base diária e/ou semanal.

A solução desenvolvida é composta por um conjunto de painéis com *dashboards* interativos e intuitivos que fornecem uma narrativa sobre diversas métricas e indicadores-chave de desempenho, identificados como relevantes para a gestão da assiduidade.

1.4. Metodologia

Este trabalho segue uma metodologia qualitativa de pesquisa, com o objetivo principal de compreender o problema da gestão de turnos da ESTG/IPL e aplicar técnicas de BI para fornecer informação que oriente a tomada de decisão pelos gestores da instituição.

Na primeira fase, é realizada um estudo prévio sobre o assunto em análise para consolidar o conhecimento teórico sobre o tema, por meio de pesquisa bibliográfica. Na segunda fase, um sistema de BI é desenvolvido, para atender aos problemas identificados.

A compreensão aprofundada do problema prático e do contexto em que está inserido foi obtida por meio de entrevistas com Coordenadores, realizadas ao longo de toda a execução do trabalho. Além disso, os Coordenadores disponibilizam dados essenciais para viabilizar o desenvolvimento de um protótipo e permitir a validação preliminar da solução proposta.

Dessa forma, a metodologia adotada abrange tanto a perspetiva teórica quanto a prática, utilizando técnicas e processos para responder às necessidades da instituição e resolver um problema real.

1.5. Estrutura do documento

Este trabalho está estruturado em cinco capítulos. O primeiro constitui a introdução, na qual se apresenta uma visão geral do tema, assim como os objetivos e as metodologias que serão utilizadas.

No segundo capítulo é realizado um estudo prévio sobre BI, abordando o seu conceito, benefícios, fatores críticos de sucesso, arquitetura, ferramentas e uma breve perspetiva sobre BI na nuvem, além de uma revisão de trabalhos relacionados com a sua aplicação em IES. Este enquadramento teórico é essencial para a compreensão da solução prática que será apresentada nos capítulos seguintes.

O terceiro capítulo descreve as etapas para a conceção do sistema de BI proposto, enquanto no quarto capítulo a implementação do sistema é detalhada.

No quinto e último capítulo são apresentados os resultados preliminares da implementação do sistema conclusões e sugestões para estudos futuros.

Os anexos contêm os códigos-fonte dos *scripts* em *Python*, *scripts* em linguagem M e um conjunto de imagens representativas de algumas interações possíveis em um dos *dashboards* desenvolvidos.

2. Estudo Prévio

Este capítulo apresenta um estudo prévio do tema BI, com o propósito de abordar conceitos fundamentais e indispensáveis à compreensão do presente trabalho.

O objetivo deste capítulo não é esgotar o tema, dada a sua amplitude e contínua evolução, mas sim proporcionar um enquadramento conceptual e uma base teórica que assegurem uma compreensão clara dos conteúdos e métodos utilizados.

Portanto, são abordados os conceitos, características, arquitetura, componentes, benefícios e fatores críticos de sucesso associados a um projeto de BI. No final do capítulo, é realizada uma contextualização e apresentadas as tendências atuais, nomeadamente, a migração cada vez mais plausível para soluções de BI na *cloud*.

Adicionalmente, são apresentados estudos correlacionados que abordam o uso de sistemas de BI em IES, analisando suas aplicações, benefícios e desafios na gestão académica.

Este enquadramento teórico é essencial para sustentar a análise e metodologias desenvolvidas na parte prática do presente trabalho, estabelecendo, assim, uma ligação consistente entre teoria e prática.

2.1. Conceito de *Business Intelligence*

Não há uma descrição universalmente aceite para definir o que é BI e são inúmeros os autores que a definem. Desta forma, a seguir, são apresentadas algumas definições sobre o tema.

Para Kimball & Ross (2015), BI é um termo abrangente que pode ser definido como um conjunto de sistemas e processos que uma organização utiliza para recuperar, processar e analisar informação, com objetivo de auxiliar à tomada de decisão (Kimball & Ross, 2015).

Kumar (2021) apresenta uma definição ainda mais ampla, ao considerar que BI é um conceito que inclui, mas não se limita a tecnologias, estratégias, ferramentas, metodologias, arquitetura e pessoas, com principal objetivo de extrair informação relevante de dados brutos, de maneira escalável e eficiente, para apoiar a tomada de decisões nas organizações (Kumar, 2021).

Kumar (2021) destaca ainda a importância da integração de todas essas dimensões para garantir o sucesso da implementação de soluções de BI e alcançar os benefícios desejados, entre eles a obtenção de *insights* estratégicos, a identificação de padrões de desempenho e a melhoria contínua dos processos empresariais (Kumar, 2021).

Para Sharda *et al.* (2017) o processo de BI baseia-se na transformação de dados em informação, depois em decisões e, por fim, em ações (Sharda *et al*., 2017).

A finalidade primordial de um sistema de BI é aprimorar a relevância e a precisão da informação ao longo do tempo para orientar a tomada de decisão (Ranjan, 2009).

Portanto, BI pode ser entendido como um conjunto integrado de sistemas, metodologias, ferramentas e competências humanas que objetivam transformar dados brutos em informação estratégica, facilitando a tomada de decisão fundamentada e a melhoria contínua dos processos empresariais.

2.2. Benefícios do uso de *Business Intelligence*

Em um contexto empresarial altamente dinâmico e competitivo, a capacidade de prever e adaptar-se às mudanças torna-se essencial para a sustentabilidade das organizações (Engelen *et al*., 2014).

BI desempenha um papel fundamental ao transformar dados em informação estratégica, permitindo às organizações identificar tendências, prever cenários e tomar decisões fundamentadas que respondam de forma eficaz aos desafios do mercado (Işik *et al*., 2013).

A utilização eficaz das ferramentas de BI contribui para a eficiência operacional. Com uma análise mais profunda dos comportamentos dos consumidores e do desempenho interno, as empresas conseguem alinhar recursos e estratégias de forma mais precisa, permitindo uma adaptação ágil às mudanças do mercado (Borra, 2024).

Para Ranjan (2009), o uso de BI elimina suposições, pois viabiliza decisões fundamentadas em indicadores internos e externos (Ranjan, 2009). Deste modo, as organizações passam a dispor de uma visão abrangente das suas capacidades e da sua posição relativa no mercado, além de compreenderem os contextos social, regulatório e político em que estão inseridas (Arefin *et al*., 2015).

Jakhar & Krishna (2020) destacam que, com o uso do BI é possível identificar padrões de comportamento do mercado e dos clientes, bem como antecipar tendências e mudanças no ambiente operacional (Jakhar & Krishna, 2020).

Esta capacidade proporciona às organizações uma vantagem competitiva significativa, permitindo-lhes agir proactivamente e adaptar-se de forma mais eficaz às dinâmicas do mercado.

O quadro seguinte (Tabela 1), proposto por Skyrius (2021), exemplifica como o uso de BI pode agregar valor para o negócio, conforme a área de aplicação.

Tabela 1- Benefícios do uso de BI/ tipo de aplicação
Fonte: Skyrius (2021) (adaptado).

Aplicação Analítica	Pergunta de Negócio	Valor para o Negócio
Segmentação de clientes	Quais as principais características dos clientes do nosso segmento de mercado?	Personalizar a comunicação com os clientes com base nas suas características e necessidades específicas.
Análise de carrinho de compras	Quais produtos são comprados juntos? Porquê?	Descobrir relações e regras para melhorar a promoção cruzada e as vendas cruzadas.
Lucratividade do cliente	Qual é a lucratividade ao longo da vida de um cliente? Quais são as dinâmicas de lucratividade dentro do ciclo de vida do cliente?	Identificar formas de gerir e aumentar a lucratividade geral dos clientes.
Deteção de fraudes	Como posso identificar quais transações ou agentes que provavelmente são fraudulentos?	Determinar fraudes antecipadamente e tomar ações imediatas para minimizar perdas ou prevenir fraudes.
Evasão de clientes	Quais clientes têm maior probabilidade de sair?	Prevenir a perda de clientes de alto valor e abrir mão de clientes de menor valor.
Monitorização de concorrentes	Quais concorrentes provavelmente serão fontes de sérios problemas no futuro próximo?	Implementar medidas preventivas bem direcionadas para proteger a posição no mercado e o potencial futuro.
Monitorização de inovações	Quais inovações tecnológicas (ou de mercado) são esperadas para causar o maior impacto na indústria?	Minimizar os riscos nas decisões relacionadas a inovações.

O trabalho de Skyrius (2021) evidencia que o BI pode ser aplicado em diversos domínios, permitindo não apenas uma compreensão mais aprofundada da própria organização, mas também uma análise estratégica do mercado e da concorrência.

2.3. Arquitetura base de um sistema de *Business Intelligence*

Conforme descrito por Sherman (2015), a arquitetura tecnológica de um sistema de BI é composta por quatro camadas fundamentais: Fontes de Dados, Integração de Dados, *Data Warehouse (*DW) e *Business Intelligence* (Sherman, 2015). A Figura 1 apresenta uma representação esquemática dessa arquitetura, evidenciando a organização em camadas.

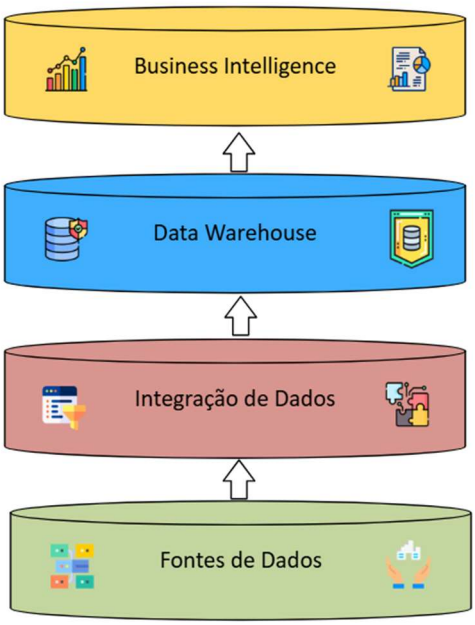


Figura 1 – Arquitetura de um sistema de BI.
Fonte: Sherman (2015) (adaptado).

Nesta estrutura a camada de Fontes de Dados é a camada inicial, representando o ponto de origem dos dados que servirão como base para o sistema de BI. É na camada de integração que os dados são extraídos das fontes, transformados e carregados através de um processo designado por *Extract, Transform and Load* (ETL) para o DW. O DW assegura o armazenamento estruturado dos dados. Por fim, a camada de BI disponibiliza os dados processados para análise, disponibilizando a informação para auxílio na tomada de decisão.

Os tópicos seguintes abordarão, de forma detalhada, cada das camadas do modelo de arquitetura proposto por Sherman (2015).

2.3.1. Camada das Fontes de Dados

Dados constituem a matéria-prima do sistema de BI. Por isso, a identificação, a verificação e a validação rigorosa das fontes desses dados são etapas cruciais para o sucesso do projeto (Sharda *et al*., 2017).

No início dos projetos de BI, as fontes de dados restringiam-se basicamente a sistemas transacionais da própria organização. Essas fontes tinham a vantagem de forneceram dados estruturados e geridos pela própria empresa. Entretanto, costumavam ser limitadas em relação à frequência de atualização e volume de dados (Sherman, 2015).

O cenário mudou bastante nos últimos anos. Atualmente, os dados podem ser disponibilizados tanto por fontes internas quanto externas à organização, além disso, podem estar em formato estruturado, semiestruturado ou não estruturado. Essa mudança causou um aumento expressivo no volume de dados e na necessidade de integração dos mesmos.

Na Figura 2, proposta por Sherman (2015), é possível identificar as diversas possibilidades de fontes que podem compor a camada de fonte de dados. Estas fontes podem incluir sistemas internos da organização, como bases de dados transacionais, arquivos de log, sistemas de gestão empresarial (ERP), e até fontes externas, como *Application Programming Interfaces* (APIs), dados provenientes da web ou serviços de dados de terceiros.

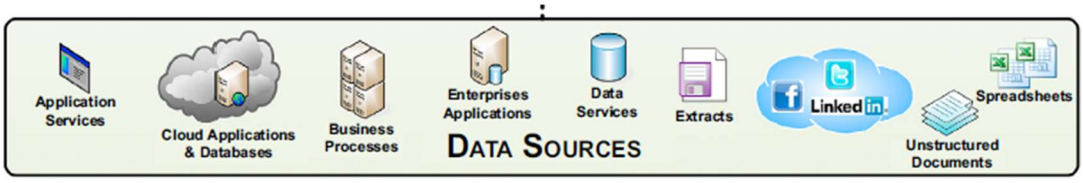


Figura 2 - *Data Sources* (Fontes de Dados).
Fonte: Sherman (2015).

A camada de Fontes de Dados é a primeira da arquitetura de um sistema de BI, desempenhando um papel fundamental na disponibilização de dados brutos que, posteriormente, serão processados e integrados nas camadas seguintes.

Por isso, a adequada definição das fontes de dados é essencial para o sucesso do desenvolvimento do sistema de BI.

2.3.2. Camada de Integração de Dados

Depois de selecionar as fontes de dados e descrevê-las, incluindo perguntas de análise e formatos, o próximo passo é extrair os dados relevantes. A etapa de extração estabelece a ligação entre o sistema de BI e os diversos sistemas operacionais da organização, sendo responsável pela recolha de dados a partir de múltiplas fontes.

Normalmente, a integração de dados é parte do processo ETL (Grossmann & Rinderle-Ma, 2015). E, em termos gerais, o processo de ETL constitui a etapa mais dispendiosa na construção de um DW (Munawar, 2021).

A extração (E) representa o ponto de entrada dos dados no processo de BI. Esta etapa consiste na transferência dos dados a partir das suas fontes originais para o ambiente do DW, garantindo que as fases subsequentes do processo de ETL possam ser executadas sem interferir com as fontes originais dos dados (Kimball & Ross, 2015).

De acordo com Kimball e Ross (2015) o processo de extração é composto por três subprocessos (Kimball & Ross, 2015):

- *Data profiling:* análise técnica para identificar e melhor compreender os dados, a sua estrutura, o seu conteúdo e a sua consistência;
- *Change Data Capture System (CDCS):* definição da forma de como os dados são transferidos entre as fontes originais e os repositórios analíticos. Usualmente, a primeira transferência de dados é completa, ou seja, envolve todos os dados. Nas transferências subsequentes, apenas as alterações relevantes desde última transferência são processadas;
- *Extraction*: núcleo da etapa de extração. Esse subprocesso é a interface entre as fontes de dados e o sistema de armazenamento que será utilizado pelo BI. Envolve tarefas de comunicação com as fontes e a efetiva transferência dos dados, e definição de detalhes como formato da transferência (fluxo, arquivo, etc.), necessidade de compressão e criptografia. A depender da quantidade e da variedade das fontes de dados, o subprocesso de extração pode ser extremamente complexo.

Após a extração, os dados precisam ser tratados/transformados. Frequentemente, a etapa de transformação (T) é a que consome mais tempo do processo de ETL, especialmente quando existem múltiplas fontes e dados em diferentes formatos (Howson, 2014).

A transformação é composta pelos subprocessos de limpeza e conformação dos dados. Nessa fase há alteração dos dados e aumento de valor para a organização.

Durante a fase de transformação podem ser criados metadados para auxiliar o diagnóstico de problemas nas fontes originais, melhorando a qualidade dos dados ao longo do tempo (Kimball & Ross, 2015).

De modo geral, o processo de transformação de dados caracteriza-se por alguma complexidade e, frequentemente, envolve múltiplas etapas que permitem a integração de dados provenientes de diversas fontes.

Para assegurar a integridade dos dados originais e aprimorar a eficiência do processo de transformação, é amplamente adotada a utilização de uma área intermediária e temporária, denominada *Staging Area*.

A etapa final do processo de ETL é o carregamento (*Load*) dos dados já tratados num *DW* ou sistema similar (Kimball & Ross, 2015).

A integração de dados não se limita à extração de conjuntos de dados de fontes internas e externas e ao carregamento deles em um DW, mas concentra-se em facilitar efetivamente a entrega de informação nos lugares certos dentro do tempo adequado (Loshin, 2013) .

2.3.3. Camada do *Data Warehouse*

O modelo de DW surgiu no final da década de 1980, como uma proposta de solução para resolver o fluxo de dados em ambientes de suporte à decisão, sendo descrito inicialmente em um artigo publicado no *IBM Systems Journal* (Devlin, 2018).

Inmon (2005) formalizou o conceito, definindo DW como um sistema projetado para executar consultas analíticas e armazenar grandes volumes de dados históricos, organizados por assunto, integrados, não-voláteis e variáveis ao longo do tempo, com o objetivo de auxiliar a tomada de decisão (Inmon, 2005).

Com base no conceito proposto por Inmon (2005), Devlin (2018) define as principais características de um DW:

- **Organização por assunto**: Os dados devem ser representados por termos e estruturas que sejam familiares aos profissionais de negócios, sendo organizados

em áreas temáticas alinhadas com as necessidades de informação da organização;

- **Integração**: Os dados presentes num DW provêm de fontes distintas. Para garantir a consistência e permitir a extração eficiente da informação, é essencial que esses dados sejam devidamente integrados, reconciliando eventuais discrepâncias entre as várias origens;

- **Não-volatilidade:** Um DW deve manter os dados de forma estável e persistente ao longo do tempo, sem sofrer alterações ou exclusões frequentes;

- **Variação ao longo do tempo:** Esta característica está diretamente associada à não volatilidade e visa garantir que todas as versões dos dados, tanto as atuais quanto as passadas, sejam preservadas, possibilitando a análise da evolução dos dados ao longo do tempo. Técnicas que passam pela utilização de *timestamps* são empregues na implementação da variação dos dados ao longo do tempo.

Para Kimbal e Ross (2015), o objetivo principal de um DW é servir como uma fonte de dados centralizada, padronizada, otimizada, confiável e de acesso simplificado, para apoio à tomada de decisão (Kimball & Ross, 2015).

O aspeto mais relevante na construção de um DW é o nível de detalhe ou agregação das unidades de dados, denominado granularidade. A granularidade exerce uma influência significativa sobre o volume de dados armazenados e sobre os tipos de consultas que podem ser efetuadas, uma vez que define o nível de detalhe disponível para análise e o esforço necessário para processar os dados (Inmon, 2005).

Quanto menor a granularidade, maior o nível de detalhe, maior o volume de dados e a necessidade de processamento.

A Figura 3 ilustra dois DWs (DW1 e DW2) onde a granularidade é diferente.

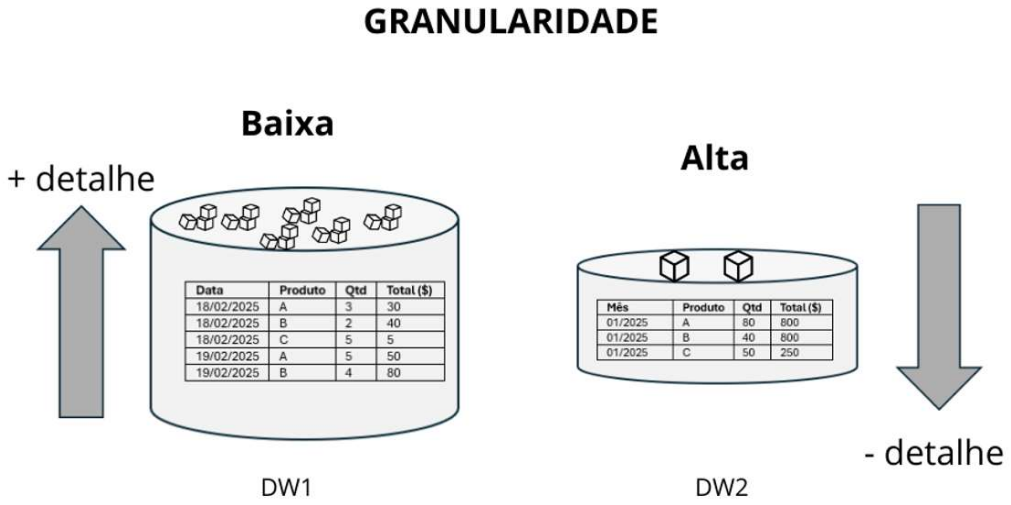


Figura 3 - Níveis de granularidade do DW.
Fonte: Governo do Estado de Mato Grosso (2006) (adaptado)

O DW1, caracterizado por uma granularidade menor, armazena dados de vendas com um maior nível de detalhe, registando transações diárias.

Em contraste, o DW2 apresenta uma granularidade maior, armazenando exclusivamente a consolidação das vendas a nível mensal. Devido ao seu maior nível de detalhe, o DW1 permite a realização de análises mais pormenorizadas, contudo, requer um volume de armazenamento substancialmente superior ao do DW2.

Apesar dos benefícios, a construção de um DW requer um investimento significativo de tempo, dinheiro e esforço de gestão por parte da organização(Coronel & Morris, 2016). Os principais fatores que influenciam no custo da construção de um DW são o escopo (âmbito) e a complexidade do projeto, a escolha da infraestrutura, os custos com software e a mão de obra especializada (Ahmed, 2024). A seguir, esses fatores são detalhados:

- **Escopo e complexidade do projeto**: O volume de dados, a quantidade de fontes a serem integradas e os requisitos de transformação afetam a necessidade de processamento e armazenamento, impactando diretamente no custo do projeto;
- **Infraestrutura (On-premises vs. Nuvem)**: Soluções *on-premises* exigem investimentos elevados em hardware, licenciamento e manutenção, enquanto as

soluções em nuvem oferecem maior flexibilidade e escalabilidade com um modelo de pagamento conforme o uso;

- **Software**: Custos com licenciamento de ferramentas de ETL bem como plataformas de análise e visualização, representam um item significativo no orçamento;

- **Mão de obra especializada**: A necessidade de profissionais qualificados, como engenheiros de dados, arquitetos de soluções e administradores de banco de dados gera custos com equipe especializada para implementar e manter a solução.

Por conta da complexidade e dos custos da construção de um DW, muitas organizações optam por criar conjuntos de dados menores, mais gerenciáveis e direcionados a atender às necessidades de uma unidade específica da organização. Esses conjuntos de dados menores são conhecidos como *data marts (DM)* e proporcionam uma resposta rápida às necessidades dessa área específica da empresa (Coronel & Morris, 2016).

Os DMs, ao se concentrarem nos objetivos e nas necessidades de suporte à decisão de um departamento específico dentro da empresa, contêm uma quantidade significativamente menor de dados, direcionada apenas às operações desse departamento (Loshin, 2013).

Em relação à arquitetura de um DW, Kimball & Ross (2015) e Inmon (2005) apresentam duas abordagens distintas.

Inmon (2005) propõe um modelo *top-down* **(**de cima para baixo), centrado num *Enterprise Data Warehouse* (EDW), onde os dados da organização são armazenados de forma centralizada, organizada e estruturada num modelo entidade-relacionamento (Sharda *et al*., 2017). Neste caso, o EDW é criado inicialmente com uma visão holística da organização, e, a partir dele, podem ser derivados DMs para atender às necessidades específicas dos departamentos (Inmon, 2005).

Em contrapartida, o modelo proposto por Kimball & Ross (2015) apresenta uma metodologia *bottom-up* (de baixo para cima), na qual a construção do DW inicia-se a partir da identificação e realização dos modelos DMs específicos para cada área de negócio. Os DMs devem possuir um conjunto padronizado de definições e estruturas (*bus architecture)* para garantir a viabilidade da integração e a construção do DW (Sharda *et al*., 2017) .

A arquitetura proposta por Kimball & Ross (2015) faz uso do modelo dimensional de dados, que será detalhado no tópico a seguir.

Breslin (2004) elaborou um quadro comparativo que apresenta as principais diferenças entre esses modelos de arquitetura de DW. Um resumo dessa informação pode ser observado na tabela abaixo.

Tabela 2 Quadro comparativo arquiteturas DW.
Fonte: Breslin (2004) (adaptado).

Categoria	Inmon	Kimball
Metodologia e Arquitetura	**Top-down**	**Bottom-up**
Estrutura Arquitetónica	DW empresarial (nível atómico) que alimenta bases de dados departamentais	*Data Marts* modelam processos de negócio individuais; consistência empresarial alcançada através do *data bus* e dimensões conformadas
Complexidade do Método	Bastante complexa	Relativamente simples
Discussão do Design Físico	Bastante aprofundada	Relativamente leve
Orientação dos Dados	Modelagem tradicional (*entidade relacionamento* - ERS, *Data Item Sets* - DISs)	Modelagem dimensional; afastamento do modelo relacional
Acessibilidade para o Utilizador Final	Baixa	Alta
Público-Alvo Principal	Profissionais de TI	Utilizadores finais
Objetivo	Fornecer uma solução técnica sólida baseada em métodos e tecnologias comprovadas de bases de dados	Fornecer uma solução que permita aos utilizadores finais consultar diretamente os dados com tempos de resposta razoáveis

A análise do quadro comparativo apresentado por Breslin (2004) revela que ambas as arquiteturas apresentam vantagens e desvantagens, não existindo uma que seja superior à outra em todos os aspetos.

Embora a construção de um EDW, consolidando bases de dados dispersas, constitua a solução mais robusta, o desenvolvimento inicial de DMs individuais pode proporcionar benefícios significativos ao longo do processo de construção de um DW, especialmente quando a organização não dispõe da capacidade ou disponibilidade para investir num projeto de grande escala (Sharda *et al*., 2017).

Modelo Dimensional

O desenvolvimento do modelo entidade-relacionamento (MER) normalizado foi um marco na evolução dos sistemas de bases de dados relacionais. Essa mudança contribuiu para a otimização das atividades comerciais e para o sucesso da utilização de sistemas informáticos nas empresas (Barcelar, 2012).

O alto grau de normalização utilizado nos MER proporciona eliminação da redundância, reduz a possibilidade de inconsistência e otimiza a utilização de espaço. Essas características são essenciais nos sistemas utilizados para gestão das atividades diárias das organizações**,** pois neles é fundamental o processamento de um elevado número de transações, em que cada uma envolve uma quantidade reduzida de dados.

Contudo, o elevado número de tabelas e relações, resultante da normalização no MER, compromete a eficiência da análise de grandes volumes de dados (Loshin, 2013). Além disso, as estruturas normalizadas, ao dispersarem os dados por múltiplas tabelas interligadas, nem sempre são intuitivas ou de fácil interpretação para os *stakeholders*.

Nesse contexto, a partir da década de 1990, com o crescimento exponencial do volume de dados e a crescente necessidade das organizações em fundamentar decisões estratégicas em análises estruturadas, Ralph Kimball desenvolveu o modelo dimensional. Este modelo foi concebido para otimizar o armazenamento e a recuperação de dados, proporcionando uma estrutura mais adequada às necessidades dos processos analíticos.

O modelo dimensional, organiza os dados num formato que oferece compreensibilidade para o utilizador, desempenho em consultas e maior capacidade de adaptação às mudanças (Kimball & Ross, 2015).

Enquanto o modelo normalizado tradicional pretende eliminar a redundância para otimizar operações de escrita de dados e evitar anomalias, no modelo dimensional as redundâncias podem existir, mas são controladas para reduzir a complexidade, melhorar o desempenho de operações de consulta e potencializar o processo de análise.

No modelo dimensional os dados são estruturados em tabelas de fatos e tabelas de dimensão. Os fatos são eventos que ocorrem frequentemente e para os quais é necessário um processo de decisão, enquanto as dimensões são propriedades que descrevem um fato específico (Garani *et al*., 2019).

Por meio das dimensões, os utilizadores podem analisar os fatos através de diferentes perspetivas e responder a perguntas específicas, tornando as dimensões essenciais como eixos de análise.

Em ambientes de DW, o modelo dimensional apresenta diversas vantagens, nomeadamente (Kimball, 1997):

- Facilita a compreensão e o acesso aos dados;
- Torna o processo de análise mais rápido, eficiente e produtivo;
- Permite a adaptação do modelo a novas necessidades das organizações;
- Possui abordagens padronizadas para lidar com situações comuns de modelagem em negócios, entre elas dimensões de mudança lenta, produtos heterogêneos, bancos de dados de eventos.

2.3.4. Camada de *Business Intelligence*

Para os gestores, responsáveis pelo ato de decidir, as fontes de dados, a integração e o DW, em geral, são tratados como um sistema fechado e enigmático.

Nesse contexto, a camada de BI, ou *Business Intelligence & Analytics Applications (BIAA)*, engloba técnicas de análise e a parte visível do sistema de BI para o utilizador. Portanto, é nessa camada que se concretiza o objetivo fundamental de todo o processo de BI, a entrega de informação relevante e *insights* valiosos (Sherman, 2015).

Uma proposta de BIAA é apresentada por Sherman (2015) na Figura 4.

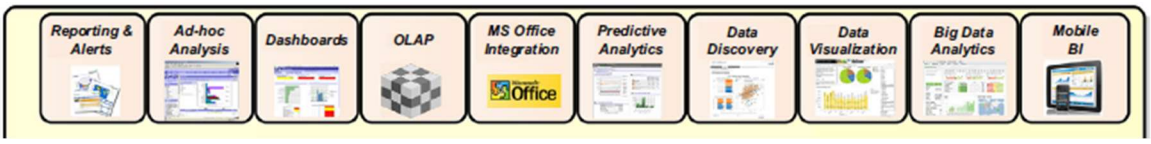


Figura 4 - Camada de BI.
Fonte: Sherman (2015).

Conforme ilustrada na figura, na BIAA estão presentes ferramentas de análise de dados, que objetivam identificar padrões subjacentes, tendências, comportamentos e correlações, proporcionando uma compreensão mais aprofundada dos dados e facilitando a obtenção de informação.

Nesse contexto, Sherman (2015) destaca o uso de processamento analítico on-line (OLAP), que proporciona uma visão multidimensional dos dados, permitindo a execução de operações analíticas essenciais, como filtragem, agregação, *drill-down* e pivotação (Sherman, 2015).

A eficácia do sistema de BI, no entanto, não se resume apenas à análise dos dados, mas também à sua capacidade de apresentar esses resultados de forma visualmente acessível. Assim, a escolha das abordagens e ferramentas adequadas para a visualização dos resultados obtidos torna-se fundamental, pois maximiza a interpretação e a utilidade da informação. Por isso, Sherman (2015) também inclui na BIAA ferramentas para visualização dos resultados.

A representação dos resultados por meio de gráficos e tabelas é mais eficaz, uma vez que facilita a identificação de padrões e correlações que, de outra forma, seriam difíceis de perceber em formatos exclusivamente tabulares (Sherman, 2015). Essa eficácia na visualização é amplificada com o uso de *dashboards.*

Few (2006) define *dashboard* como uma ferramenta de visualização que exibe informação mais relevante para alcançar um ou mais objetivos de análise, consolidando-a e organizando-a em uma única tela, de forma a permitir uma monitorização eficiente e ágil (Few, 2006). Em um único *dashboard* é possível integrar diversos recursos de visualização, como gráficos, tabelas e textos, para facilitar a interpretação e análise dos dados.

Os *dashboards* distinguem-se de outras ferramentas de análise de dados pela capacidade de apresentar uma visão consolidada da informação mais relevante, permitindo a aplicação de filtros para excluir dados não pertinentes a uma análise específica (Baptista *et al*., 2017). Esta abordagem facilita a interpretação dos dados, assegurando que os utilizadores acedem rapidamente a indicadores essenciais sem se perderem em detalhes secundários.

O sistema de BI a ser desenvolvido na parte prática deste trabalho utiliza *dashboards* com funcionalidades interativas, visando proporcionar uma visão clara e objetiva da informação e uma análise centrada nas métricas de interesse dos utilizadores.

2.4. Fatores críticos de sucesso

A identificação e compreensão dos fatores que permitem às organizações gerir a complexidade, mitigar riscos, superar desafios e garantir a implementação eficaz de sistemas de BI são essenciais para o alinhamento destas soluções com os objetivos estratégicos.

Designados como fatores críticos de sucesso (CSF), estes elementos desempenham um papel determinante no sucesso da implementação de BI nas organizações (Magaireah *et al*., 2017).

Yeoh e Koronios (2010) propõem um *framework* com CSF na conceção e implementação de um sistema de BI, categorizando-os em três dimensões: *Technology, Organization and Process* (TOP *Model*) (Yeoh & Koronios, 2010).

Apesar da criticidade das três dimensões, para Yeoh e Koronios (2010), o sucesso de um sistema de BI é mais influenciado por fatores organizacionais e relacionados a processos do que a fatores tecnológicos e relacionados a dados (Yeoh & Koronios, 2010).

Em um estudo posterior, mas ainda sob a ótica do TOP *Model,* Williams *et al.* (2022) revisaram o modelo de CSF proposto por Yeoh e Koronios (2010). Um resumo sistemático das dimensões e dos fatores críticos de sucesso, proposto por Williams *et al.* (2022), pode ser visto na Tabela 3 (Williams *et al*., 2022).

Tabela 3 - Fatores Críticos de Sucesso.
Fonte: Williams *et al.* (2022) (adaptado).

Perspetiva	Critério	Fatores
I. Tecnológica	**A. Qualidade do sistema**	A.1. Flexibilidade e escalabilidade
		A.2. Integrações
		A.3. Acessibilidade
	B. Qualidade da informação	B.1. Fiabilidade e integridade dos dados
		B.2. Gestão eficaz de dados
	C. Satisfação do utilizador	C.1. Facilidade de utilização
		C.2. Vantagem relativa
II. Organizacional	**A. Apoio da gestão**	A.1. Patrocínio executivo
		A.2. Apoio dos principais intervenientes
	B. Cultura organizacional	B.1. Missão, visão e objetivos bem definidos
		B.2. Cultura de aprendizagem organizacional
		B.3. Cultura corporativa aberta e analítica
	C. Disponibilidade adequada de recursos	C.1. Recursos funcionais e de suporte ao sistema
		C.2. Competências da equipa
		C.3. Centro de competência em BI
III. Processo	**A. Gestão de projetos**	A.1. Abordagem à gestão de projetos
		A.2. Processo de gestão de projetos

	B. Envolvimento dos utilizadores	B.1. Participação e envolvimento dos utilizadores
		B.2. Desenvolvimento iterativo conduzido pelos utilizadores finais
		B.3. Requisitos orientados para o negócio
	C. Gestão da mudança	C.1. Gestão da mudança orientada para os utilizadores
		C.2. Formação e capacitação dos utilizadores
		C.3. Resistência dos utilizadores

Para Williams *et al.* (2022), na dimensão tecnológica, o sucesso na implementação de BI requer um projeto orientado para os negócios, escalável e flexível, com capacidade de expansão e adaptação às mudanças no ambiente e às necessidades dos utilizadores (Williams *et al*., 2022).

Para o sucesso do programa de BI, a dimensão organizacional é ainda mais relevante que a tecnológica e inclui três critérios: apoio da gestão, cultura organizacional e disponibilização adequada de recursos (Williams *et al*., 2022).

Na implementação de BI é necessário um compromisso de todos os envolvidos. É fundamental que os objetivos do projeto de BI estejam alinhados com os objetivos do negócio e que a cultura organizacional seja direcionada para a aprendizagem contínua e para a promoção de tomada de decisão baseada em fatos e resultados analíticos (Williams *et al*., 2022).

O apoio da gestão, com patrocínio executivo e o suporte dos principais interessados, garante que o projeto tenha os recursos, o direcionamento e a colaboração dos *stakeholders*, fatores fundamentais para superar desafios, incertezas e resistência dos utilizadores (Yeoh & Popovič, 2016).

A dimensão de processo é subdividida em três categorias: gestão de projetos, envolvimento do usuário e a gestão de mudanças (Williams *et al*., 2022).

A gestão de projeto envolve as tarefas de elaborar um escopo claro e bem comunicado, definir expectativas e cronogramas realistas e reservar o orçamento apropriado. Essas atividades são realizadas antes mesmo do primeiro dia de implementação e são responsáveis por até noventa porcento da probabilidade de sucesso do projeto (Nasser *et al*., 2018).

A participação e o envolvimento dos utilizadores aumentam as probabilidades de o projeto de BI atender às reais necessidades da organização e agregar efetivo valor ao negócio (Nasser *et al*., 2018).

A implementação de sistemas de BI deve ser considerada um processo evolutivo. E, nesse contexto, a gestão de mudanças reduz a resistência dos envolvidos e facilita a implantação e manutenção do sistema (El-Adaileh & Foster, 2019).

Para que as mudanças sejam eficazes, atinjam os resultados pretendidos e sejam aceites pelos *stakeholders* é essencial a utilização de Indicadores Chave de Desempenho (*KPIs*). Estes indicadores são ferramentas fundamentais para avaliar o estado atual e definir um plano de ação que permita alcançar os objetivos estabelecidos (Varouchas *et al*., 2018).

Alguns estudos propõem ainda a extensão do TOP Model, com a inclusão de uma dimensão (*enviroment dimension*) para representar fatores externos e a pressão competitiva *(*Nasser *et al., 2018)*.

No contexto das instituições de ensino superior, Hasan, *et al*. (2015) destacam a necessidade da inclusão de um fator relacionado às políticas e requisitos educacionais. Como muitas IES são financiadas pelo governo, é necessário que haja uma apresentação contínua de relatórios para garantir o financiamento e a acreditação e o uso de BI pode auxiliar nessa tarefa (Abu Hasan *et al*., 2015).

2.5. Uma introdução ao BI na *Cloud*

Apesar dos benefícios que o uso de BI proporciona às organizações, em alguns casos, os custos e a dificuldade de gestão da infraestrutura necessária para processar e armazenar um grande volume de dados podem desencorajar a adoção dessa tecnologia (ElMalah & Nasr, 2019).

Soluções baseadas em *cloud (nuvem)* têm o potencial de mitigar essas dificuldades, proporcionando maior escalabilidade, flexibilidade e menores custos operacionais (Fernandes *et al*., 2023).

A computação em *cloud* consiste na disponibilização de recursos computacionais à medida através da internet, eliminando a necessidade de infraestrutura local e permitindo um uso mais eficiente da infraestrutura tecnológica (Mell & Grance, 2011).

Os principais benefícios da utilização de soluções de BI em nuvem (*Cloud* BI) são apresentados por ElMalah & Nasr (2019) e apresentados resumidamente na Tabela 4.

Tabela 4 - Benefícios de *Cloud* BI.
Fonte: ElMalah & Nasr (2019) (adaptado).

Benefícios das Soluções em Nuvem	Descrição
Implementação Rápida e Económica	A ausência de infraestruturas locais permite um retorno mais rápido sobre o investimento (ROI).
Ausência de Despesas com Hardware	A redução de custos com infraestrutura resulta num baixo Custo Total de Propriedade (TCO).
Fiabilidade	A computação em nuvem, com sites redundantes, assegura locais seguros e fiáveis para o armazenamento de dados, essenciais para a recuperação de desastres e continuidade dos negócios.
Atualizações e Manutenção Automáticas	O fornecedor da solução é responsável pelas atualizações e manutenção, sem custos adicionais, impactos no tempo ou recursos de TI.
Flexibilidade e Escalabilidade	A nuvem oferece flexibilidade para ajustar a utilização de recursos conforme as necessidades, mantendo os custos controlados.
Pagamento Conforme o Consumo	Os utilizadores pagam apenas pelos recursos efetivamente consumidos, evitando desperdícios e mantendo os custos reduzidos.
Escalabilidade Rápida	As soluções em nuvem permitem uma rápida adaptação à procura crescente, sem necessidade de investimentos em hardware adicional.
Flexibilidade Adaptativa	As soluções em nuvem possibilitam ajustes rápidos, respondendo às necessidades de evolução tecnológica e funcional da organização.

Observa-se, portanto, que adoção de soluções com *Cloud* BI oferece vantagens significativas, como redução de custos, maior flexibilidade e escalabilidade, além de garantir maior fiabilidade e continuidade dos negócios.

Al-Aqrabi *et al*. (2015) esclarecem que computação em *cloud* abrange três formas principais de serviços: *software como serviço*, *plataforma como serviço* e *infraestrutura como serviço* (Al-Aqrabi *et al*., 2015).

- **SaaS (Software como Serviço)**: Disponibiliza aplicações completas, hospedadas na nuvem. Os utilizadores acedem através da internet, sem

necessidade de instalação local. Exemplos comuns incluem serviços de e-mail e ferramentas de produtividade;

- **PaaS (Plataforma como Serviço)**: Oferece uma plataforma sobre a qual os desenvolvedores podem criar, testar e implementar aplicações, sem se preocupar com a gestão da infraestrutura subjacente. Este modelo facilita a construção de soluções de software personalizadas, alicerçadas em ambientes prontos para o uso;

- **IaaS (Infraestrutura como Serviço)**: Proporciona recursos computacionais fundamentais, como servidores, armazenamento e redes, através da nuvem. Este modelo permite às organizações gerir as suas próprias plataformas e aplicações, sem a necessidade de investir em hardware físico, oferecendo flexibilidade e escalabilidade.

A Figura 5, proposta por ElMalah & Nasr (2019), ilustra como os principais fornecedores de aplicações *on-premise* (locais) passaram a oferecer também serviços *on-demand* (na nuvem).

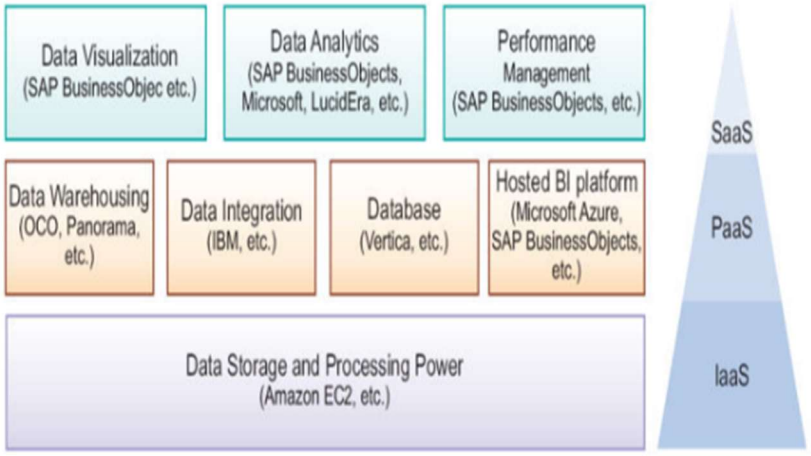


Figura 5 - BI na nuvem.
Fonte: ElMalah & Nasr (2019).

Este movimento, que envolve tanto a criação de novos serviços como a extensão dos produtos já existentes, reflete a adaptação das empresas às novas exigências do mercado, que privilegia soluções mais flexíveis e escaláveis.

Uma solução de *Cloud* BI pode recorrer a qualquer uma das três modalidades de serviço (SaaS, PaaS e IaaS). De forma análoga, qualquer uma das camadas da arquitetura da solução

pode ser adaptada para operar na nuvem, consoante os requisitos específicos e necessidade da organização.

A crescente adoção da computação em nuvem tem impulsionado uma transformação no processo de ETL. Nesse novo contexto, em que o processamento e o armazenamento de dados ocorrem remotamente, o carregamento precede a etapa de transformação, alterando a dinâmica tradicional do fluxo de dados de ETL para Extração-Load-Transformation (ELT) (Hussain & Aradhya, 2024).

No processo de ELT, os dados são inicialmente extraídos (E) das suas fontes e carregados (L), na sua forma bruta e não processada, para uma solução de armazenamento baseada na nuvem. A etapa de transformação (T), que exige elevado poder computacional, é posteriormente realizada na nuvem, tirando partido da escalabilidade e dos recursos de computação à medida, disponibilizados pelas plataformas em nuvem (Yilmaz *et al*., 2020).

A Figura 6 apresenta um comparativo visual entre os processos de ETL e ELT.

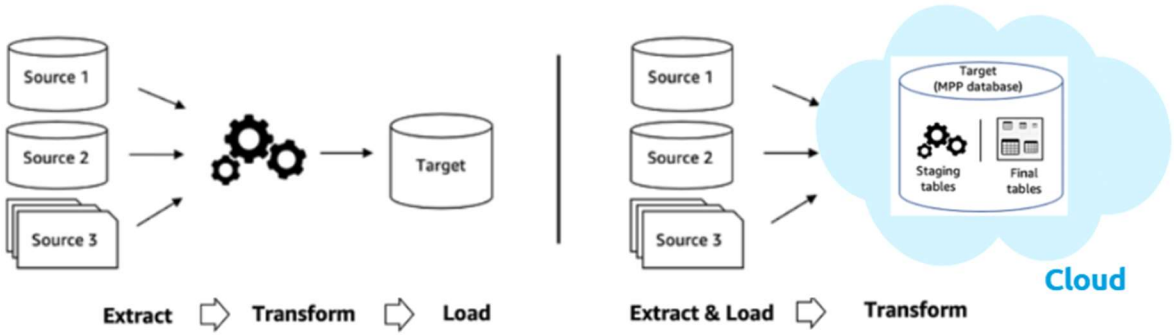


Figura 6 - Comparação ETL vs ELT.
Fonte: Amazon (2023) (adaptado).

A transição para o paradigma ELT oferece algumas vantagens, nomeadamente a escalabilidade, a otimização de custos e uma maior rapidez na obtenção de resultados (Yilmaz *et al*., 2020).

Em um estudo conduzido por Seenivasan (2022) foram comparados os processos de ETL e ELT por meio de um teste prático, utilizando uma base de dados composta por 50 milhões de registos, com o objetivo de avaliar o desempenho, os custos e a escalabilidade das soluções.

Nas simulações, o ELT demonstrou ser mais eficiente, especialmente em cenários que envolvem grandes volumes de dados, apresentando um ganho de aproximadamente 33% em performance (Seenivasan, 2022).

Enquanto o ETL mostrou-se mais adequado para conjuntos de dados pequenos, devido ao seu menor custo inicial e reduzida necessidade de processamento, o ELT revelou-se mais escalável e eficiente para grandes volumes de dados, aproveitando a infraestrutura de nuvem (Seenivasan, 2022).

Apesar dos benefícios, a adoção do ELT pode implicar desafios relacionados com a governança de dados, segurança e conformidade, pois os dados brutos permanecem armazenados na nuvem antes de serem transformados. As organizações devem avaliar cuidadosamente as suas estratégias de gestão de dados e implementar os devidos controlos. (Yilmaz *et al*., 2020).

Os principais fornecedores de soluções de análise e visualização de dados também disponibilizam funcionalidades e ferramentas que permitem a integração das suas plataformas *on-premises* com a *cloud*. Essa integração possibilita que equipas acessem painéis e relatórios de qualquer lugar do mundo e, em tempo real, realizem análises colaborativas.

O Microsoft Power BI, por exemplo, integra-se com o Azure, plataforma de computação em nuvem da Microsoft, proporcionando ao utilizador acesso a funcionalidades avançadas, como escalabilidade dinâmica e armazenamento flexível. Esta integração permite o processamento grandes volumes de dados e a realização de análises complexas em tempo real (Microsoft, 2023).

A Salesforce disponibiliza o Tableau Online, uma ferramenta de visualização de dados em nuvem que se articula com a sua plataforma de CRM, permitindo a análise e exploração de dados sem necessidade de infraestrutura local (Tableau, 2025).

Por sua vez, a Amazon, através da AWS, oferece o QuickSight, uma solução de BI que possibilita a análise e visualização de dados em tempo real. A compatibilidade com serviços como o Amazon S3, sistema de armazenamento escalável na nuvem, e o Amazon RDS, serviço de bases de dados relacionais, permite a utilização integrada dos dados armazenados, proporcionando um ambiente unificado para processamento e análise (Amazon, 2025).

A utilização de *Cloud BI* ainda encontra resistência, principalmente por questões relacionadas a ameaças de segurança e perda do controlo absoluto sobre a solução e os dados (Fernandes *et al*., 2023), embora, frequentemente, as soluções em nuvem ofereçam níveis de proteção superiores aos sistemas locais (Mondal, 2022).

Outros desafios importantes são a dificuldade de integração entre os silos de dados (locais e remotos), a necessidade de conexões com a internet de alta velocidade e estabilidade e restrições impostas pelos dispositivos de segurança das organizações (Mondal, 2022).

2.6. *Business Intelligence* no ensino superior

Como qualquer organização, as IES também têm objetivos estratégicos que precisam ser alcançados. Nesse contexto, a complexa administração de estudantes, recursos humanos e finanças impõe um desafio significativo aos gestores das IES (Nazri *et al*., 2020).

A implementação de sistemas de BI em IES fundamenta-se em cinco pilares (Sorour *et al*., 2020):

- **Tecnologia**: Infraestrutura adequada e metodologias avançadas de análise de dados;
- **Organização**: Estruturação, liderança e gestão eficaz dos processos institucionais;
- **Ambiente**: Cumprimento de normativas internas e externas, bem como adaptação a desafios externos;
- **Negócio**: Otimização de recursos financeiros e operacionais para sustentabilidade e competitividade;
- **Social**: Influência de fatores humanos e dinâmicas institucionais na aceitação dos sistemas.

A integração e a monitorização desses fatores, com sistemas de BI, são essenciais para que as IES possam aprimorar continuamente seus processos e tomar decisões mais informadas e eficazes.

A avaliação da qualidade do ensino/aprendizagem e a análise da retenção de estudantes constituem os principais desafios que as IES procuram resolver ao implementar soluções de BI (Villegas-Ch *et al*., 2020).

No entanto, a aplicação de BI não se restringe a estas áreas. Apraxine e Stylianou (2017) apresentam casos práticos em que o uso de sistemas de BI promove a melhora da comunicação interdepartamental, a previsão de picos de procura, a otimização dos processos de recrutamento/matrícula e o aprimoramento da gestão de recursos, proporcionando às respetivas instituições uma administração mais eficiente (Apraxine & Stylianou, 2017).

Segundo AlKhatnai e Shawyun (2022), as IES, frequentemente, utilizam inquéritos de perceção de satisfação para avaliar o desempenho dos seus serviços. Estes estudos, quando aliados a sistemas de BI, possibilitam a integração de dados provenientes de diferentes áreas, convertendo-os em informação estruturada, que promove decisões estratégicas mais fundamentadas (AlKhatnai & Shawyun, 2022).

Morais e Lopes (2019) descrevem a implementação de uma solução de BI na Universidade Portucalense, uma IES portuguesa. Esse projeto teve como principal objetivo apoiar o sistema de qualidade institucional e melhorar a estratégia futura, com foco na área da Ensino-Aprendizado, e foi estruturado em oito etapas fundamentais (Morais & Castro Lopes, 2019):

Tabela 5 - Etapas fundamentais do projeto BI - Universidade Portucalense.
Fonte: Morais & Castro Lopes (2019) (adaptado).

1. Análise dos Processos	Análise dos principais processos que suportam o vetor Ensino e Aprendizagem
2. Análise dos Indicadores	Identificação e associação de KPIs aos processos através da análise de documentos internos (planos de desempenho, decretos, etc.).
3. Revisão e Definição dos KPIs	Atualização dos KPIs com base na revisão da literatura, objetivos da IES e experiência dos autores.
4. Validação dos KPIs	Validação dos KPIs com gestores, resultando na introdução de novos KPIs e ajustes.
5. Identificação dos Sistemas de Informação	Inventário dos sistemas de informação da instituição
6. Análise das Fontes de Informação	Identificação dos sistemas que alimentam os KPIs, em colaboração com o diretor de sistemas de informação.
7. Definição do Perfil de Acesso	Definição dos responsáveis pelos processos e respetivos perfis de acesso aos sistemas de BI, conforme o sistema interno.

8. Seleção e Implementação do BI	Seleção do Qlik Sense como solução de BI com base no quadrante mágico da Gartner e conhecimento da equipe de desenvolvimento. Painéis já desenvolvidos e apresentados ao gestor.

Como resultado, o sistema de BI implementado na Universidade Portucalense proporcionou uma visão mais abrangente e detalhada da atividade docente, auxiliou na tomada de decisões estratégicas e assegurou a sustentabilidade e a melhoria contínua do sistema de qualidade, em consonância com os princípios da gestão estratégica e da avaliação institucional (Morais & Castro Lopes, 2019).

Júnior *et al*. (2022) realizaram um estudo numa universidade pública brasileira com o objetivo de identificar os fatores determinantes da evasão (abandono) académica, recorrendo à análise de dados históricos da instituição por meio de um sistema de BI.

Os resultados demonstraram que a maioria dos estudantes que abandonam os cursos têm idade superior a 25 anos e apresentam um desempenho académico insatisfatório, particularmente nas disciplinas iniciais dos cursos. A evasão é mais pronunciada nos cursos da área de ciências humanas, embora alguns cursos da área de ciências exatas registem taxas de evasão próximas de 50%. Adicionalmente, verificou-se que alguns cursos apresentam uma taxa de conclusão inferior a 20% dos estudantes ingressantes, o que evidencia a necessidade de implementação de medidas estratégicas para mitigar o impacto da evasão no ensino superior (Júnior *et al*., 2022).

O estudo de Júnior *et al*. (2022) sublinha a relevância do desenvolvimento de políticas institucionais direcionadas para a retenção e o sucesso académico dos estudantes, com especial enfoque nos grupos e áreas identificados pelo sistema de BI.

Boulila et al. (2023) descreve a implementação de uma solução de BI para apoiar a gestão académica na Universidade de Taibah, Arábia Saudita. A análise detalhada dos dados institucionais permitiu identificar padrões recorrentes, fornecendo indicadores chave, como o número de estudantes por curso e departamento, médias de desempenho académico segmentadas por género e nacionalidade, e a identificação de disciplinas com elevadas taxas de insucesso (Boulila *et al*., 2023).

Os *insights* gerados pelo sistema proporcionaram melhor compreensão de questões académicas, como a eliminação frequente de determinadas disciplinas, as altas taxas de

insucesso em certos cursos, os atrasos no cumprimento dos planos de estudo pelos estudantes e auxiliaram na definição de indicadores para a seleção de disciplinas que podem contribuir para a melhoria do desempenho académico (Boulila *et al*., 2023).

A integração de BI no processo académico demonstrou ser uma ferramenta eficaz para superar as limitações do modelo tradicional, oferecendo soluções concretas para problemas recorrentes nas práticas académicas e permitindo uma gestão mais eficiente (Boulila *et al*., 2023).

Os estudos evidenciam que a implementação de BI nas IES desempenha um papel fundamental na otimização dos processos académicos e no suporte à tomada de decisão estratégica.

A utilização de BI não apenas facilita o acompanhamento do desempenho académico e administrativo, como também contribui para o desenvolvimento de políticas institucionais orientadas para a retenção estudantil e a melhoria contínua da qualidade do ensino.

2.7. Ferramentas e tecnologias

A Gartner, empresa especializada em pesquisa e consultoria, através do resultado de análise de mercado, avalia ferramentas e elabora relatórios sobre tendências na área de tecnologia de informação.

Um dos recursos utilizado pela Gartner é o quadrante mágico, através do qual é apresentada uma visão ampla sobre posições relativas de concorrentes e produtos em determinada área de atuação (Schlegel *et al*., 2024).

No quadrante mágico sobre *Business Intelligence e Analitycs,* as ferramentas/soluções são avaliadas utilizando-se doze critérios, incluindo *Automated insights, Data source connectivitye Data visualization.*

Em seguida, de acordo com os resultados obtidos, os fornecedores/desenvolvedores são classificados em quatro categorias (Schlegel *et al*., 2024), que estão descritas abaixo:

- **Leaders**: executam bem em relação à sua visão atual e estão bem-posicionadas para o futuro;

- **Visionaries**: entendem para onde o mercado está indo ou têm uma visão para mudar as regras do mercado, mas não executam bem;
- **Niche Players**: concentram-se com sucesso em um pequeno segmento ou não se concentram e não superam nem inovam mais do que os outros;
- **Challengers:** executam bem hoje ou podem dominar um segmento grande, mas não demonstram uma compreensão da direção do mercado.

O quadrante mágico de *Business Intelligence and Analitycs (BI and Analitycs)* do ano de 2024 é apresentado a seguir.

Figura 7 - Quadrante Mágico BI *and Analitycs*.
Fonte: Schlegel *et al*. (2024).

Microsoft e Salesforce (Tableau) destacam-se no quadrante de *Leaders*.

A Salesforce (Tableau) estreou no quadrante mágico de BI da Gartner no ano de 2010 e até o ano de 2012 figurava entre as c*hallengers*. No ano de 2013 apareceu pela primeira vez como *leader* e, desde então, não alterou essa classificação.

A Gartner destaca que o Tableau é uma ferramenta direcionada à exploração baseada em visualização, que atende consumidores de todos os tamanhos, permitindo acesso, preparação, análise de dados além de apresentação de resultados.

A Microsoft integra o quadrante de *Leaders* de *BI and Analitycs* da Gartner desde o ano de 2008. Sendo que, no ano de 2024, o *Microsoft PowerBI* foi a plataforma de BI mais utilizada no mundo (Manis, 2024).

A inclusão da ferramenta de BI no pacote Office 365, a crescente integração com a plataforma de comunicação e colaboração Teams, a ampliação do serviço de nuvem e a possibilidade de integrar o trabalho de toda a equipa (como engenheiros de dados, profissionais de data warehousing e cientistas de dados), através do da ferramenta *Fabric,* foram os grandes responsáveis por essa expansão massiva (Manis, 2024).

O *PowerBI* é uma ferramenta *low-code* e de *self-service* que permite a conexão com diversas fontes de dados, a transformação e modelagem desses dados e o desenvolvimento de painéis personalizados.

A interface, similar a outros produtos do Office 365, proporciona que utilizadores da suíte façam uma transição suave entre as ferramentas, com uma rápida curva de aprendizagem. Permite ainda uma colaboração aprimorada entre equipas, com compartilhamento de relatórios e *dashboards*.

Além de ser líder no segmento de *BI e Analitycs*, a *Microsoft* estabeleceu um protocolo com o IPL para disponibilizar ferramentas educacionais aos estudantes, destacando o Office 365, que, recentemente, incorporou a plataforma de BI, *PowerBI*.

Diante dessa realidade, para a realização do projeto prático serão utilizadas tecnologias e ferramentas da Microsoft, sobretudo na etapa de visualização de resultados, que será realizada em *PowerBI*.

3. Conceção do sistema de *Business Intelligence*

No capítulo anterior, foi realizado um estudo prévio sobre os principais conceitos e práticas no desenvolvimento de sistemas de BI. No presente, são apresentadas as etapas para a conceção do sistema aplicado ao caso de estudo.

A conceção de um sistema de BI envolve a definição da arquitetura do sistema, a identificação das necessidades organizacionais (requisitos), a elaboração do modelo dimensional e a seleção das tecnologias apropriadas. Nas secções seguintes, proceder-se-á à análise pormenorizada de cada uma das etapas referidas.

3.1. Arquitetura do sistema

Conforme apresentado no capítulo de revisão da literatura, um sistema de BI caracteriza-se por uma estrutura em múltiplas camadas e componentes, que englobam as fontes de dados, a integração (ETL), o DW e as ferramentas de análise e visualização.

Por isso, a definição da arquitetura constitui uma das etapas-chave no processo de conceção e desenvolvimento de um projeto de BI, dado que todo o funcionamento do sistema está diretamente condicionado pelas diretrizes estabelecidas nesse projeto.

Uma arquitetura bem delineada assegura que os dados sejam organizados de forma coerente, acessível e fiável, proporcionando uma visão integrada e detalhada da informação estratégica necessária.

Na conceção do sistema de BI deste projeto, a arquitetura foi concebida para abranger todas as etapas essenciais do processo e está representada na Figura 8.

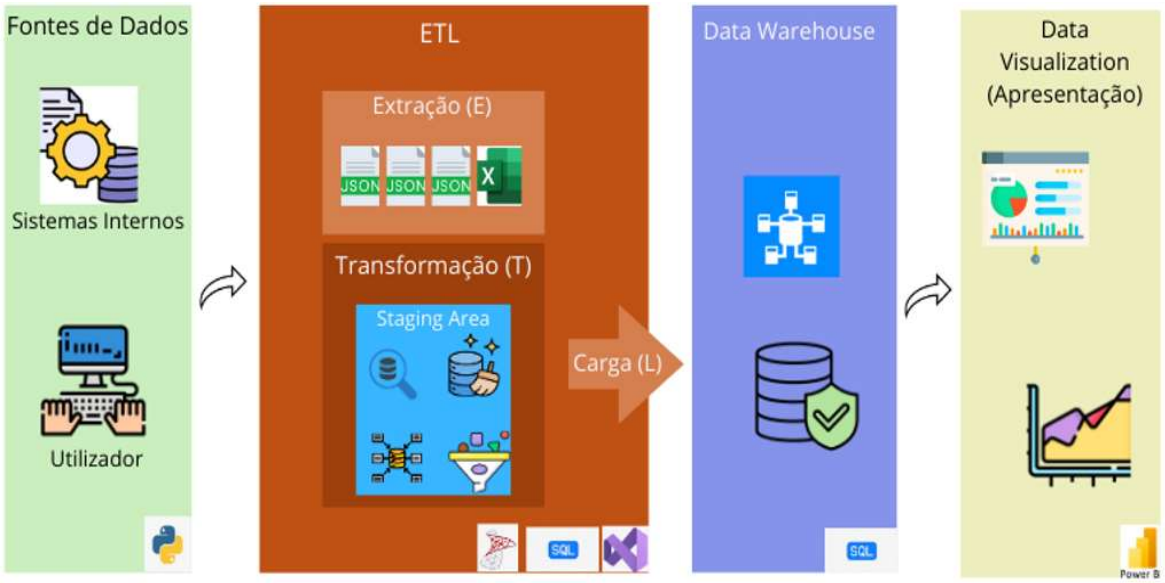


Figura 8 - Arquitetura do sistema proposto.

O processo inicia-se com a identificação das fontes de dados (*Data Sources*), que inclui o acesso aos sistemas internos do IPL, através de API, e ficheiros externos fornecidos pelos *stakeholders* do projeto.

A etapa subsequente envolve o processo de ETL, fundamental para a integração dos dados. Na subfase de Extração, os dados sobre a assiduidade nas aulas são recolhidos a partir do consumo de APIs, que fazem o acesso a subsistemas do sistema académico do IPL e devolvem os dados no formato *json* (*javascript object notation*). Enquanto ficheiros externos que possuem dados dos cursos, docentes e unidades curriculares são fornecidos pelos *stakeholders* em formato XLSX (ficheiros *Excel*).

Adicionalmente, por meio de um *script* desenvolvido em *Python*, o utilizador pode gerar dados alusivos a datas especiais e alterar as datas de início e fim dos anos e semestres letivos. Esses dados são armazenados em ficheiros *json* independentes.

Os dados extraídos das fontes supramencionadas são então transferidos para uma *Staging Area*, um ambiente temporário onde ocorrerá a sua transformação. A Figura 9 representa graficamente o processo de Extração.

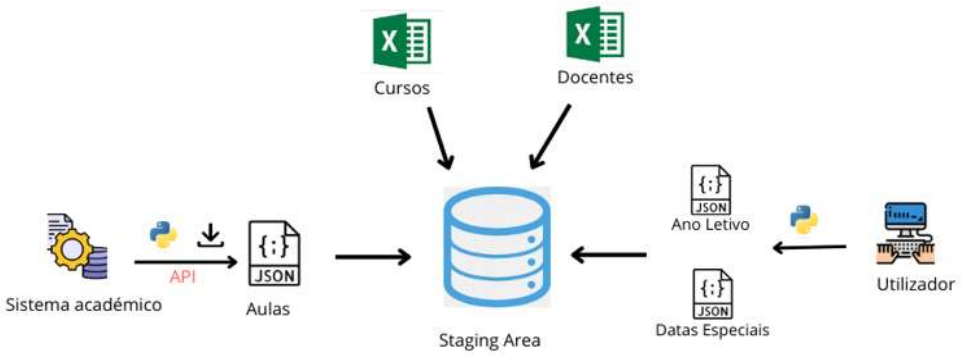


Figura 9 - Extração dos dados.

Na subfase seguinte, de transformação, os dados são submetidos a processos de limpeza, padronização e integração, de modo a garantir a sua qualidade e consistência para o carregamento posterior no DW.

O DW serve como repositório central estruturado, proporcionando uma visão integrada dos dados e permitindo agilizar a sua consulta e a análise.

Finalmente, na fase de análise e visualização, a informação é disponibilizada através de *dashboards* interativos, que oferecem a possibilidade de uma análise em ambiente visual.

Estes *dashboards* têm como objetivo facultar a capacidade de proporcionar *insights* valiosos, apoiando a tomada de decisões estratégicas com base em evidencias quantitativas.

3.2. Requisitos funcionais

O sistema a ser desenvolvido deve fornecer respostas às questões relacionadas à assiduidade dos alunos ao longo do semestre letivo, permitindo uma análise detalhada e fundamentada deste indicador académico.

Para orientar o processo de desenvolvimento, foi elaborada uma lista preliminar com os requisitos mínimos que o sistema de BI deve atender. Esses requisitos devem ser respondidos para cada um dos semestres dos respetivos anos letivos.

A lista de requisitos iniciais é apresentada na tabela a seguir.

Tabela 6 – Requisitos.

ID	**Requisito**
Q1	Num determinado período, qual o número total de docentes que estão a lecionar aulas?
Q2	Num determinado período, qual o número de docentes que leciona aulas a cada ciclo de estudos (TeSP, licenciatura, mestrado)?
Q3	Qual a carga horária média de aulas dos docentes?
Q4	Qual a quantidade média de aulas por docente?
Q5	Quais docentes possuem a maior carga horária?
Q6	Quais docentes possuem o maior número de aulas?
Q7	Qual a média de assiduidade por aula por tipologia (P, TP, T, entre outras)?
Q8	Qual a assiduidade média por turno?
Q9	Qual a média acumulada de assiduidade para cada um dos turnos até as semanas críticas (5 e 10)?
Q10	Qual o mínimo de assiduidade para cada um dos turnos até as semanas críticas?
Q11	Qual o máximo de assiduidade para cada um dos turnos até as semanas críticas?
Q12	Qual a mediana de assiduidade para cada um dos turnos até as semanas críticas?
Q13	Como evolui a assiduidade de cada turno ao longo das semanas letivas do semestre?

A partir das questões preliminares foram elaborados os requisitos funcionais do sistema, que são descrições das funcionalidades mínimas que o sistema deve oferecer aos seus utilizadores.

A lista de requisitos funcionais encontra-se a seguir listada na Tabela 7.

Tabela 7 - Requisitos Funcionais.

ID	**Breve descrição do requisito**	**Prioridade**
RF01	O sistema deve apresentar o número de docentes que ministram aulas num determinado semestre.	Média
RF02	O sistema deve determinar o número de docentes por ciclo de estudo (mestrado, licenciatura, TeSP).	Média
RF03	O sistema deve calcular a carga horária média de aulas dos docentes.	Média

RF04	O sistema deve calcular a quantidade média de aulas ministradas por docente.	Média
RF05	O sistema deve identificar os docentes com a maior carga horária de aulas.	Média
RF06	O sistema deve identificar os docentes com o maior número de aulas ministradas.	Média
RF07	O sistema deve calcular a média de assiduidade por UC e por componente.	Alta
RF08	O sistema deve determinar a assiduidade média por turno.	Alta
RF09	O sistema deve calcular a média acumulada de assiduidade para cada turno até as semanas críticas (semana 5 e semana 10).	Alta
RF10	O sistema deve determinar o mínimo de assiduidade para cada turno até as semanas críticas.	Alta
RF11	O sistema deve determinar o máximo de assiduidade para cada turno até as semanas críticas.	Alta
RF12	O sistema deve calcular a mediana de assiduidade para cada turno até as semanas críticas.	Alta
RF13	O sistema deve apresentar a evolução da assiduidade de cada turno ao longo das semanas letivas do semestre.	Alta
RF14	O sistema deve agilizar a navegação pelos dados das unidades curriculares de um curso permitindo realizar operações de *Drill-Down, Slice* e *Dice* de forma a observar para cada um dos seus turnos a evolução semanal das presenças em sala aula e assim conseguir comparar e observar a evolução das presenças	Alta
RF15	O sistema deve agilizar a navegação pelos dados permitindo realizar operações de *Slice* e *Dice* utilizando atributos da hierarquia data da dimensão data.	Alta

3.3. Identificação das fontes de dados

Para responder às necessidades identificadas e às questões preliminares, foi reconhecida a necessidade de proceder à recolha de dados das seguintes entidades: docentes, aulas, cursos e unidades curriculares.

As fontes de dados utilizadas provêm, predominantemente, de sistemas internos da ESTG/IPL.

A comunicação entre o sistema de BI e os sistemas internos da instituição é assegurada através de interfaces previamente estabelecidas. Além disso, o sistema de BI também será alimentado por ficheiros previamente disponibilizados pelos gestores.

Estes ficheiros seguem formatos específicos e são colocados em pastas dedicadas de onde os dados são carregados.

Com o objetivo de assegurar a atualização de datas sensíveis relacionadas com o calendário letivo, que é volátil e em que a sua definição depende dos órgãos da escola, é desenvolvido um *script* em *Python* que permite a sua definição pelos utilizadores, garantindo flexibilidade e autonomia no seu processo de gestão.

3.4. Análise preliminar dos dados

Antes do efetivo desenvolvimento do sistema foi realizada uma análise preliminar dos dados iniciais, fornecidos pela Coordenadora do Departamento de Engenharia Informática da ESTG/IPL.

O principal objetivo dessa fase é verificar a qualidade, a consistência e a relevância dos dados existentes e a possibilidade efetiva de resolução do problema.

3.4.1. Problemática de inconsistência na extração de dados

O IPL possui um sistema académico que regista os dados sobre cada uma das aulas previamente agendadas ou programadas.

Inicialmente foram disponibilizadas duas opções para a obtenção dos dados de aulas do IPL — uma através de um ficheiro *excel (*planilha*,* extraída manualmente do sistema académico)

e outra pela obtenção direta dos dados usando uma API desenvolvida por um técnico superior da instituição, que devolve um ficheiro *json*. Os dados extraídos por ambos os métodos foram comparados e identificou-se uma inconsistência na quantidade de registos.

Para o período inicial de análise, compreendido entre 01/11/2022 e 17/12/2022, o ficheiro em formato *json* possui mais que o dobro de registos da planilha.

No início, verificou-se que a planilha continha apenas dados relativos às aulas realizadas na ESTG, o que indicava a utilização de uma filtragem anterior. Enquanto o ficheiro *json*, devolvido pela API, continha dados das aulas realizadas em todas as unidades orgânicas da instituição.

Contudo, ao aplicar o mesmo filtro nos dados obtidos pela API, para considerar apenas as aulas da ESTG, o número de registos reduziu-se de forma tão significativa que ficou inferior ao registado na planilha, indicando assim uma nova e inesperada inconsistência.

Mais uma vez, os dados foram analisados e identificou-se que a API, aparentemente, trazia dados parciais. O problema foi diagnosticado: para os dados obtidos pela API cada registo possuía um identificador único, *aula_id*. Entretanto, nos dados obtidos no ficheiro *excel* esse identificador não era único. Nos dados da planilha havia registos com *aulas_id* iguais, mas com dados de UCs diferentes.

A Figura 10 e Figura 11 ilustram a inconsistência identificada. Ao filtrar os dados pela data/hora '14-12-2022 11:00:00' e por um código de docente específico (dados do docente anonimizados), o ficheiro obtido através da interface (Figura 10) apresenta apenas um registo: a aula com *aula_id* 1359002, referente a uma UC do 'Curso Técnico Superior Profissional de Sistemas ...'.

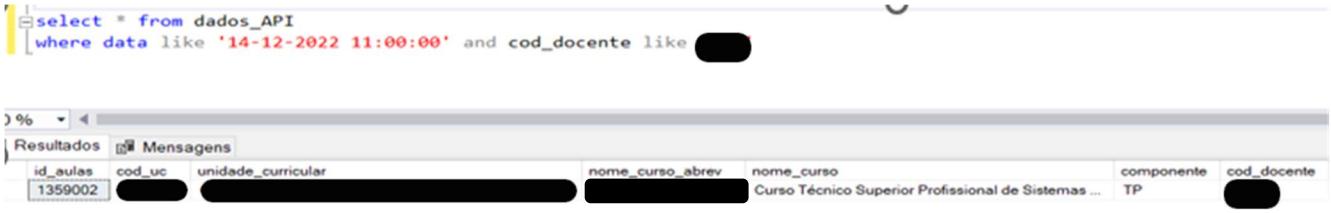


Figura 10 - Interface: dados da aula do dia 14-12-2022 11:00:00, dados anonimizados.

Contudo, ao aplicar o mesmo filtro nos dados do ficheiro foram obtidos dois registos (Figura 11).

1359002	[redacted]	Curso Técnico Superior Profissional de Fabricação Automática	TP	[redacted]
1359002		Curso Técnico Superior Profissional de Sistemas Eletromecânicos	TP	

Figura 11 - Planilha: dados da aula do dia 14-12-2022 11:00:00, dados anonimizados.

Além do registo devolvido pela API, foi identificado outro registo, com o mesmo identificador da aula (*aula_id)*, mesmo docente e mesma data e hora do primeiro, mas associado a uma UC e a um curso distinto. Esse segundo registo não existia nos dados obtidos com o auxílio da API.

Com o objetivo de compreender a dinâmica de funcionamento da API, que recolhia os dados diretamente do sistema académico interno da instituição, foi realizada uma reunião de esclarecimento com o técnico responsável pelo desenvolvimento da ferramenta.

Após a apresentação do problema, o técnico esclareceu que a interface desenvolvida tratava o campo *aula_id* como uma chave primária. Em virtude dessa configuração, a interface, naquele momento, era incapaz de recuperar registos distintos que partilhassem o mesmo identificador da aula.

Procedeu-se, então, a uma nova análise dos dados para compreensão das razões pelas quais registos distintos partilhavam o mesmo identificador e determinar se esse comportamento, identificado exclusivamente nos dados do ficheiro *Excel*, correspondia a uma característica desejável ou a uma anomalia.

A comparação entre os dados revelou que a existência de registos com o mesmo identificador de aula ocorria devido à organização em grupos de disciplinas. Nesses casos específicos, dois ou mais turnos de UCs distintas são agrupados para partilharem uma mesma aula, como estratégia de otimização de recursos.

Nessas situações, a aula é única (*mesmo aula_id*), pois é ministrada pelo mesmo docente, na mesma data e hora. Contudo, para efeitos de controlo dos turnos das disciplinas, cada turno possui um registo distinto da aula, com os respetivos dados de unidade curricular, curso, turno e número de alunos presentes.

Dessa forma, comprovou-se que o campo *aula_id* não poderia ser utilizado isoladamente como chave primária, uma vez que a existência de registos com o mesmo *aula_id* era uma característica esperada do sistema académico.

Uma vez identificada a origem do problema, o responsável pelo desenvolvimento da interface procedeu com as alterações necessárias para garantir que os dados fossem recolhidos de maneira adequada junto ao sistema académico.

A Figura 12 apresenta o resultado da consulta ao ficheiro *json*, devolvido pela API, após a resolução do problema.

	id_aulas	cod_uc	unidade_curricular	nome_curso_abrev	nome_curso	componente	cod_docente	turma	motivo_falta	estado
1	1359002	[redacted]		Sist Eletromecânicos	Curso Técnico Superior Profissional de Sistemas ...	TP	[redacted]	TESTG1D		Fechado
2	1359002			Fabricação Autom	Curso Técnico Superior Profissional de Fabricaçã...	TP		TESTG1D		Fechado

Figura 12 - Resultado interface após ajuste, dados anonimizados.

Para os novos dados, diante da impossibilidade de utilizar o atributo *aula_id* como chave, foi introduzido um novo atributo, designado *aula_id_curso*, resultante da concatenação do identificador *aula_id* com o código do curso. Este atributo passou a ser utilizado como chave primária dos registos, assegurando a unicidade e a consistência dos dados.

3.4.2. Análise exploratória dos dados iniciais das aulas

A análise exploratória de dados (AED) é uma etapa fundamental no processo de investigação científica e análise estatística. A AED representa uma forma de analisar e investigar um conjunto de dados, resumindo suas principais características, possibilitando uma compreensão e adequada interpretação dos dados (Costa, 2024).

Através de métodos estatísticos e ferramentas gráficas, a AED permite compreender a estrutura dos dados, identificar padrões, tendências e anomalias, detetar relações entre as variáveis e auxiliar na formulação de hipóteses relacionadas aos dados.

O ficheiro *json* com os dados preliminares sobre as aulas (devolvidos pela API), no intervalo de 01/11/2022 a 17/12/2022, foi submetido à AED para melhor compreensão dos dados disponibilizados. A análise foi realizada em *Python* e o *script* completo está disponível no Anexo A – *Script* análise exploratória *(Python)*.

O ficheiro possui um total de 29.732 registos, sendo que 12.900 registos são de aulas da ESTG. Foi aplicado um filtro com o objetivo de restringir a análise exclusivamente às aulas da ESTG.

Os dados estão estruturados em dezanove colunas (*id_aulas, cod_uc, unidade_curricular, curso, nome_curso, componente, cod_docente, nome_docente, turma, motivo_falta, estado, data, data_inicio, data_fim, carga, grupo_disciplinar, turno, n_alunos, escola*).

A colunas id*_aulas, cod_uc, cod_docente e n_alunos* estão em formato numérico, enquanto as demais são do tipo *object,* que permite armazenar objetos *Python*, dados no formato texto etc. A estrutura dos dados é apresentada na Figura 13.

```
RangeIndex: 12900 entries, 0 to 12899
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id_aulas            12900 non-null  int64
 1   cod_uc              12626 non-null  object
 2   unidade_curricular  12900 non-null  object
 3   curso               12900 non-null  object
 4   nome_curso          12900 non-null  object
 5   componente          12626 non-null  object
 6   cod_docente         12900 non-null  int64
 7   nome_docente        12900 non-null  object
 8   turma               12900 non-null  object
 9   motivo_falta        1931 non-null   object
 10  estado              12900 non-null  object
 11  data                12900 non-null  object
 12  data_inicio         12900 non-null  object
 13  data_fim            12900 non-null  object
 14  carga               12900 non-null  object
 15  grupo_disciplinar   1933 non-null   object
 16  turno               12900 non-null  object
 17  n_alunos            12900 non-null  int64
 18  escola              12900 non-null  object
```

Figura 13 – Análise exploratória- Estrutura de dados.

Apenas as colunas *cod_uc, componente, motivo_falta, e grupo disciplinar* possuem valores/nulos/ausentes, conforme pode ser observado na Figura 14. Além disso, não foram encontrados registos duplicados nos dados disponibilizados.

```
Nulos/Ausentes:
id_aulas                    0
cod_uc                    274
unidade_curricular          0
curso                       0
nome_curso                  0
componente                274
cod_docente                 0
nome_docente                0
turma                       0
motivo_falta            10969
estado                      0
data                        0
data_inicio                 0
data_fim                    0
carga                       0
grupo_disciplinar       10967
turno                       0
n_alunos                    0
escola                      0
```

Figura 14 - Análise exploratória - Valores nulos/ausentes.

Apesar dos 12.900 registos, analisando a unicidade de *id_aulas* indica que os registos referem-se a 12.065 aulas. Conclui-se, portanto, que há registos de aulas para grupos de disciplinas.

Relativamente à análise da assiduidade, verificou-se que 2.287 registos apresentam n_alunos iguais a zero. Destes, 1.919 registos possuem o atributo motivo_falta preenchido, portanto, o valor 0 é justificado, por exemplo, devido a um dia feriado ou a um evento académico em que efetivamente não ocorreram aulas. Enquanto, 368 registos possuem n_alunos iguais a zero e não há motivação aparente para isso.

Nos dados analisados, o número médio de alunos por registo foi de 15, com uma mediana de 14 alunos, refletindo uma distribuição ligeiramente assimétrica. O *boxplot* do número de alunos por registo, Figura 15, indica que existem registos com valores de assiduidade que estão significativamente afastados da distribuição geral, caracterizando-se como *outliers*.

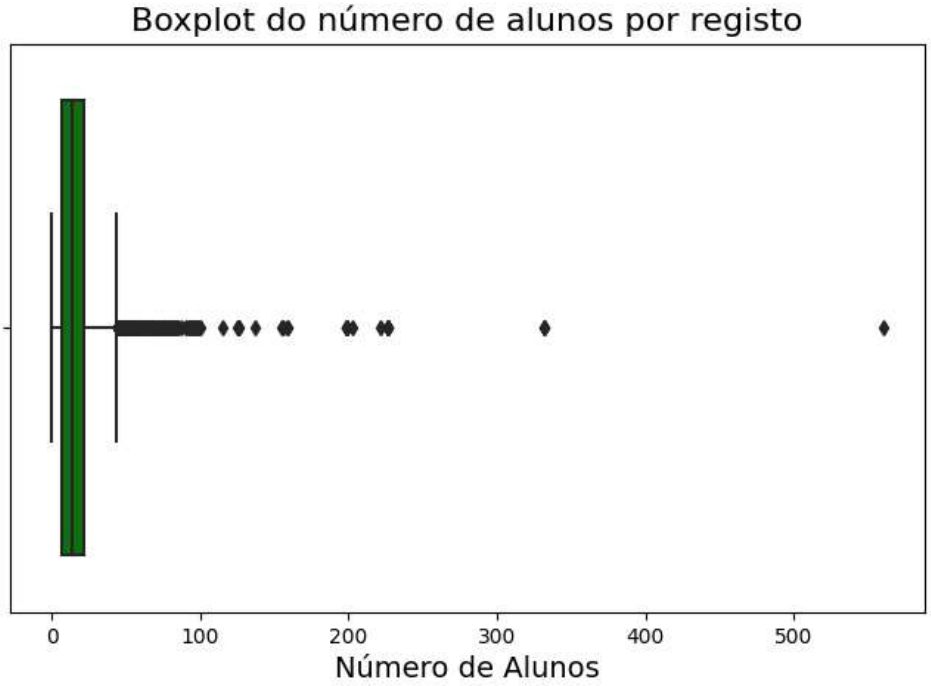


Figura 15 – Análise exploratória - Boxplot número de alunos por registo.

Existem 26 registos de aulas com assiduidade igual ou superior a 100 alunos, e a aula com maior assiduidade foi de uma UC do curso Licenciatura em Engenharia Eletrotécnica, com número de presenças de 560 alunos, conforme observa-se na Figura 16.

Figura 16 - Análise exploratória - Registo com maior n_alunos.

Como a inserção do número de estudantes em cada uma das aulas é feita manualmente pelo docente, é possível que ocorram erros no momento do preenchimento.

O conjunto de dados apresenta 441 códigos únicos de docentes, com a variável *cod_docente* a variar entre 4 e 4682. Este intervalo confirma a inexistência de valores negativos ou iguais a zero, o que sugere, numa análise preliminar, a ausência de anomalias.

No período analisado, cada docente ministrou em média 27.36 aulas, cada aula teve uma duração média de 2.28 horas de carga horária.

Foram encontrados 1.523 turnos distribuídos em 780 unidades curriculares, referentes a 67 cursos distintos.

A AED realizada permitiu conhecer melhor a estrutura dos dados disponíveis e como eles estão organizados, o que é fundamental para a conceção do modelo dimensional do projeto.

3.5. Conceção do modelo dimensional

Como visto na revisão de literatura, o modelo dimensional é uma abordagem amplamente utilizada na conceção de sistemas de DW, sendo especialmente adequado para suportar análises e consultas complexas de dados.

Estruturado em torno de tabelas fato e dimensões, este modelo deve organizar os dados de forma a facilitar a exploração e a interpretação da informação, facilitando a resposta às questões de análise.

A simplicidade e eficiência do modelo dimensional tornam-no ideal para representar cenários reais.

No âmbito do presente projeto, o objeto de estudo centra-se na análise da assiduidade dos estudantes em contexto letivo. Por isso, definiu-se o evento *aula* como tabela facto do modelo dimensional, na qual se registam, de forma sistemática, dados sobre as presenças dos alunos.

A avaliação da assiduidade será conduzida a partir de diferentes eixos de análise, nomeadamente as UCs, o corpo docente, o ano letivo, o grupo disciplinar e o curso. Para cada uma desses eixos, foram estabelecidas dimensões analíticas, garantindo uma abordagem multidimensional e organizada.

Esta estrutura metodológica permite uma análise estruturada e da evolução da assiduidade ao longo dos semestres e anos letivos. A Figura 17 ilustra o modelo dimensional em formato de estrela *(star schema)* do sistema de BI a ser desenvolvido.

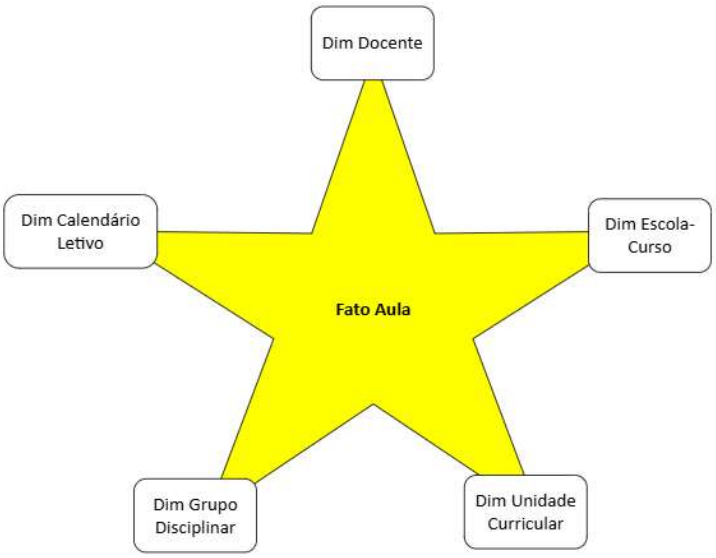


Figura 17 - Modelo dimensional simplificado (*Star Schema*).

A tabela fato (fato_aula) ocupa a posição central e relaciona-se diretamente com cada uma das cinco dimensões (docente, escola-curso, unidade curricular, grupo disciplinar e calendário letivo).

Ao longo do tempo, prevê-se um aumento substancial no número de registos na tabela fato (fato_aula), ao passo que as tabelas de dimensões, em geral, possuem um maior número de registos na primeira carga do sistema, nas cargas seguintes não apresentam um crescimento elevado.

3.5.1. Tabelas fato e dimensões

De acordo com o modelo dimensional, foram concebidas tabela fato e tabelas de dimensões com o propósito de dar resposta aos requisitos previamente estabelecidos.

Tabela Fato (fato_aula)

A *fato-aula* é composta maioritariamente por medidas e por chaves estrangeiras, e é com base nela que são feitas agregações e somas.

Os atributos *dim_docente_key, dim_escola_curso_key, dim_grupo_uc_key* e *dim_uc_componente_turno_key, dim_grupo_uc_key* são chaves estrangeiras para as dimensões docente, escola_curso, unidade curricular (Uc_componente_turno) e grupo de unidade curricular (grupo_uc), respetivamente.

O atributo *data_aula* estabelece a relação entre os registos de assiduidade e o calendário letivo, permitindo análises temporais alinhadas com a estrutura académica da instituição.

O atributo *n_aluno* regista a assiduidade dos alunos nas aulas, enquanto o atributo *motivo_falta* descreve a razão pela qual a aula não ocorreu em dias específicos.

O sistema académico regista algumas aulas, mesmo que estas não tenham efetivamente ocorrido, desde que exista uma justificação válida para a ausência da aula. Estas situações podem ocorrer, por exemplo, quando existe um dia feriado ou um dia em que exista um evento académico.

Nestes casos, é atribuído o valor zero (0) ao atributo *nr_presenças* e a justificação é inserida no atributo *motivo_falta*. Por isso, durante o processo de conceção, foram avaliadas a possibilidades de transformar o *motivo_falta* numa das dimensões do sistema ou mantê-lo como um atributo da tabela fato (fato_aula).

Considerando que não havia necessidade de uma análise detalhada dos motivos das faltas e que este dado é utilizado apenas para verificar se a aula ocorreu efetivamente e para justificar algumas situações em que o valor *nr_presenças* é igual a zero, optou-se por mantê-lo como um atributo da entidade fato_aula.

Esta decisão foi tomada com base na simplicidade e na eficiência do modelo, evitando a criação de uma dimensão adicional desnecessária.

Uma representação da tabela fato_aula, armazenada no DW, pode ser vista na Tabela 8.

Tabela 8 – Atributos Fato_Aula.

Atributo	Descrição	Tipo	Obs
aula_id	Identificador da aula (não único)	Int	-
aula_id_curso	Identificador da aula (chave primária natural)		

fato_aula_key	Identificador- chave primária anterior	int	PK
carga_aula	medida carga horária da aula	float	
data_aula	Relação calendário_letivo	date	FK
dim_docente_key	Relação dimensão docente	int	FK
dim_escola_curso_key	Relação dimensão escola curso	int	FK
dim_grupo_uc_key	Relação dimensão grupo_uc	int	FK
dim_uc_componente_turno_key	Relação dimensão uc_componente_turno	int	FK
n_alunos	Indicador número presenças na aula	int	-
motivo_falta	Descrição motivo da falta (se houver)	texto	
expired	Controle histórico	texto	

Dimensão Docente (Dim_Docente)

A dimensão docente *(dim_docente)* armazena dados sobre os docentes que lecionam as aulas.

Além do código do docente, a dimensão docente também armazena o nome do docente. Na Tabela 9 estão descritos os atributos da dimensão docente.

Tabela 9 – Atributos Dimensão docente.

Atributo	Descrição	Tipo	Obs
key_dim_docente	Chave primária "artificial" (surrogate key)	int	PK
docente_cod	Código docente no IPL (chave natural)	Int	-
docente_nome	Nome do docente	Texto	-

Dimensão Unidade Curricular (dim uc_componente_turno)

Os turnos estão relacionados com as componentes de uma unidade curricular. Por esse motivo foi criada uma dimensão com a granularidade do turno para garantir que a informação obtida está de acordo com o esperado.

Para assegurar a atribuição de um identificador único aos registos, os dados são inseridos no *DW* com base na premissa de que a combinação da tupla (*uc_cod*, componente e turno) deve ser suficiente para identificar de forma exclusiva cada registo. Então, é gerada uma chave artificial, designada *key_uc_componente_turno*, que garante a singularidade de cada entrada na dimensão.

Os atributos da dimensão uc_componente_turnos estão descritos na Tabela 10.

Tabela 10 – Atributos Dimensão Uc_Componente_Turno.

Atributo	Descrição	Tipo	Obs
key_uc_componente_turno	Chave primária "artificial" (surrogate key)	int	PK
uc_cod	Código da uc (chave natural)	Int	
turno	Turno da componente (0,1,2), (chave natural)	int	"sem turno" convertido para 0
componente	Dados sobre componente (T,P, etc) (chave natural)	Texto	
uc_nome	Nome da uc	Int	
uc_ano	Ano do uc	Int	

Além desses atributos, no momento da visualização são criados os seguintes atributos calculados para facilitar a visualização dos dados: *componente-turno(concat), uc_dados_completos(concat), uc_nome(uc_cod)(concat), uc_componente_turno(concat), uc_nome_uc_cod_curso_regime(concat).*

Dimensão Escola Curso (dim_escola_curso)

Cada um dos cursos está associados a uma escola específica do IPL, de acordo com a sua "área de saber". Verificou-se que não há cursos com o mesmo nome ou código vinculados a

escolas distintas. Adicionalmente, no que diz respeito à escola, apenas o nome desta é utilizado nas análises, sendo este atributo utilizado exclusivamente para filtros preliminares.

Deste modo, optou-se por agregar a informação sobre a escola e os cursos numa única dimensão, designada *dim_escola_curso*, de forma a simplificar a estrutura e otimizar o processo de análise. Os atributos da dimensão escola_curso são descritos na Tabela 11.

Tabela 11 – Atributos Dimensão Escola_Curso.

Atributo	Descrição	Tipo	Obs:
key_dim_escola_curso	Chave primária "artificial" (surrogate key)	Int	PK
curso_cod	Código do curso (chave natural)	Int	-
curso_nome	Nome do curso	Texto	
curso_regime	Regime do curso	Texto	
curso_modalidade	Ciclo de estudo (licenciatura, mestrado etc.)	Texto	
escola_nome	Nome da escola vinculada ao curso	Texto	
expired	Controle histórico	Texto	-

Adicionalmente, no momento da visualização de dados, foi criado o atributo com a junção dos nomes do curso e do regime para facilitar a visualização da informação.

Dimensão Grupo Disciplinar (dim_grupo_uc)

Como referido anteriormente, na maioria dos casos, cada aula é ministrada para apenas um turno de uma unidade curricular. Portanto, nessas ocorrências, cada *aula_id* está relacionado a uma única unidade curricular.

No entanto, em alguns casos, turnos de diferentes unidades curriculares são agrupado, para otimizar o uso de recursos, formando os *grupos disciplinares*.

A existência de um grupo disciplinar implica que dois registos de aulas apresentarão a mesma aula_id. Contudo, nessas situações, os dados das unidades curriculares dos registos serão distintos.

Nos dados de aulas, os registos de aulas que pertencem a um grupo disciplinar possuem o atributo *grupo_disciplinar* preenchido (em formato textual). Portanto, a dimensão *dim_grupo_uc* utilizará essa informação como premissa.

A dimensão *dim_grupo_uc* armazena dados relacionados com o grupo (como a designação do grupo) e dos turnos das componentes das unidades curriculares.

Assim, se um grupo for composto por dois turnos, haverá dois registos distintos para esse grupo na tabela *dim_grupo_uc*.

Estes registos partilharão o mesmo atributo *grupo_uc_descricao*, mas terão valores diferentes para o atributo *dim_uc_componente_turno_key*, que identifica de forma única cada turno associado. Na Tabela 12 estão descritos os atributos da dimensão Grupo_Uc.

Tabela 12 – Atributos Dimensão Grupo_Uc.

Atributo	Descrição	Tipo	Obs
key_dim_grupo_uc	Chave primária "artificial" (surrogate key)	int	PK
grupo_uc_descricao	Descrição do grupo (obtida no ficheiro de aula	int	-
dim_uc_componente_turno_key	Chave estrangeira de uc_componente_turno_key	int	FK
uc_cod	Código da UC	Int	
Turno	Turno da UC	texto	
Componente	Componente da UC	texto	

Tabelas auxiliares e a Dimensão Calendário Letivo (dim_calendario_letivo)

No contexto de uma IES, as atividades estão diretamente vinculadas ao calendário letivo da instituição, que pode diferir do calendário civil tradicional. Por isso, a dimensão Calendário Letivo foi modelada para a realização das análises temporais.

Em termos gerais, cada ano letivo é composto por dois *semestres letivos.* Ao contrário do calendário civil tradicional, no calendário letivo as datas de início e fim dos anos e dos semestres letivos costumam variar, e só são conhecidas pouco antes do início do ano letivo. Além disso, ao longo dos semestres letivos, podem existir datas "especiais" que influenciam diretamente na assiduidade dos alunos nas aulas.

No IPL, por exemplo, no início do primeiro semestre, existe uma data reservada para o *Desfile do Caloiro,* mas essa data varia ao longo dos anos letivos. Neste dia, as aulas são suspensas.

Para a adequada análise da assiduidade, é relevante é que esses dados estejam presentes no modelo. Entretanto, nas fontes disponibilizadas não foram identificados dados referentes aos anos letivos e nem às datas especiais.

Por isso, foi desenvolvido um *script*, em linguagem *Python,* que permite a inserção ou modificação de datas especiais, bem como a alteração das datas de início e fim dos anos e semestres letivos, forma eficiente e personalizada.

Dessa forma, o utilizador tem a capacidade de atualizar os dados do calendário letivo sempre que considerar necessário.

A dimensão calendário letivo é gerada a partir de duas tabelas auxiliares: *tab_ano_letivo*, que contém dados sobre o início e o fim de cada período letivo, e *tab_datas_especiais_stg*, que regista as datas especiais para a instituição.

No início do processo de visualização, os dados provenientes dessas duas tabelas são consolidados, e, a partir dessa integração, a dimensão *dim_calendário_letivo* é criada, garantindo que toda a informação relevante sobre o calendário académico da instituição esteja disponível para análise.

Esta abordagem foi adotada devido ao facto de os dados relativos aos anos e semestres letivos poderem ser alterados pelo utilizador. A manutenção desta dimensão no DW aumentaria a complexidade do processo de atualização.

A dimensão calendário letivo relaciona-se à tabela fato através da data da aula.

Os atributos da dimensão calendário letivo, e as respetivas descrições, podem ser vistos na Tabela 13.

Tabela 13 - Atributos Dimensão calendário letivo, modelo dimensional expandido.

Atributo	Descrição	Tipo	Obs
data	Data (dd/mm/aaaa)	Data	PK
ano letivo	Ano letivo (ex:2022/2023)	Texto	-
semestre_letivo(calc)	Semestre Letivo	Texto	Ex: ‘S1’’S2’
ano_semestre_letivo(concat)(calc)	Concatenação ano com semestre letivo		

data inicio período	Data do início do período letivo	Data	
data especial	Indicador se a data é especial (ex: feriado ou interrupção letiva)	Texto	-
dia_da_semana(abreviado)(calc)	Dia da semana abreviado	Texto	
dia_da_semana(extenso)(calc)	Dia da semana	Texto	
dia_da_semana(numero)(calc)	Número do dia da semana	Int	
semana_letiva	Indicador da semana letiva	Int	-

O código utilizado para criação da dimensão calendário encontra-se no Anexo B – Código Dimensão Calendário (M).

Modelo dimensional e modelo dimensional expandido do projeto

O modelo dimensional do projeto, armazenado no DW, e o modelo dimensional expandido, utilizado para a realização das análises na aplicação de visualização, podem ser vistos a seguir, respetivamente na Figura 18 e na Figura 19.

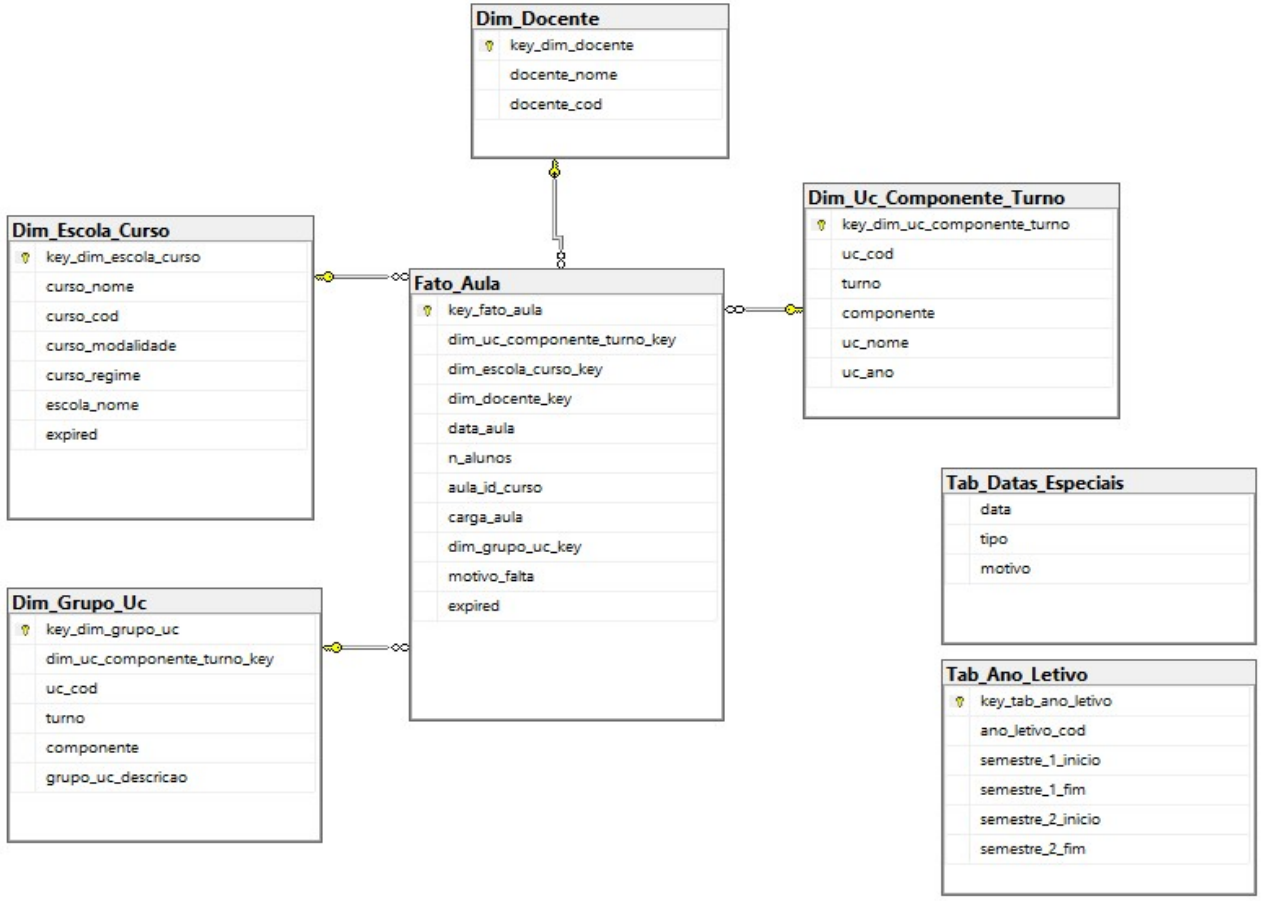


Figura 18 - Modelo dimensional, DW.

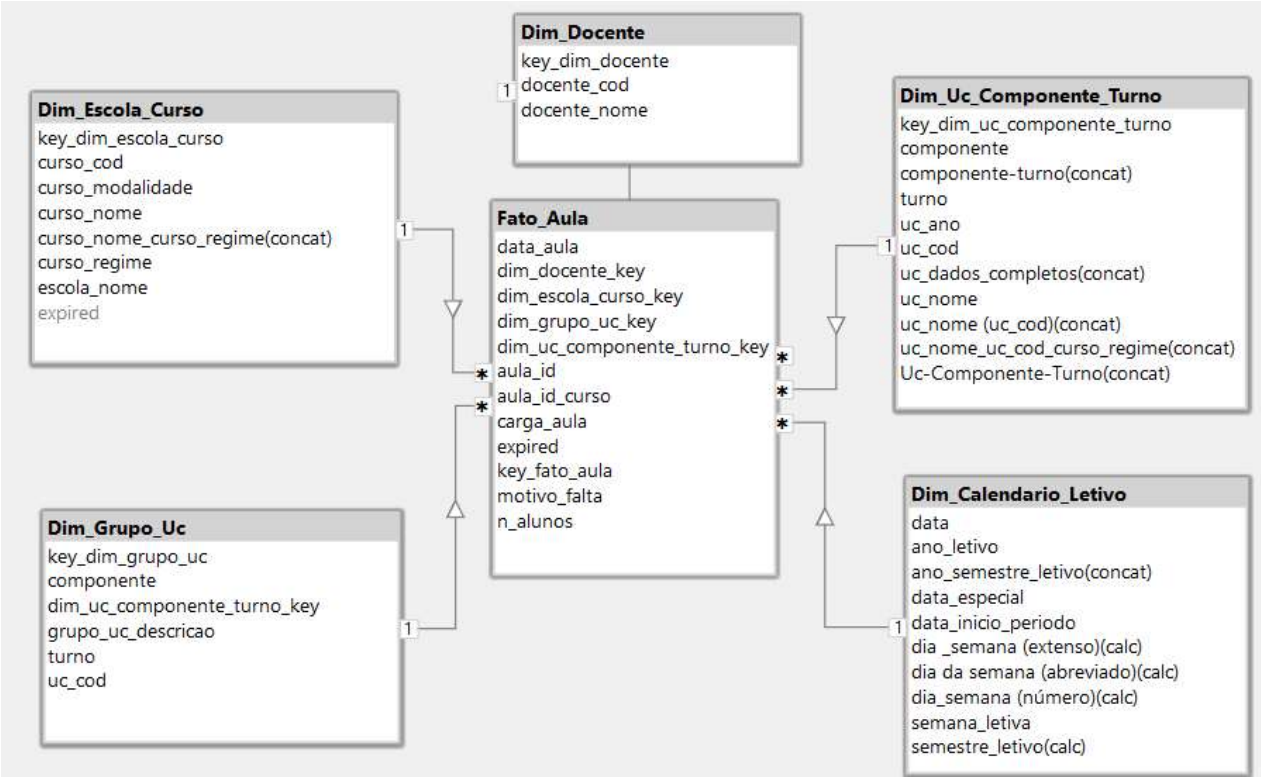


Figura 19 - Modelo dimensional expandido.

3.6. Identificação de ferramentas utilizadas

A implementação de um projeto de BI requer a utilização de ferramentas para garantir a recolha, transformação, armazenamento e visualização de dados de forma eficiente.

No âmbito deste projeto, a seleção das ferramentas teve em consideração três fatores fundamentais: (1) a familiaridade do programador com as tecnologias envolvidas, permitindo uma implementação mais eficiente e reduzindo a curva de aprendizagem; (2) a facilidade de manutenção, assegurando a sustentabilidade e a escalabilidade da solução a longo prazo; (3) o acesso à tecnologia, garantindo a viabilidade da implementação com base na disponibilidade de recursos e infraestruturas adequadas.

Considerando esses fatores, para a gestão da base de dados, utilizou-se o SQL Server como Sistema de Gestão de Bases de Dados (SGBD) e a *Structured Query Language* (SQL) como linguagem principal para manipulação e transformação dos dados.

Na fase de integração de dados optou-se pela utilização do SQL Server Integration Services (SSIS). Essa plataforma de integração, desenvolvida pela Microsoft, permite a criação de *pipelines* de ETL de forma visual para integrar dados de fontes heterogéneas, utilizando componentes pré-configurados.

Foram desenvolvidos *scripts* em SQL, os quais são invocados em *containers* do SSIS. Adicionalmente, outros componentes do SSIS são utilizados para a carga de dados no DW, uma vez que facilitam a implementação do processo de *Slowly Changing Dimensions (*SCD*).*

Para a conceção dos *dashboards*, recorreu-se ao *Power BI.* O programador já possuía experiência com outras aplicações do Microsoft Office, e a interface semelhante, aliada à integração nativa, facilitou o processo de aprendizagem. Adicionalmente, a possibilidade de utilização da linguagem *Data Analysis Expressions* (DAX) permitiu a criação de medidas e cálculos dinâmicos, viabilizando análises avançadas e personalizadas, requisitos alinhados com as necessidades do projeto.

Para automatizar determinadas tarefas, possibilitar a inserção de dados por parte do utilizador e realizar a análise exploratória da informação, foram desenvolvidos *scripts* em linguagem *Python*. A escolha do *Python* justifica-se pela sua versatilidade, eficiência e vasta biblioteca de ferramentas para manipulação, transformação e análise de dados.

A utilização combinada destas ferramentas/linguagens proporcionou escalabilidade, flexibilidade e facilidade de manutenção ao sistema desenvolvido.

4. Implementação do sistema de *Business Intelligence*

Este capítulo apresenta as principais etapas do processo de implementação do sistema de *BI*, os desafios enfrentados e as práticas essenciais para garantir o sucesso e a sustentabilidade do sistema.

4.1. Desenvolvimento do *Pipeline*

Desde a criação do ambiente operacional até a disponibilização da informação em relatórios e *dashboards*, a implementação e a manutenção de um sistema de *BI* envolvem uma série de etapas interdependentes, que precisam ser cuidadosamente planeadas e executadas.

Este processo, caracterizado pela natureza dinâmica e contínua das operações, exige um fluxo de trabalho eficiente. Enquanto algumas etapas são de execução única, outras necessitam de repetição regular para assegurar que o sistema seja consistentemente alimentado com dados atualizados e fiáveis.

Neste contexto, a automação das tarefas assume um papel fundamental, permitindo não apenas a redução de erros manuais, mas também a otimização de recursos e a melhoria da consistência e precisão dos dados.

Para orquestrar as diferentes etapas do processo, foi implementado um *pipeline*, desenvolvido com recurso do SSIS. Este *pipeline* assegura a integração, transformação e carga dos dados de forma estruturada, planeada e eficiente.

O *pipeline* está estruturado em três macrofases principais: a criação do ambiente operacional (*staging area* e DW*),* a extração e transformação dos dados, e, por fim, a sua carga no DW. Uma representação das fases do *pipeline* pode ser vista na Figura 20.

Figura 20 – *Pipeline* do sistema proposto.

Enquanto a fase de criação do ambiente operacional é realizada apenas uma vez, durante a implantação inicial do sistema, as fases subsequentes — extração, transformação e carga dos dados — devem ser executadas de forma recorrente, sempre que seja necessária a atualização dos dados.

4.2. Desenvolvimento da *Staging Area*

No processo de ETL é fundamental garantir a eficiência e a integridade dos dados. Por isso, não é aconselhável a manipulação dos dados diretamente nas fontes. Por esse motivo, para o desenvolvimento do sistema de BI optou-se pela utilização de uma S*taging Area.*

A adoção dessa solução oferece algumas vantagens como a possibilidade de realizar transformações complexas de dados de forma isolada, sem impactar o ambiente de produção, a possibilidade de facilitar a deteção e correção de erros nos dados antes que estes sejam carregados para o destino final e a possibilidade de processamento dos dados de forma mais controlada e eficiente.

Estas áreas intermediárias armazenam apenas os dados em processo de transformação. Somente após a conclusão de todo o processo de transformação é que os dados serão carregados no DW.

Considerando que o armazenamento na *Staging Area* é temporário e de apenas parte dos dados, o impacto da utilização de espaço em disco é muito pequeno e as vantagens obtidas justificam o uso dessa metodologia.

Neste projeto, a *Staging Area* é implementada através de uma base de dados SQL, designada *Assiduidade_Stg*, cuja estrutura apresenta semelhanças com a do DW. A sua conceção foi

orientada pela necessidade de responder eficazmente aos requisitos que motivaram o desenvolvimento do projeto.

O banco de dados Assiduidade_Stg possui 10 tabelas: tab*_ano_letivo, aulas_json, docente, docentes_ficheiro, escola_curso, fato_aula_stg, grupo_uc, tab_datas_especiais_stg, uc_componente_turno, uc_curso*. Algumas dessas tabelas são apenas auxiliares, enquanto outras armazenam os dados já tratados e são seus registos transferidos diretamente para o DW.

4.3. Desenvolvimento do *Data Warehouse* e da *Staging Area*

O DW foi concebido para refletir o modelo dimensional delineado no âmbito do projeto.

Assim, a estrutura do DW inclui uma tabela de fatos, designada por *fato_aula*, quatro tabelas de dimensão (*dim_escola_curso*, *dim_docente*, *dim_grupo_uc*, *dim_componente_turno)* e duas tabelas auxiliares (*tab_ano_letivo* e *tab_datas_especiais)* que são a base para a dimensão dim_calendario_letivo, que será construída dinamicamente.

Esta arquitetura visa assegurar a organização eficiente e a análise sistemática dos dados, alinhando-se com os princípios do modelo dimensional. O código parcial, em linguagem SQL, que cria o DW é apresentado na Figura 21.

```
CREATE table fato_aula(
    key_fato_aula INT IDENTITY(1,1) PRIMARY KEY,
    dim_uc_componente_turno_key  INT REFERENCES dim_uc_componente_turno(key_dim_uc_componente_turno),
    dim_escola_curso_key INT REFERENCES dim_escola_curso(key_dim_escola_curso),
    dim_docente_key INT REFERENCES dim_docente(key_dim_docente),
    data_aula date,
    aula_id_curso bigINT,
    carga_aula numeric(4,2),
    expired varchar(10),
    dim_grupo_uc_key INT default null REFERENCES dim_grupo_uc(key_dim_grupo_uc)


    )
```

Figura 21 - Criação do DW (parcial).

De forma semelhante ao DW, também há um *script* para a criação da *Staging Area.*

A efetivação da implementação do DW e da *Staging Area* é realizada com a invocação dos respetivos *scripts* SQL em containers do SSIS, conforme pode ser observado na Figura 22.

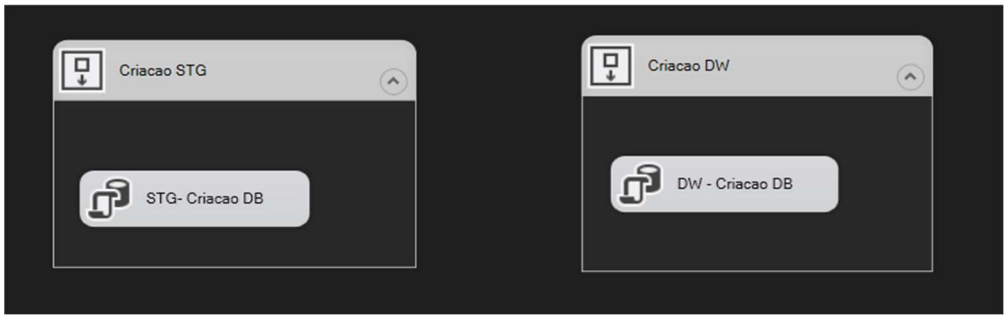


Figura 22 - Orquestração da criação da Stg e do DW no SSIS.

É fundamental ressaltar que o processo de criação do DW e da *Staging Area* é realizado apenas uma vez, no momento de implantação do sistema.

4.4. Desenvolvimento do processo de ETL

Como visto no capítulo de revisão de literatura, o processo de ETL (*Extract, Transform, Load)* é uma etapa fundamental na construção e manutenção de um *DW*, assegurando a integração e a preparação dos dados provenientes de diversas fontes.

No processo de ETL o SSIS é utilizado para invocação de *scripts* nas fases de extração e transformação e, utilizado, como forma de desenvolvimento do processo de carregamento por uma questão de simplificação de implementação de SCDs tipo 1 e tipo 2, responsáveis por controlar o comportamento da evolução de alterações em atributos de algumas das dimensões.

4.4.1. Processo de Extração

Na fase de Extração (E), os dados são obtidos a partir das diversas fontes de informação relevantes para o projeto.

Especificamente no contexto desse projeto, os dados extraídos referem-se às aulas ministradas, os docentes envolvidos, as unidades curriculares/componentes/turnos, os cursos e respetivas escolas, bem como os períodos letivos abrangidos.

Por conta da flexibilidade e da possibilidade de automação do processo, a recolha dos dados relacionados às aulas é realizada através de uma interface com o sistema académico da

instituição. Para automatizar a extração dos dados dessa interface, foi desenvolvido um *script* em *Python*.

Com essa solução, sempre que o utilizador queira atualizar os dados deve executar o *script* e a atualização será realizada de forma transparente e automática, armazenando os dados em um ficheiro *json.*

Ficheiro de cursos

Os primeiros dados extraídos referem-se aos cursos oferecidos pelo IPL. Para esse propósito, foi disponibilizado pelos gestores um ficheiro, denominado *dim_curso.xlsx.* O documento contém os seguintes dados dos cursos: *código_escola, código_curso, designação_curso e regime.*

De forma a evitar problemas de incompatibilidade, a extração dos dados contidos neste ficheiro é realizada diretamente no *pipeline* do SSIS, logo após a criação da *Staging Area.*

Os dados extraídos são então transferidos para a tabela *escola_curso* da *Staging Area*, garantindo a integridade e a consistência dos dados ao longo do processo.

Ficheiro de aulas

Os dados provenientes do ficheiro de aulas (*base-aulas-atualizada.json*) são importados para a tabela *aulas_json*. Neste processo, cada entrada presente no ficheiro é convertida num registo individual, enquanto cada campo do *json* é mapeado para uma coluna correspondente da tabela.

A partir desse momento, todas as transformações passam a ser realizadas exclusivamente dentro da *Staging Area*, sem qualquer modificação no ficheiro original.

Ficheiros de cursos e unidades curriculares

Os dados sobre os cursos e as UCs são disponibilizados pelos gestores em um ficheiro no formato *xlsx*. A atualização desses dados depende da atualização manual desses ficheiros.

4.4.2. Processo de Transformação

A transformação é comumente a etapa mais custosa do processo de ETL. Ao longo dessa etapa, os dados são submetidos a uma série de operações incluindo limpeza, normalização,

filtragem e agregação. Tudo isso para garantir a qualidade e a consistência dos dados antes do efetivo carregamento.

O processo de transformação é desenvolvido usando a linguagem SQL. No entanto, é utilizado o SSIS de forma a automatizar essa tarefa.

Padronização- Designação(nome) dos cursos

Ao analisar a designação (nome) dos cursos do ficheiro de cursos e compará-las com as designações presentes no ficheiro de aulas, foram identificadas algumas discrepâncias.

Por isso, para assegurar a consistência e a integridade dos dados ao longo do processo, procedeu-se com a padronização das designações dos cursos.

Em geral os ajustes nos nomes dos cursos foram realizados na tabela aulas_json. A única padronização realizada na tabela escola_curso referiu à troca do termo *'Profissional em'* para *'Profissional de'* nos casos de Curso Técnico Superior em alguns registos que apresentava nomenclatura diferente. A Figura 23 mostra o código SQL utilizado para realização dessa padronização.

```
UPDATE escola_curso
SET curso_nome =
    CASE
        WHEN curso_nome LIKE 'Curso Técnico Superior Profissional em %' THEN REPLACE (curso_nome, 'Profissional em', 'Profissional de')
        ELSE curso_nome
    END
```

Figura 23 - Padronização nome Curso Técnico Superior.

Filtragem de Dados

A etapa de filtragem de dados é fundamental para eliminar dados irrelevantes ou inconsistentes, garantindo que as etapas subsequentes de transformação e integração sejam realizadas de forma eficiente e com um conjunto de dados limpo e preciso, adequado aos objetivos do projeto. Foram realizadas filtragens nos dados referentes às aulas e aos cursos.

Tabela aulas_json

O sistema académico contém dados relativos às aulas realizadas em todas as escolas do IPL. No entanto, inicialmente, o sistema de BI é restrito à ESTG. Dessa forma, é imprescindível realizar uma filtragem dos dados obtidos pela interface.

Adicionalmente, para garantir a correta integração dos dados, é imprescindível que todos os cursos presentes no ficheiro de aulas também estejam registados no ficheiro de cursos.

Nesse sentido, além da filtragem por escola, é realizada uma filtragem adicional por curso, com o intuito de assegurar que todos os cursos analisados estão devidamente contemplados no ficheiro de cursos, evitando assim a inclusão de dados inconsistentes ou ausentes na base de dados.

Tabela escola_curso

A tabela *escola_curso*, proveniente do ficheiro *dim_curso.xlsx*, contém dados sobre todos os cursos oferecidos pelo IPL. No entanto, como o projeto se destina exclusivamente aos cursos da ESTG, foi realizada uma filtragem na tabela para garantir que apenas os dados referentes aos cursos dessa escola fossem mantidos.

Tratamento de dados truncados

A correção de dados truncados envolve a revisão e a atualização dos registos com alguma inconsistência, garantindo que todos os dados estejam completos e adequados ao formato adequada para as etapas posteriores.

Em alguns registos de aulas, foram identificadas concatenações indevidas de atributos. Nesses casos, o atributo *unidade_curricular* armazenava, de forma inadequada, diversos dados, enquanto os atributos correspondentes (*uc_cod*, *uc_nome*, *uc_ano* e *uc_componente*) estavam vazios, sem qualquer informação associada.

Para resolver essa questão, foi necessário realizar a desagregação dos dados concatenadas e redistribuí-los nos atributos corretos, garantindo a precisão e a consistência. O código parcial, utilizado para realização do tratamento de dados truncados, pode ser visto na Figura 24.

```
--- Correção das linhas com UC null
UPDATE aulas_json
SET uc_cod = LEFT(unidade_curricular, PATINDEX('%-%', unidade_curricular) - 1),
    unidade_curricular = SUBSTRING(unidade_curricular, PATINDEX('%-%', unidade_curricular) + 1, LEN(unidade_curricular))
    WHERE uc_cod =''

--- Extração ano da unidade curricular

UPDATE aulas_json
    SET uc_ano = SUBSTRING(unidade_curricular, CHARINDEX(' Ano', unidade_curricular) + 4, LEN(unidade_curricular))
    WHERE unidade_curricular LIKE '% Ano[0-9]' OR unidade_curricular LIKE '% Ano [0-9]';

--- Extração do nome da UC
UPDATE aulas_json
SET uc_nome =
    CASE
        WHEN uc_ano IS NOT NULL THEN SUBSTRING(unidade_curricular, 1, LEN(unidade_curricular) - LEN(uc_ano) - 4)
        ELSE unidade_curricular
    END;
```

Figura 24 - Tratamento de dados concatenados (parcial).

Tratamento de dados Nulos/Ausentes

A presença de dados nulos ou a ausência de dados pode comprometer a qualidade e a fiabilidade das análises subsequentes. Dados com essas características podem introduzir vieses ou distorções nos resultados, levando a conclusões erradas ou imprecisas. Por esse motivo, é imperativo que o tratamento dessas situações seja realizado de forma sistemática e criteriosa.

No ficheiro de aulas, mesmo após o tratamento do atributo *unidade_curricular* que possuía dados truncados, alguns registos continuaram com ausência de dados no atributo *componente.* Para tratar essa situação, recorreu-se à técnica de substituição por valor constante, substituindo os valores em falta pelo termo "N.I.", que representa "Não Informado".

Esta abordagem garante que os dados ausentes sejam identificados de forma clara e não causem interferência na análise, permitindo a continuidade do processo sem a necessidade de exclusão de dados e sem comprometer a integridade dos mesmos.

Preenchimento atributos 'tipo_curso'

Para atender aos requisitos preliminares do projeto, é imprescindível que os cursos sejam classificados de acordo com o respetivo ciclo de estudo (Curso Técnico Superior, Licenciatura, Mestrado, Doutoramento, Outros). Contudo, nos ficheiros fornecidos, não existe um atributo que contenha essa informação de forma isolada.

Entretanto, observou-se que o atributo *nome_curso* faz referência ao ciclo de estudo do curso. Sendo assim, essa informação foi extraída do nome do curso e armazenada em um novo atributo denominado *curso_modalidade*. O código que realiza essa operação pode ser visto Figura 25.

```
--- Preenchimento da coluna tipo_curso na tabela curso
UPDATE escola_curso
SET  escola_curso.curso_modalidade =
    CASE
        WHEN escola_curso.curso_nome LIKE 'Undergraduate%' THEN 'Licenciatura'
        WHEN escola_curso.curso_nome LIKE 'Licenciatura%' THEN 'Licenciatura'
        WHEN escola_curso.curso_nome LIKE 'Curso%Técnico Superior Profissional%' THEN 'TeSP'
        WHEN escola_curso.curso_nome LIKE 'Pós-Graduação%' THEN 'Pós-Graduação'
        WHEN escola_curso.curso_nome LIKE 'Mestrado%' OR escola_curso.curso_nome LIKE 'Master%' THEN 'Mestrado'
        WHEN escola_curso.curso_nome LIKE 'Doutoramento%' THEN 'Doutoramento'
        ELSE 'Outros' -- Se nenhum dos padrões acima corresponder, será definido com Outros
    END;
GO
```

Figura 25 - Preenchimento atributo curso_modalidade.

Esta abordagem permite a categorização eficiente dos cursos, atendendo às necessidades do projeto.

Preenchimento atributo 'escola_nome'

Para associar a escola à qual o curso pertence, o ficheiro *dim_escola* utiliza um atributo numérico. Os cursos da ESTG, por exemplo, estão relacionados à escola de número '2'.

Contudo, para facilitar a análise e proporcionar maior clareza na visualização dos dados, é essencial que o nome da escola seja utilizado de maneira explícita. Assim, na tabela *escola_curso*, foi criado um atributo denominado *escola_nome*, e esse atributo foi preenchido com o nome da escola correspondente.

Essa modificação visa melhorar a compreensão e a análise dos dados, tornando a informação mais acessível e intuitiva.

Correção do código do curso nos registos de aulas_json

Durante a análise preliminar, em alguns registos de aulas, foram identificadas inconsistências no atributo referente ao código do curso.

Para proceder à correção dessas discrepâncias, os dados dos cursos presentes na tabela *aula_json* foram comparados com os dados correspondentes na tabela *escola_curso.* Então, o atributo *curso_cod* da tabela *aulas_json* foi devidamente atualizado nos casos necessários.

Uma das dificuldades encontradas no processo de análise referiu-se aos cursos ministrados no campus de Torres Vedras e que possuíam o mesmo nome de curso ministrados em outros *campi*, uma vez que na tabela *escola_curso* não existia uma referência direta que permitisse fazer a correta distinção.

Após uma análise mais aprofundada dos dados, constatou-se que, nos casos em questão, o código do curso ministrado no campus de Torres Vedras era sempre superior ao código do curso de mesmo nome, ministrado no campus principal. Com base nesta observação, foi possível estabelecer uma lógica para distinguir corretamente os cursos, resolvendo assim o problema identificado.

Nesses casos, adicionalmente, procedeu-se a atualização do atributo de designação do curso (*curso_nome*) da tabela *escola_curso,* de acordo com os dados relativos ao campus de Torres Vedras que estavam, indiretamente, presentes na tabela *aulas_json.*

Esta atualização consistiu na adição do termo "TV" ao final do nome do curso, de modo a assegurar a identificação precisa dos cursos associados ao campus de Torres Vedras, promovendo a consistência e a fiabilidade dos dados.

O processo de atualização do código e do nome dos cursos permitiu identificar e corrigir as divergências, assegurando que os dados estivessem consistentes entre as tabelas *aulas_json e escola_curso.*

Atualização do código da Unidade Curricular

Na tabela de aulas, designada por *aulas_json*, foram identificadas discrepâncias em alguns registos entre o código da unidade curricular (*uc_cod*), o respetivo nome (*uc_nome*) e o curso associado. Para mitigar este problema, foi criado um atributo na tabela *aulas_json*, denominado *uc_cod2*, bem como uma tabela auxiliar, intitulada *uc_curso*, com o objetivo de facilitar a análise e a verificação dos dados.

Cada unidade curricular possui um código de identificação único composto de 7 dígitos. Os primeiros quatro dígitos deste código correspondem ao curso a que a UC pertence, enquanto os três dígitos finais identificam especificamente a unidade curricular.

A tabela auxiliar *uc_curso* estabelece a relação entre as unidades curriculares e os respetivos cursos. Nos casos em que existe concordância entre os dados de *aulas_json*, a relação é efetuada diretamente. Contudo, nos casos em que os dados apresentam divergências, como, por exemplo, quando o nome não corresponde ao código de curso implícito no identificador da unidade curricular, são realizadas verificações adicionais para assegurar a adequação dos dados.

Nos registos de aulas das unidades curriculares *Estágio*, *Projeto* e *Dissertação*, foi necessário proceder a um tratamento individualizado para assegurar a padronização dos dados. Identificaram-se três variações na nomenclatura de cada uma destas unidades curriculares, exigindo que a correta identificação do *cod_uc* correspondente fosse realizada com extrema atenção, de modo a evitar quaisquer erros.

Preenchimento da tabela fato_aula_stg

A última etapa do processo de transformação é o preenchimento da tabela *fato_aula_stg*. Esta tabela armazena os dados que, posteriormente, serão transferidos para a tabela *fato* no DW. Ela serve como uma etapa intermediária no processo de integração dos dados,

consolidando os dados provenientes das aulas e preparando-as para a carga final na tabela *fato* do DW.

4.4.3. Processo de Carregamento

Após a conclusão da etapa de transformação, os dados encontram-se preparados para serem carregados no DW. Este processo constitui a última fase do *pipeline* do fluxo de trabalho.

O projeto foi concebido de maneira a assegurar que, de forma geral, cada uma das dimensões do DW receba os dados oriundos de uma tabela específica *staging área*.

A ordem de carregamento revela-se crucial para assegurar a integridade dos dados. Para tal, foram utilizados containers que permitem determinar a ordem precisa das tarefas, garantindo que as dependências e sequências sejam respeitadas de forma rigorosa.

Uma imagem do *pipeline* de carregamento é apresentada na Figura 26.

Figura 26 - *Pipeline* de carregamento.

A tabela fato está relacionada com as dimensões por meio de chaves. Assim, para assegurar a consistência dos dados, é imprescindível que as tabelas de dimensões sejam carregadas antes da tabela fato.

Entre as dimensões, a única dependência existente é entre a dimensão *dim_grupo_disciplinar* e a dimensão *dim_Uc_componente_turno*. Em virtude disso, a *dim_Uc_componente_turno* deve ser carregada antes da *dim_grupo_disciplinar*.

A relação entre as dimensões e a tabela fato é estabelecida através de *surrogate keys* (ou chaves artificiais/substitutas). As *surrogate keys* substituem o uso das chaves primárias originais nas relações e têm como principal objetivo garantir que, ao longo do tempo, seja

possível alterar pontualmente os dados de uma dimensão mantendo o histórico. Alem disso, geralmente, o uso de *surrogate keys* otimiza o modelo, pois, as chaves artificiais são numéricas e sequenciais o que facilita a indexação e melhora a performance.

O atributo *expired* é utilizado para identificar a versão do registo da dimensão que é válida para as análises e quais são históricos de atualizações.

A inserção dos dados na tabela fato constitui uma tarefa complexa, uma vez que implica estabelecer corretamente a relação entre o registo de aula e as dimensões correspondentes. Este fluxo de dados, por si só, envolve mais de quinze tarefas intermediárias. A parte final da etapa de carregamento da tabela fato é vista na Figura 27.

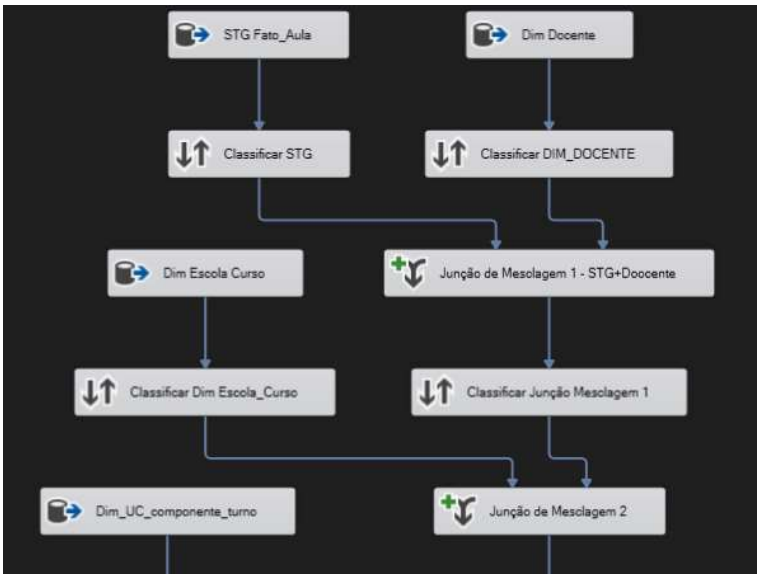


Figura 27 - Fluxo de tarefas de carregamento da tabela fato (parcial).

Script Python

Os semestres letivos apresentam variações nas datas de início e fim, a depender do ano letivo. Nas fontes fornecidas, não foram encontrados dados sobre esse tema. Como o sistema depende desses dados para o adequado enquadramento das aulas nos respetivos semestres letivos, foi desenvolvido um *script* em *Python* para que o utilizador pudesse inserir esse dado manualmente.

Para assegurar que sempre haverá uma correspondência entre uma aula e um semestre letivo, no momento da construção do DW a tabela *tab_ano_letivo*, utilizada para construção da dimensão calendário letivo, é previamente preenchida com datas definidas para os inícios e términos dos semestres até ao ano de 2040.

A partir desse momento, o utilizador pode utilizar o *script* desenvolvido para proceder à atualização das datas de início e fim dos semestres letivos. Na Figura 28, apresenta-se uma ilustração do *script* utilizado para a atualização desses dados.

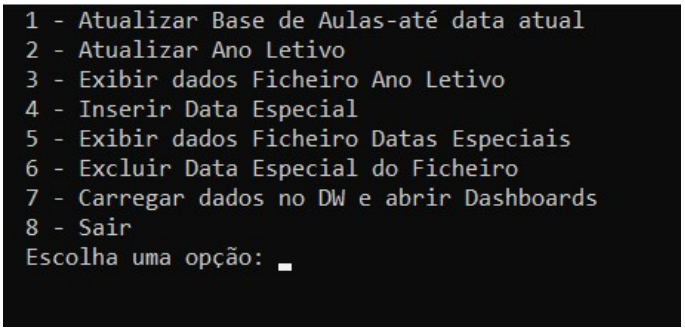


Figura 28 - *Script* para atualização dos dados.

Os dados inseridos pelo utilizador são armazenados num ficheiro *json* e, no momento da atualização do DW, são utilizados para proceder às alterações na tabela *tab_ano_letivo*.

Com o intuito de evitar a introdução de dados incorretos, foram implementados mecanismos de controlo que asseguram a conformidade e adequação desses dados.

Uma vez que a dimensão de calendário letivo é gerada dinamicamente apenas no momento da visualização dos dados e para as datas de aulas que constam na tabela fato, as alterações efetuadas não comprometem a integridade nem geram inconsistências na informação apresentada.

4.5. Desenvolvimento de painéis e *dashboards*

Após o tratamento dos dados e o subsequente armazenamento no DW, iniciou-se a fase de desenvolvimento dos painéis e *dashboards*. Nesta etapa, a informação é disponibilizada ao utilizador de forma estruturada, concretizando-se, assim, os objetivos do projeto.

Essa fase foi realizada no *Power BI* e os tópicos seguintes abordam as medidas e parâmetros, elementos essenciais para a personalização e dinamização das análises, bem como a conceção e estruturação dos painéis e *dashboards*.

4.5.1. Medidas e parâmetros

No cálculo das métricas, recorreu-se à utilização de medidas. As medidas são fórmulas que realizam cálculos dinâmicos com base nos dados presentes no modelo, permitindo a adaptação das análises às interações do utilizador, como filtros e seleções. No *Power BI*, a definição dessas medidas é realizada por meio da linguagem DAX.

Dada a complexidade das análises envolvidas, o desenvolvimento dos dashboards deste projeto exigiu a criação de mais de cinquenta medidas. A construção destas medidas não só permitiu a obtenção de indicadores precisos e adaptáveis, como também garantiu a eficiência e a coerência dos cálculos ao longo dos diferentes visuais.

A Figura 29 ilustra uma das medidas criadas, utilizada para calcular a média de presenças/aula.

```
1 Media_Assiduidade_Aula =
2                 DIVIDE(
3                     sum(Fato_Aula[n_alunos]),
4                     DISTINCTCOUNT(Fato_Aula[aula_id]),
5                     0
6                     )
```

Figura 29 - Medida para cálculo da média de assiduidade por aula.

Além das medidas, para possibilitar a personalização dos *dashboards*, foram utilizados parâmetros. No *Power BI*, um parâmetro é utilizado para ajustar dinamicamente as análises e visualizações, sem a necessidade de modificar diretamente os dados ou as medidas.

Um dos exemplos da utilização de parâmetros é o ajuste dinâmico do número mínimo de presenças para a realização das análises de assiduidade, realizado com um parâmetro do tipo campo. Ao modificar esse valor, o *dashboard* é automaticamente atualizado para refletir o atendimento ou não do mínimo estabelecido. A Figura 30 ilustra o parâmetro de campo criado para possibilitar que o utilizador ajuste o número mínimo de alunos na análise de assiduidade e como esse elemento foi integrado a um segmentador.

Tabela de Valores:

```
Parâmetro Campo Mínimo_Alunos_Tabela = GENERATESERIES(0,100, 1)
```

Seleção:

```
Parâmetro Valor Mínimo_Alunos = SELECTEDVALUE('Parâmetro Campo Mínimo_Alunos'[Parâmetro Mínimo_Alunos], 10)
```

Visual:

Figura 30 - Parâmetro Valor mínimo.

Os dois primeiros itens, (1) tabela de valores e (2) seleção, são integrantes do parâmetro de campo, enquanto o terceiro (3) é o segmentador com a utilização do recurso de parâmetro.

4.5.2. Painéis e *dashboards*

Para assegurar que a informação seja apresentada de forma clara e intuitiva, os *dashboards* foram organizados de maneira segmentada, com foco nos eixos de análise.

Assim, foi concebido um *dashboard* específico para cada tema de análise, nomeadamente: assiduidade, docentes, ciclos de estudo e grupo de UCs.

Adicionalmente, foram desenvolvidos três painéis complementares: (1) painel de capa do projeto, (2) painel destinado à apresentação de uma visão geral dos dados, e (3), um *dashboard* de identificação de equivalências de UCs, de caráter preliminar, concebido para apoiar o gestor na identificação de UCs semelhantes para facilitar a criação de grupos disciplinares.

Com exceção do painel de capa, os demais foram estruturados em elementos principais:

1. **Identificação** – Título do *dashboard*, nome da escola e demais itens que permitem a identificação do painel;
2. **Navegação** – Elementos que permitem ao utilizador deslocar-se entre diferentes *dashboards*, garantindo uma experiência intuitiva. O ícone correspondente ao painel atualmente apresentado surge destacado a amarelo, enquanto os restantes permanecem a branco;
3. **Personalização** – Conjunto de filtros e segmentadores que possibilitam a personalização da análise, permitindo ao utilizador restringir ou expandir

os dados apresentados conforme necessário. No *dashboard* de Visão Geral, a componente de personalização foi substituída por um sumário;

4. **Indicadores/Informação** – Exibição estruturada dos dados, recorrendo a gráficos, tabelas e métricas relevantes.

Essa padronização teve como objetivo facilitar a familiarização dos utilizadores com a estrutura e os tipos de visualizações, assegurando uma análise mais eficiente e precisa dos dados.

Painel de Capa

O painel de capa foi concebido como uma interface inicial de navegação, apresentando uma imagem representativa e um menu interativo. Este painel serve como um ponto de entrada para os demais *dashboards* e é uma forma de apresentação do projeto ao utilizador. O menu de opções orienta o utilizador no acesso às diferentes áreas de análise e é um ponto comum entre todos os *dashboards*.

O painel de capa pode ser visto na Figura 31.

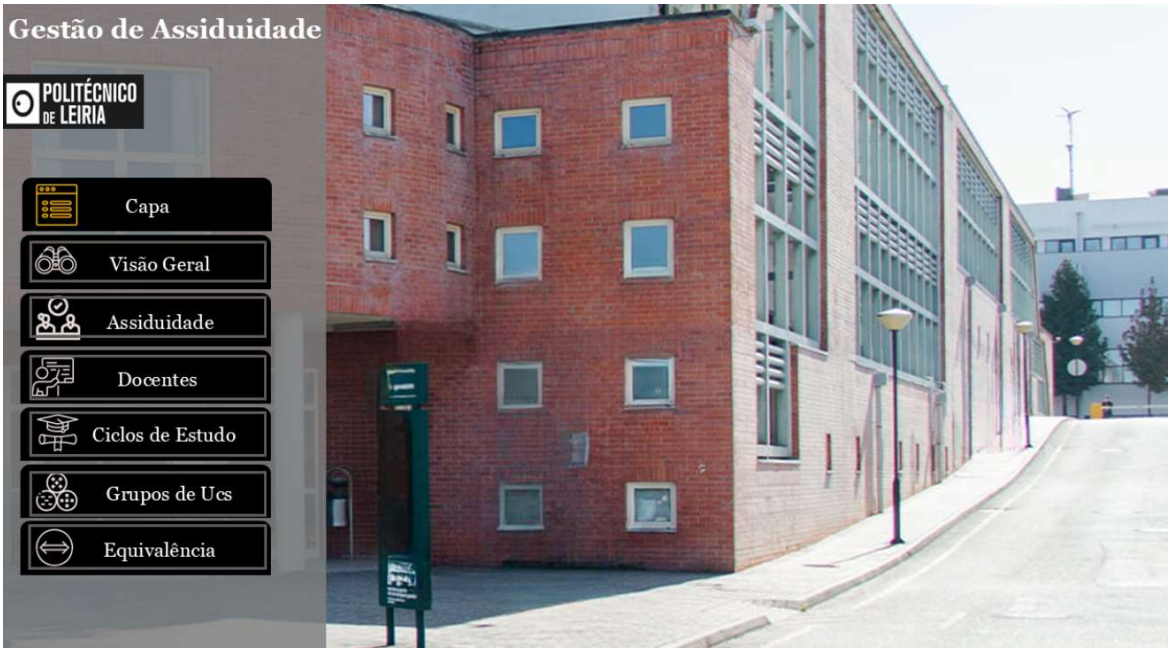


Figura 31 - Painel de Capa.

Painel de Visão Geral

O painel de Visão Geral foi estruturado como um ponto de acesso inicial à análise de dados, proporcionando uma visão global organizada dos dados presentes no DW.

Como o objetivo deste *dashboard* é apresentar uma visão geral sobre todo o conjunto de dados do DW, a componente de personalização foi substituída por um painel com o sumário dos dados.

A Figura 32 apresenta uma imagem do painel de Visão Geral e a identificação das componentes.

Figura 32 - *Dashboard* Visão Geral.

As componentes desse painel são apresentadas a seguir:

1. **Identificação** – Apresenta o nome do painel;
2. **Sumário** – Exibe um conjunto de dados sobre o DW, nomeadamente o número de anos letivos, escolas, ciclos de estudos, registos de aulas, aulas com faltas justificadas, docentes, cursos, unidades curriculares e turnos. Adicionalmente, indica o período a que os dados do DW se referem;
3. **Menu de navegação** – Comum a todos os *dashboards*, permite ao utilizador navegar rapidamente entre os diferentes painéis;

4. **Indicadores/Informação –** A componente de indicadores/Informação foi subdividida em duas áreas:
    i. **Indicadores de média** – No centro desta secção, apresenta-se um gráfico do tipo "velocímetro" com a média geral de assiduidade por aula, juntamente com o indicador mínimo estabelecido pela instituição. Além disso, incluem-se painéis auxiliares que detalham as médias de assiduidade por componente e por ciclo de estudo;
    ii. **Indicadores de totalidade por semestre letivo** – Esta secção contém gráficos que representam o número total de UCs por ciclo de estudo e por turno, bem como um segundo gráfico com o total de aulas e de docentes e, por fim, um terceiro gráfico com o total de turnos por semestre letivo.

O painel de Visão Geral painel agrega a informação mais relevante dos demais *dashboards*, permitindo uma compreensão imediata do contexto geral antes da exploração aprofundada de áreas específicas.

Dashboard de Assiduidade

O *dashboard* de assiduidade constitui o elemento central deste projeto, agregando e exibindo a informação que justificou a sua conceção e desenvolvimento. Nele, encontram-se os principais indicadores e métricas relacionados com a assiduidade, permitindo uma análise detalhada dos padrões de presença e outras variáveis relevantes para a gestão e tomada de decisão.

Assim, este *dashboard* serve como instrumento essencial para monitorizar a assiduidade no contexto em que se insere.

A Figura 33 apresenta uma imagem do *dashboard* de assiduidade.

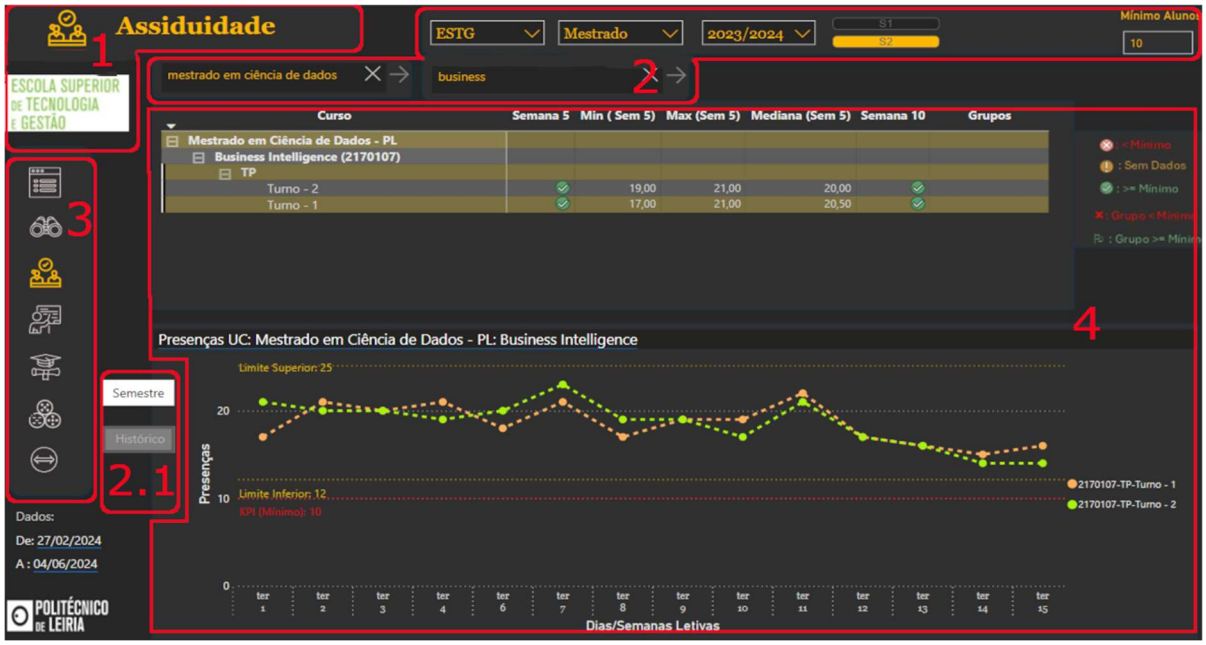


Figura 33 - *Dashboard* Assiduidade.

Os itens (1) e (3) referem-se às componentes de identificação e de navegação, respetivamente. A componente de personalização (2) permite ao utilizador personalizar a visualização do painel de acordo com as suas necessidades, recorrendo a filtros e segmentadores.

Os dados podem ser filtrados de acordo com a escola, o ciclo de estudo, o semestre letivo, o nome do curso ou nome/código da UC. Além de segmentadores, foram utilizados recursos de filtragem para garantir mais flexibilidade na filtragem dos dados.

Ainda na componente de personalização, o utilizador pode definir o número mínimo de presenças, índice fundamental para identificar o cumprimento de metas. A possibilidade de ajustar esse índice possibilita a adaptação da análise a diferentes contextos.

A componente de indicadores (4) está subdividida em duas partes:

a. Matriz de indicadores – A matriz de indicadores apresenta os principais indicadores sobre a assiduidade da UC: cumprimento da meta de assiduidade até a quinta semana e até a décima semana, mediana, mínimo e máximo até a quinta semana se a UC é integrante de algum grupo disciplinar. Para facilitar a visualização, as métricas de cumprimento de metas nas semanas críticas são representadas por alertas visuais de rápida identificação;

b. Gráficos: Na secção dos gráficos pode-se alternar entre um gráfico de linhas, que representa a assiduidade do semestre escolhido, ou um gráfico com o histórico de todos os semestres. A seleção entre os gráficos é realizada através do botão de personalização (2.1).

A seção de gráficos proporciona ao utilizador acesso intuitivo e visual à informação sobre a assiduidade de determinada UC para o semestre letivo em análise. Caso a componente apresente mais de um turno no período escolhido, é exibida uma linha para cada um deles, o que possibilita uma comparação rápida da evolução da assiduidade de cada um dos turnos.

A assiduidade do semestre é representada por um gráfico de linhas, no qual o eixo X exibe as semanas do semestre e o eixo Y representa a respetiva assiduidade.

Além disso, são incluídas linhas horizontais de referência que indicam os limites superior e inferior para a deteção de outliers, bem como o valor mínimo de presenças definido pelo utilizador na secção de personalização.

Ao posicionar o rato sobre um ponto do gráfico de linhas de assiduidade, é exibida informação detalhadas relativas à aula registada nesse dia, conforme ilustrado na Figura 34.

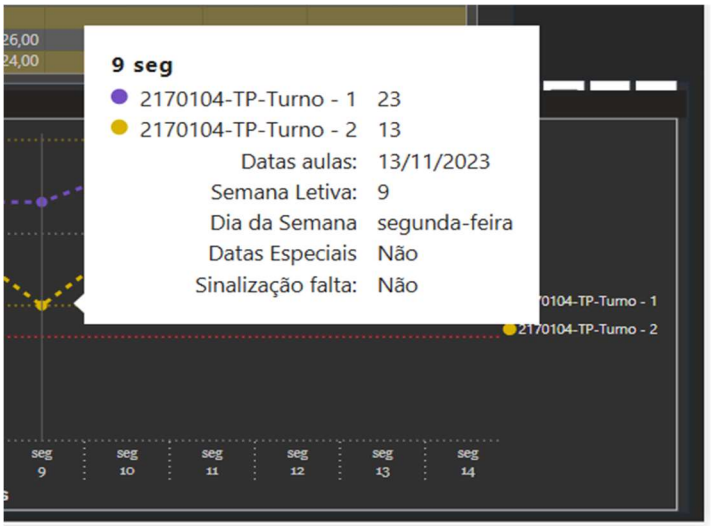


Figura 34 – *Dashboard* Assiduidade: Informação registo aula.

Para obter uma perspetiva sobre a evolução da assiduidade da UC ao longo dos semestres letivos, pode-se alternar entre o gráfico do semestre letivo e o histórico, com o uso do segmentador (2.1). O histórico de assiduidade é representado por um gráfico de barras, com a média de assiduidade por semestre, e gráfico de linha, representando o número de turnos, conforme Figura 35.

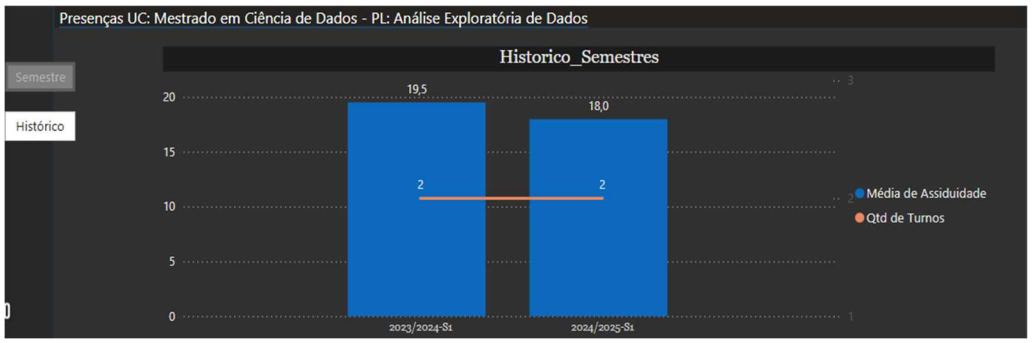


Figura 35 - *Dashboard* Assiduidade: Histórico assiduidade/semestre.

Através desse gráfico é possível avaliar a evolução da média de assiduidade da UC ao longo dos semestres e se existe uma relação entre essa média e a quantidade de turnos.

Imagens representativas de algumas interações disponíveis no *dashboard* de assiduidade podem ser vistas no Anexo C – *Interatividade dashboard* assiduidade.

Dashboard de Docentes

O *dashboard* de docentes foi desenvolvido para proporcionar uma análise detalhada da distribuição e do desempenho dos docentes ao longo do semestre, com foco no ciclo de estudo em que ministram aulas. Para isso, combina diferentes representações visuais que permitem uma interpretação abrangente dos dados, como pode ser consultado na Figura 36.

Figura 36 – *Dashboard* Docentes.

Os principais itens da componente de indicadores/informação desse painel são:

- **Total de Docentes do semestre** – Um indicador consolidado que apresenta o número total de docentes ativos no período, fornecendo uma visão global da dimensão do corpo docente;
- **Gráfico Total de Docentes por ciclo de estudo** – Representação visual que exibe a quantidade total de docentes alocados em cada um dos tipos de ciclos de estudo, possibilitando uma análise comparativa da distribuição do corpo docente entre os diferentes níveis de ensino;
- **Diagrama de Venn e tabela informativa** – O diagrama de Venn representa a distribuição dos docentes por tipo de ciclo de estudo (Licenciatura, Mestrado, TeSP), evidenciando interseções entre essas categorias e permitindo identificar graficamente se docentes lecionam em mais de um tipo de ciclo de estudo. Para a conceção deste diagrama, foi necessária a utilização de uma estrutura auxiliar, onde os dados foram reestruturados, além da definição de medidas específicas. Após esse processo, foi utilizado o visual *Venn Diagram da MAQ Software*, disponível na *Microsoft Store*, para representar graficamente as interações e sobreposições entre as diferentes

categorias de dados. Para complementar a análise visual, é utilizada uma tabela com a informação sobre as interações entre os docentes e os ciclos de estudo;

- **Top 3 Docentes**– Essa seção utiliza gráficos de barras para destacar os três docentes com maior número de aulas ministradas e os três docentes com maior carga horária atribuída, permitindo avaliar a distribuição da carga letiva e identificar docentes com maior volume de trabalho. A seleção dos ciclos de estudo é realizada por botões laterais.

Dashboard de Ciclos de Estudo

No *dashboard* de Ciclos de Estudo é possível fazer uma análise dos indicadores em função dos tipos de ciclos do curso. Na Figura 37 pode-se observar uma imagem do *dashboard* de Ciclos de Estudo.

Figura 37 - *Dashboard* Ciclos de Estudo.

Na parte superior são apresentados cartões com dados sobre a quantidade de cursos, docentes, UCs, turnos e a média de assiduidade por aula para o ciclo selecionado.

No centro, os cursos pertencentes ao ciclo selecionado são classificados e apresentados num ranking dos três primeiros, de acordo com o critério definido pelo utilizador: número de aulas, média de assiduidade, número de unidades curriculares (UCs) ou número de turnos.

Considerando que as componentes das UCs possuem características distintas, a média de assiduidade é calculada de forma segmentada, conforme o tipo de componente. Adicionalmente, a distribuição do número de aulas por regime de oferta é visualizada através de um gráfico do tipo *Brick Chart*, desenvolvido com o visual *Brick Chart by MAQ Software,* possibilitando uma análise visual da distribuição de aulas por regime.

Na secção inferior, são apresentados gráficos que ilustram a evolução dos principais indicadores — média de assiduidade, número de aulas e número de docentes — ao longo do semestre letivo, permitindo uma análise temporal detalhada do comportamento dessas variáveis.

Dashboard de Grupos UCs

Como explicado anteriormente, as unidades curriculares podem ser reunidas em grupos de UCs para otimizar a gestão de recursos.

Por isso motivo, foi desenvolvido um *dashboard* onde é possível monitorar esses grupos disciplinares. Uma imagem do dashboard de "Grupos Ucs" pode ser vista a seguir.

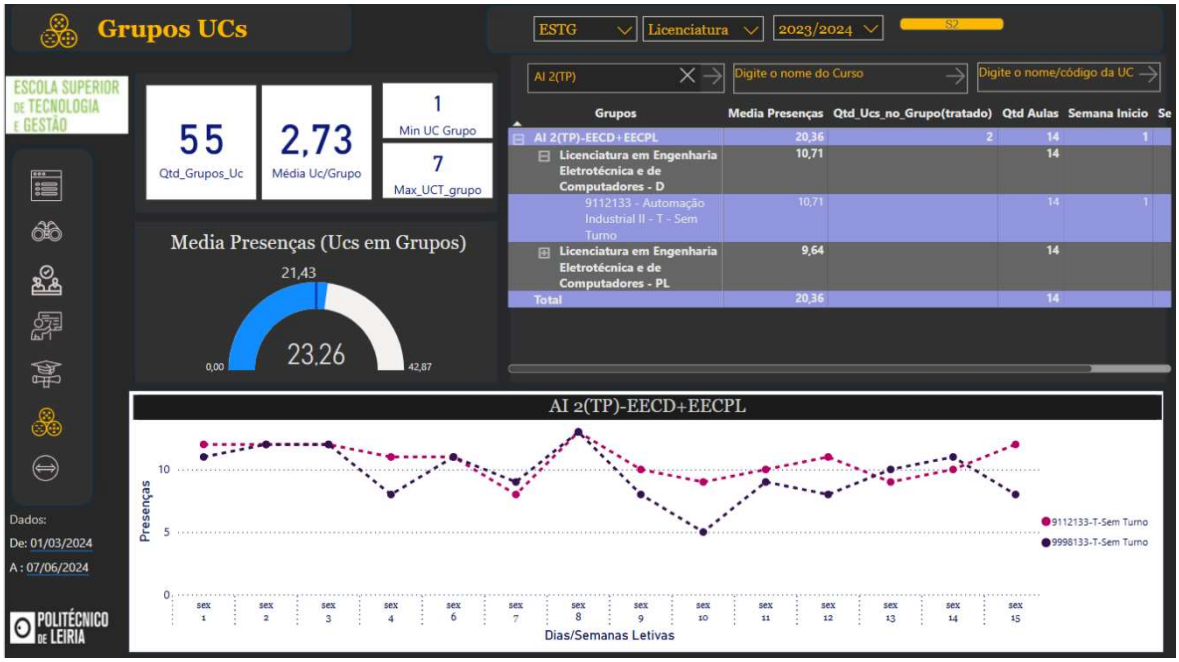


Figura 38 - *Dashboard* Grupos UCs.

Neste *dashboard*, a componente de indicadores/informação é apresentada da seguinte forma:

- **Cartões de Total de Grupos, Média Ucs/Grupo, Quantidade Mínima e Máxima de UCs por Grupo** – Indicadores visuais que apresentam, de forma

concisa, o número total de grupos existentes, bem como a média de Ucs por grupo, as quantidades mínima e máxima de UCs atribuídas a cada grupo;

- **Visual de Velocímetro** – Exibe a média de assiduidade dos grupos, com um marcador adicional que indica a média de assiduidade das UCs que não estão organizadas em grupo. Este componente visual facilita a comparação entre a assiduidade dos grupos e das UCs individuais;
- **Matriz de Grupos** – Apresenta uma tabela interativa onde é possível visualizar a quantidade de Ucs de cada grupo, a quantidade de aulas e as semanas de início e de fim do grupo. Adicionalmente, ao escolher um grupo é gerado em seguida, um gráfico de linhas que mostra a evolução da assiduidade das UCs que compõem o grupo;
- **Gráfico de assiduidades das Ucs do grupo -** Este gráfico permite uma análise temporal detalhada da assiduidade das Ucs que compõem o grupo, facilitando a identificação de padrões ou flutuações ao longo do semestre.

Dashboard de Equivalências

O *dashboard* de equivalências foi desenvolvido com o objetivo de apoiar o gestor na identificação de unidades curriculares semelhantes, permitindo a comparação da assiduidade entre elas e, eventualmente, contribuindo para a formação de grupos disciplinares.

É possível visualizar uma imagem do *dashboard* de equivalências na Figura 39.

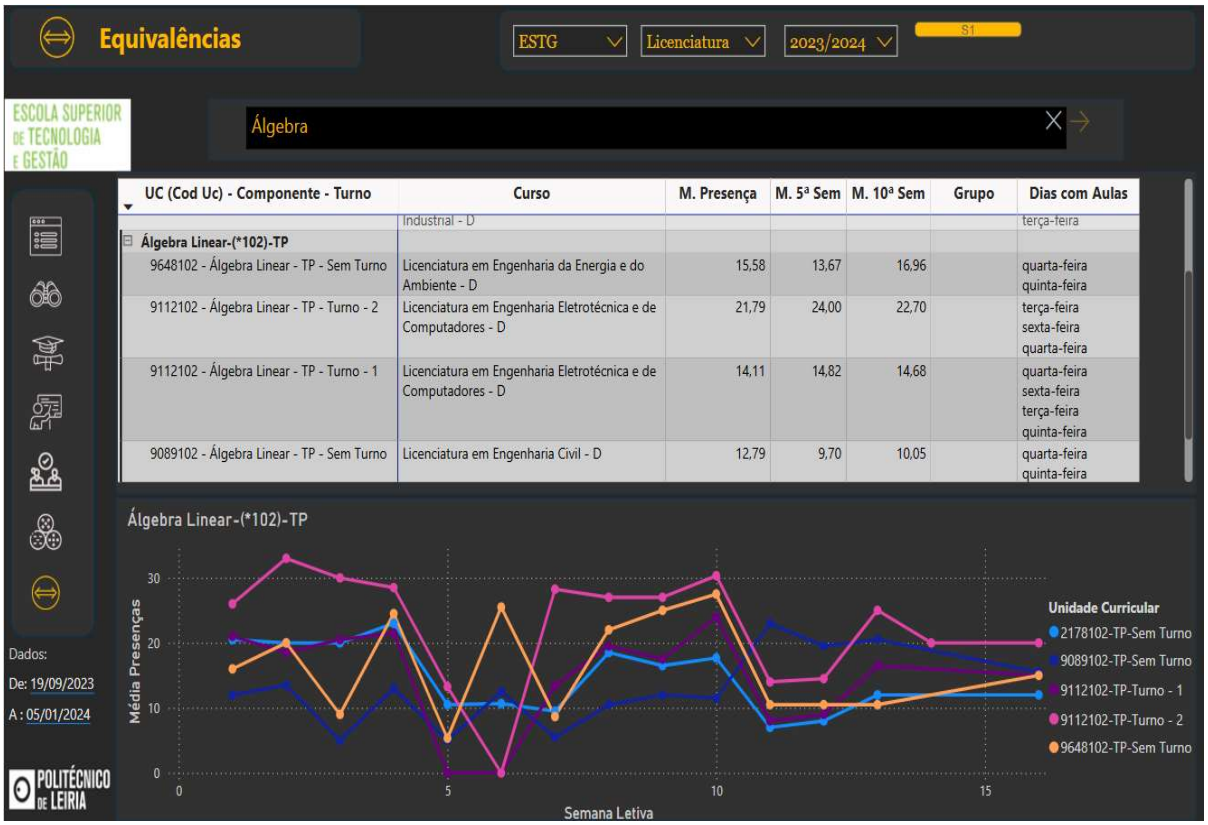


Figura 39 - *Dashboard* Equivalências.

Para conceção do *dashboard* de equivalências, duas UCs são consideradas semelhantes se possuírem mesmo nome, mesmo regime e mesma terminação do código de UC (últimos três dígitos do *uc_cod*).

Ao introduzir o nome de uma unidade curricular ou a terminação do respetivo código, são apresentadas as equivalências identificadas para o semestre letivo em análise. Para cada UC equivalente, é disponibilizada informação detalhadas, incluindo o nome e o curso a que pertencem, as métricas de assiduidade, a indicação de eventual integração em grupos disciplinares e os dias da semana em que possuem aulas registadas.

Para facilitar a análise comparativa, é gerado um gráfico de linhas que representa as médias de assiduidade das unidades curriculares equivalentes, permitindo uma visualização clara das variações existentes. Esta abordagem possibilita aos gestores um exame rápido e preciso das correspondências entre UCs, auxiliando na tomada de decisão relativa à organização dos grupos disciplinares e à otimização da gestão académica.

4.6. Avaliação de resultados

Ao longo deste projeto, o sistema de BI para análise de assiduidade foi integralmente desenvolvido, desde a fase conceptual até a implementação, abrangendo a arquitetura, a criação do DW e da área de *staging*, bem como a construção completa do processo de ETL e dos *dashboards*.

Na fase de conceção, foi identificada uma inconsistência nos dados fornecidos por uma API utilizada pela instituição. A correção proposta resolveu a falha, garantindo a adequada integração dos dados.

Os painéis e *dashboards* desenvolvidos permitem aos gestores monitorizar, de forma contínua, a evolução da assiduidade nos turnos das UCs, possibilitando a tomada de decisões atempadas e fundamentadas sobre o encerramento ou desdobramento de turnos. Essa análise pode ser realizada não só nas semanas críticas, mas ao longo de todo o semestre

Além disso, é possível realizar análises sobre os ciclos de estudos e sobre os docentes, garantindo uma visão integrada das atividades letivas.

O sistema permite o ajuste de parâmetros para a realização de análises personalizadas, em função das necessidades específicas dos utilizadores, conferindo flexibilidade e versatilidade à solução. Com a funcionalidade de evolução temporal, é possível a realização de uma análise detalhada da progressão da assiduidade ao longo dos semestres analisados, oferecendo uma visão clara das tendências e variações dos dados.

A solução final oferece um conjunto de ferramentas que possibilitam a emissão de alertas visuais, facilitando a chamada de atenção do utilizador para a informação mais relevante e com base nos dados parametrizados. Além de fornecer *insights* detalhados sobre assiduidade, a solução também apoia o gestor na formação de grupos disciplinares, permitindo a identificação de UCs com características semelhantes, otimizando a organização e o planeamento académico.

A modularidade aliada à utilização das linguagens *Python* e SQL contribui de forma significativa para a manutenibilidade e escalabilidade do sistema, proporcionando maior flexibilidade e facilitando a adaptação a necessidades futuras.

A validação do sistema contou com a análise e as observações da Coordenadora do Departamento de Engenharia Informática da ESTG/IPL, garantindo que a interpretação dos

dados considerasse as especificidades académicas e evitando conclusões imprecisas ou descontextualizadas.

Este contributo destacou a importância de que as análises sejam conduzidas por *stakeholders* com um conhecimento aprofundado das especificidades dos cursos, de modo a assegurar uma interpretação precisa dos resultados.

De acordo com os resultados obtidos, pode-se concluir que o objetivo do projeto foi plenamente alcançado e que o sistema desenvolvido poderá contribuir para melhorar o controlo a gestão de assiduidade da instituição.

5. Conclusão e trabalhos futuros

Este projeto consistiu na conceção e implementação de um sistema de BI para auxiliar a análise de assiduidade e melhorar a gestão de turnos na ESTG/IPL.

No capítulo dedicado ao estudo prévio, foram estabelecidos os fundamentos essenciais para a compreensão do presente trabalho, através da construção de um enquadramento teórico sobre BI e a sua aplicação no contexto académico. A abordagem aos conceitos fundamentais, à arquitetura e aos benefícios dos sistemas de BI e a análise de tendências permitiu consolidar uma base teórica para as discussões subsequentes. Adicionalmente, a apresentação de estudos correlatos possibilitou a validação da relevância destas soluções na gestão académica, destacando a sua crescente adoção e o impacto das inovações tecnológicas no suporte à tomada de decisão em IES.

As fases de conceção e implementação representaram a componente prática do presente trabalho, evidenciando as dificuldades enfrentadas e as soluções adotadas no decurso do desenvolvimento do projeto.

A complexidade inerente ao processo de gestão de turnos foi uma dificuldade encontrada ao longo do projeto. A superação desta dificuldade foi possibilitada pelo apoio da Coordenadora do Departamento de Engenharia Informática da ESTG/IPL, cujo contributo se revelou fundamental para garantir a coerência e a fiabilidade da interpretação dos dados tratados ao longo do projeto.

A compreensão e o tratamento dos dados representaram um desafio significativo, uma vez que os dados relativos às aulas, extraídos do sistema académico da instituição, não se encontravam completamente organizados e estruturados, exigindo um esforço adicional para normalização e integração.

Foram desenvolvidos *dashboards* com uma interface interativa e personalizável, permitindo aos utilizadores monitorizar, de forma dinâmica, a evolução da assiduidade dos estudantes.

Estes instrumentos possibilitam a identificação de padrões, tendências e eventuais anomalias, o que pode orientar a tomada de decisão informada e a implementação de medidas corretivas atempadas, sempre que necessário.

O sistema de BI desenvolvido cumpre os objetivos estabelecidos no âmbito do projeto. Contudo, ao longo do processo de desenvolvimento, foram identificadas potenciais linhas de investigação para trabalhos futuros, nomeadamente:

Expansão para outras escolas do IPL – Prevê-se a implementação do sistema em outras escolas do Instituto Politécnico de Leiria, permitindo a gestão e monitorização da assiduidade num contexto mais amplo. Para garantir uma integração eficiente, será essencial a participação dos gestores e de equipe técnica das com conhecimento específico sobre as especificidades de cada escola para assegurar o adequado tratamento e normalização dos dados;

Desenvolvimentos de perfis de utilizadores - A implementação de perfis personalizados de utilizadores contribuirá para uma utilização mais personalizada do sistema, otimizando o seu potencial e garantindo que cada perfil de utilizador tenha as ferramentas necessárias para tomar decisões informadas, de acordo com o seu nível de acesso e função dentro da instituição;

Automatização do processo de atualização – A atualização foi parcialmente automatizada através de um *script*. No entanto, a sua eficiência pode ser melhorada caso o IPL disponibilize automaticamente os dados do calendário letivo. Essa integração permitirá ao sistema obter essa informação de forma autónoma e atualizar o DW sem necessidade de intervenção manual.

Bibliografia

Abu Hasan, N., Miskon, S., Ahmad, N., Mat Ali, N., Hashim, H., Syed Abdullah, N., Alinda Alias, R., & Aizaini Maarof, M. (2015). Business Intelligence Readiness Factors for Higher Education Institution. *ARPN Journal of Engineering and Applied Sciences*, *10*(23), 1–7. www.arpnjournals.com

Ahmed, I. (2024, Agosto 16). *Everything You Need To Know About The Cost Of Building A Data Warehouse*. Astera. https://www.astera.com/type/blog/building-a-data-warehouse-cost-estimation/

Al-Aqrabi, H., Liu, L., Hill, R., & Antonopoulos, N. (2015). Cloud BI: Future of business intelligence in the Cloud. *Journal of Computer and System Sciences*, *81*(1), 85–96. https://doi.org/10.1016/j.jcss.2014.06.013

AlKhatnai, M., & Shawyun, T. (2022). Powering HEI Survey System for Data Analytics. *Journal of Institutional Research South East Asia*, *20*(2), 64–93.

Amazon. (2023, Maio 25). *Qual é a diferença entre ETL e ELT?* https://aws.amazon.com/pt/compare/the-difference-between-etl-and-elt/

Amazon. (2025, Fevereiro 13). *Perguntas frequentes sobre o Amazon QuickSight*. https://aws.amazon.com/pt/quicksight/resources/faqs/

Apraxine, D., & Stylianou, E. (2017). Business Intelligence in a Higher Educational Institution - The case of University of Nicosia. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 1735–1746.

Arefin, M. S., Hoque, M. R., & Bao, Y. (2015). The impact of business intelligence on organization's effectiveness: An empirical study. *Journal of Systems and Information Technology*, *17*(3), 263–285. https://doi.org/10.1108/JSIT-09-2014-0067

Baptista, A., Lopes, P., & Caldeira, P. C. (2017). *Aplicação de Técnicas de Business Intelligence a Base de Dados Prosopográficas* [Dissertação de Mestrado]. Universidade de Evora.

Barcelar, R. R. (2012). *Banco de Dados: Introdução ao estudo de bancos de dados*. http://www.ricardobarcelar.com.br

Borra, P. (2024). Evaluation of Top Cloud Service Providers' Bi Tools: A Comparison of Amazon Quicksight, Microsoft Power BI, and Google Looker. *International Journal of Computer Engineering and Technology (IJCET)*, *15*(3), 150–156. https://doi.org/10.17605/OSF.IO/RKZG4

Boulila, W., Al-kmali, M., Farid, M., & Mugahed, H. (2023). A business intelligence based solution to support academic affairs: case of Taibah University. *Wireless Networks*, *29*(3), 1051–1058. https://doi.org/10.1007/s11276-018-1880-3

Breslin, M. (2004). Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. *Business Intelligence Journal • Winter*, *7*(1), 6–20.

Coronel, C., & Morris, S. (2016). *Database Systems: Design, Implementation & Management, 12th edition* (12.ª ed.). Course Technology.

Costa, T. R. M. da. (2024). *Estatística e Probabilidade*. Senac São Paulo.

Devlin, B. (2018). Business Intelligence Thirty Years of Data Warehousing BI Expert's Perspective: Have We Finally Gotten Self-Service Right? *Business Intelligence Journal*, *23*(1), 12–24.

El-Adaileh, N. A., & Foster, S. (2019). Successful business intelligence implementation: a systematic literature review. *Journal of Work-Applied Management*, *11*(2), 121–132. https://doi.org/10.1108/JWAM-09-2019-0027

ElMalah, K., & Nasr, M. (2019). Cloud Business Intelligence. *International Journal of Advanced Networking Applications*, *10*(06), 4120–4124. https://doi.org/10.35444/ijana.2019.100612

Engelen, A., Kube, H., Schmidt, S., & Flatten, T. C. (2014). Entrepreneurial orientation in turbulent environments: The moderating role of absorptive capacity. *Research Policy*, *43*(8), 1353–1369. https://doi.org/10.1016/j.respol.2014.03.002

Fernandes, F., Correia, J., & Pontes, A. (2023). Business Intelligence: Tendências e o Impacto das Tecnologias Emergentes. *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*. https://doi.org/https://doi.org/10.1007/978-3-662-46531-8

Few, S. (2006). *Information Dashboard Design: the Effective Visual Communication of Data.* O'Reilly.

Garani, G., Chernov, A. V., Savvas, I. K., & Butakova, M. A. (2019). A Data Warehouse Approach for Business Intelligence. *Proceedings - 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019*, 70–75. https://doi.org/10.1109/WETICE.2019.00022

Governo do Estado de Mato Grosso. (2006). *Conceitos de Business Intelligence*.

Grossmann, W., & Rinderle-Ma, S. (2015). *Fundamentals of Business Intelligence*. Springer.

Howson, C. (2014). *Successful Business Intelligence - Unlock the value of BI & Big Data, Second Edition* (2.ª ed.). McGraw-Hill Education.

Hussain, S., & Aradhya, M. (2024). Big Data Integration & Transformation: A Comparative Analysis of SnapLogic and AWS Glue. *8th IEEE International Conference on Computational System and Information Technology for Sustainable Solutions, CSITSS 2024*. https://doi.org/10.1109/CSITSS64042.2024.10816957

Inmon, W. H. (2005). *Building the Data Warehouse, Fourth edition* (4.ª ed.). Wiley.

Işik, Ö., Jones, M. C., & Sidorova, A. (2013). Business intelligence success: The roles of BI capabilities and decision environments. *Information and Management*, *50*(1), 13–23. https://doi.org/10.1016/j.im.2012.12.001

Jakhar, R., & Krishna, C. (2020). Business Intelligence: As a Strategic Tool for Organization Development (A Literature Review). *ANWESH: International Journal of Management and Information Technology*, *5*(1), 44–46. http://www.publishingindia.com/anwesh

Júnior, O. de G. F., de Carvalho, V. D. H., Barros, P. A. M., & Braga, M. de M. (2022). An Experience with Business Intelligence to Support the Academic Management at a Brazilian Federal University. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, *46*, 5–20. https://doi.org/10.17013/risti.46.5-20

Kimball, R. (1997). A dimensional modeling manifesto. *DBMS*, *10*(9), 58–70.

Kimball, R., & Ross, M. (2015). *The Data Warehouse Toolkit- The Complete Guide to Dimensional Modeling, Third Edition* (3.ª ed.). Wiley.

Kumar, A. V. K. (2021). *Business Intelligence Demystified: Understand and Clear All Your Doubts and Misconceptions About BI*. BPB Publications.

Loshin, D. (2013). *Business intelligence : the savvy manager's guide, 2nd Ed.* Elsevier.

Magaireah, A. I., Ali, N., & Sulaiman Hidayah. (2017). Theoretical Framework of Critical Success Factors (CSFs) for Business Intelligence (BI) System. *8th International Conference on Information Technology (ICIT)*, 455–463.

Manis, K. (2024, Junho 24). *Microsoft named a Leader in the 2024 Gartner® Magic Quadrant™ for Analytics and BI Platforms*. https://powerbi.microsoft.com/pt-pt/blog/microsoft-named-a-leader-in-the-2024-gartner-magic-quadrant-for-analytics-and-bi-platforms/

Mell, P. M., & Grance, T. (2011). *The NIST definition of cloud computing, Special Publication 800-145*. https://doi.org/10.6028/NIST.SP.800-145

Microsoft. (2023, Março 23). *Azure e Power BI Banco de Dados SQL do Azure e Power BI*. https://learn.microsoft.com/pt-br/power-bi/connect-data/service-azure-and-power-bi?utm_source=chatgpt.com

Mondal, S. (2022). Cloud Business Intelligence as a solution for empowering SMEs. *EPRA International Journal of Multidisciplinary Research*, *8*(9), 202–204. https://doi.org/10.36713/epra2013

Morais, P., & Castro Lopes, F. (2019). Implementing a Business Information System to Improve the Quality Assurance Mechanisms in a Portuguese Higher Education

Institution. *13th International Technology, Education and Development Conference (INTED2019)*, 5623–5632. https://doi.org/10.21125/inted.2019.1382

Munawar. (2021). Extract Transform Loading (ETL) Based Data Quality for Data Warehouse Development. *Proceedings of 2021 1st International Conference on Computer Science and Artificial Intelligence, ICCSAI 2021*, 373–378. https://doi.org/10.1109/ICCSAI53272.2021.9609770

Nasser, A., Zaied, H., Grida, M. O., & Hussein, G. S. (2018). Evaluation of Critical Success Factors for Business Intelligence Systems Using Fuzzy AHP. *Journal of Theoretical and Applied Information Technology*, *96*(19), 6406–6422. www.jatit.org

Nazri, S., Ashaari, M. A., Hanum, Y., Iskandar, P., & Bakri, H. (2020). The Impact of Business Intelligence Adoption on Organizational Performance Among Higher Education Institutions in Malaysia. *Proceedings of the First ASEAN Business, Environment, and Technology Symposium (ABEATS 2019)*, 48–51. https://doi.org/10.2991/aebmr.k.200514.011

Ranjan, J. (2009). Business Intelligence: concepts, components, techniques and benefits. *Journal of Theoretical and Applied Information Technology*, *9*(1), 60–70. www.jatit.org

Schlegel, K., Ganeshan, A., Pidsley, D., Sun, J., O'callaghan, G., Long, C., Quinn, K., Fei, F., Macari, E., & O'brien, J. (2024). *Magic Quadrant for Analytics and Business Intelligence Platforms*. https://www.gartner.com/doc/reprints?id=1-2HVUGEM6&ct=240620&st=sb

Seenivasan, D. (2022). ETL vs ELT: Choosing the right approach for your data warehouse. *International Journal for Research Trends and Innovation*, *7*(2), 2456–3315. https://doi.org/10.6084/m9.doione.IJRTI2202018

Sharda, R., Delen, D., & Turban, E. (2017). *Business intelligence, analytics, and data science: a managerial perspective, 4th Ed.* (4.ª ed.). Pearson.

Sherman, R. (2015). *Business intelligence guidebook: from data integration to analytics.* (1.ª ed.). Elsevier.

Skyrius, R. (2021). *Business Intelligence - A Comprehensive Approach to Information Needs, Technologies and Culture* (1.ª ed.). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-67032-0

Sorour, A., Atkins, A., Stanier, C., Alharbi, F., & Campion, R. (2020). Integrated dashboards with social media analysis capabilities for monitoring quality in higher education institutions. *Proceedings of EDULEARN20 Conference*, 2862–2870.

Tableau. (2025). *Integração entre o Salesforce e o Tableau*. https://www.tableau.com/pt-br/solutions/salesforce

Universidade de Coimbra. (2025). *Informação académica para Novo/a Estudante UC*. https://www.uc.pt/academicos/uc/index

Varouchas, E., Sicilia, M. ángel, & Sánchez-Alonso, S. (2018). Academics' perceptions on quality in higher education shaping key performance indicators. *Sustainability (Switzerland)*, *10*(12), 4752. https://doi.org/10.3390/su10124752

Villegas-Ch, W., Palacios-Pacheco, X., & Luján-Mora, S. (2020). A business intelligence framework for analyzing educational data. *Sustainability (Switzerland)*, *12*(14), 1–21. https://doi.org/10.3390/su12145745

Williams, R. A., Sheikh, N. J., Duman, G. M., & Kongar, E. (2022). Critical Success Factors of Business Intelligence Systems Implementation. *IEEE Engineering Management Review*, *50*(4), 88–97. https://doi.org/10.1109/EMR.2022.3197096

Yeoh, W., & Koronios, A. (2010). Critical success factors for business intelligence systems. *Journal of Computer Information Systems*, *50*, 23–32. http://hdl.handle.net/10536/DRO/DU:30033043

Yeoh, W., & Popovič, A. (2016). Extending the understanding of critical success factors for implementing business intelligence systems. *Journal of the Association for Information Science and Technology*, *67*(1), 134–147. https://doi.org/10.1002/asi.23366

Yilmaz, N., Demir, T., Kaplan, S., & Demirci, S. (2020). Demystifying Big Data Analytics in Cloud Computing. *Fusion ofMultidisciplinary Research, An International Journal (FMR)*, *1*(1), 25–36.

Anexo A – *Script* análise exploratória *(Python)*

```python
# Importação Bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

✓ 18.0s Python

```python
# Identificação do ficheiro que possui dados das aulas
ficheiro_json= 'base-aulas-preliminar.json'
dados_aulas = pd.read_json(ficheiro_json)

```

✓ 0.5s Python

```python
#Primeiros e últimos registos do DF
print (dados_aulas)
```

✓ 0.1s Python

Dados anonimizados

```python
#Estrutura do DF
dados_aulas.info()
```

✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29732 entries, 0 to 29731
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   id_aulas            29732 non-null  int64 
 1   cod_uc              29732 non-null  object
 2   unidade_curricular  29732 non-null  object
 3   curso               29732 non-null  object
 4   nome_curso          29732 non-null  object
 5   componente          29732 non-null  object
 6   cod_docente         29732 non-null  int64 
 7   nome_docente        29732 non-null  object
 8   turma               29732 non-null  object
 9   motivo_falta        29732 non-null  object
 10  estado              29732 non-null  object
 11  data                29732 non-null  object
 12  data_inicio         29732 non-null  object
 13  data_fim            29732 non-null  object
 14  carga               29732 non-null  object
 15  grupo_disciplinar   29732 non-null  object
 16  turno               29732 non-null  object
 17  n_alunos            29732 non-null  int64 
 18  escola              29732 non-null  object
dtypes: int64(3), object(16)
memory usage: 4.3+ MB
```

```
[29732 rows x 19 columns]
```

+ Código + Markdown

Adicionar Célula de Código

```python
# Filtro no DF para aulas da ESTG
aulas_estg = dados_aulas[dados_aulas['turma'].str.contains('estg', case=False, na=True)].reset_index(drop=True)
```

✓ 0.0s Python

```python
#Substituição ' ' por NaN para facilitar a análise
aulas_estg= aulas_estg.replace(' ', np.nan)
# Quantidade de valores ausentes nas colunas
print(f"Nulos/Ausentes:\n{aulas_estg.isna().sum()}")
```

✓ 0.0s Python

```
Nulos/Ausentes:
id_aulas                0
cod_uc                274
unidade_curricular      0
curso                   0
nome_curso              0
componente            274
cod_docente             0
nome_docente            0
turma                   0
motivo_falta        10969
estado                  0
data                    0
data_inicio             0
data_fim                0
carga                   0
grupo_disciplinar   10967
turno                   0
n_alunos                0
escola                  0
dtype: int64
```

```python
#Análise da estrutura do DF aulas da ESTG
aulas_estg.info()
```

0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12900 entries, 0 to 12899
Data columns (total 19 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id_aulas           12900 non-null  int64
 1   cod_uc             12900 non-null  object
 2   unidade_curricular 12900 non-null  object
 3   curso              12900 non-null  object
 4   nome_curso         12900 non-null  object
 5   componente         12900 non-null  object
 6   cod_docente        12900 non-null  int64
 7   nome_docente       12900 non-null  object
 8   turma              12900 non-null  object
 9   motivo_falta       12900 non-null  object
 10  estado             12900 non-null  object
 11  data               12900 non-null  object
 12  data_inicio        12900 non-null  object
 13  data_fim           12900 non-null  object
 14  carga              12900 non-null  object
 15  grupo_disciplinar  12900 non-null  object
 16  turno              12900 non-null  object
 17  n_alunos           12900 non-null  int64
 18  escola             12900 non-null  object
dtypes: int64(3), object(16)
memory usage: 1.9+ MB
```

```python
#Identificação valores Duplicados
print(f"Registros Duplicados: {aulas_estg.duplicated().sum()}")
```

0.1s Python

```
Registros Duplicados: 0
```

```python
# Sumário estatístico para a coluna n_alunoss
print("\nSumário estatístico da coluna n_alunos:")
print(aulas_estg['n_alunos'].describe())

```

0.0s Python

```
Sumário estatístico da coluna n_alunos:
count    12900.000000
mean        15.581628
std         15.508081
min          0.000000
25%          7.000000
50%         14.000000
75%         22.000000
max        560.000000
Name: n_alunos, dtype: float64
```

```python
# Datas  do primeiro e do último registo das aulas na ESTG
print (f"Data primeiro Registo: {aulas_estg['data_inicio'][0]}")
print (f"Data último Registo: {aulas_estg['data_inicio'][len(aulas_estg)-1]}")
```

0.0s Python

```
Data primeiro Registo: 01-11-2022 08:00:00
Data último Registo: 17-12-2022 14:30:00
```

```python
# Identificação número de aulas únicas
aulas_estg['id_aulas'].nunique()
```

0.0s Python

```
2065
```

+ Código + Markdown

Adicionar Célula de Código

```python
#Substituição ' ' por NaN para facilitar a análise
aulas_estg= aulas_estg.replace(' ', np.nan)
# Quantidade de valores ausentes nas colunas
print(f"Nulos/Ausentes:\n{aulas_estg.isna().sum()}")
```

0.0s Python

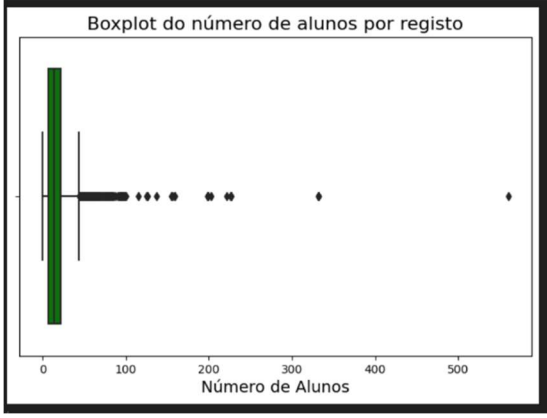
```python
# Análise das aulas com assiduidade = zero

qt_registos_0_alunos=((aulas_estg['n_alunos'] == 0)).sum()
print(f"Quantidade de registos 0_alunos: {qt_registos_0_alunos }")
qt_registos_0_alunos_com_motivo=((aulas_estg['n_alunos'] == 0) & (aulas_estg['motivo_falta'].notnull())).sum()
print(f"Quantidade de registos 0_alunos_com_motivo: {qt_registos_0_alunos_com_motivo }")
qt_registos_0_alunos_sem_motivo=((aulas_estg['n_alunos'] == 0) & (aulas_estg['motivo_falta'].isnull())).sum()
print(f"Quantidade de registos 0_alunos_sem_motivo: {qt_registos_0_alunos_sem_motivo }")
qt_registos_com_motivo_falta_sumario_atraso=((aulas_estg['n_alunos'] >= 0)  & (aulas_estg['motivo_falta']=='Sumário em atraso')).sum()
print(f"Quantidade de registos com sumário em atraso e zero alunos: {qt_registos_com_motivo_falta_sumario_atraso}")
```

0.0s Python

```
Quantidade de registos 0_alunos: 2287
Quantidade de registos 0_alunos_com_motivo: 1919
Quantidade de registos 0_alunos_sem_motivo: 368
Quantidade de registos com sumário em atraso e zero alunos: 119
```

```python
# Boxplot do n_alunos
plt.figure(figsize=(8, 5))
sns.boxplot(x=aulas_estg['n_alunos'], color='green')
plt.title('Boxplot do número de alunos por registo', fontsize=16)
plt.xlabel('Número de Alunos', fontsize=14)
plt.show()
```

0.4s Python




```python
#Quantidade de registos com mais de 100 alunos
qt_registros_mais_100_alunos = (aulas_estg['n_alunos'] > 100).sum()

print(f"Quantidade de registros com mais de 100 alunos: {qt_registros_mais_100_alunos }")
```

0.0s Python

```
Quantidade de registros com mais de 100 alunos: 25
```

```python
# Dados da aula com maior n_alunos
aulas_estg.loc[aulas_estg['n_alunos'].idxmax()]
```

0.0s Python

```
id_aulas
cod_uc
unidade_curricular
curso
nome_curso
componente
cod_docente
nome_docente
turma
motivo_falta
estado
data
data_inicio
data_fim
carga
grupo_disciplinar
turno
n_alunos                                                560
escola                                                 ESTG
Name: 8041, dtype: object
```

Dados anonimizados

###Docentes

```python
# Número de códigos de docentes únicos
qtd_docentes= aulas_estg['cod_docente'].nunique()

print(f"Número de códigos de docentes únicos: {qtd_docentes}")
```

0.0s Python

```
Número de códigos de docentes únicos: 441
```

```python
#Valores mínimos e máximos de cod_docente
min_cod_docente = aulas_estg['cod_docente'].min()
print(f"Mínimo código de docente: {min_cod_docente}")
max_cod_docente = aulas_estg['cod_docente'].max()
print(f"Máximo código de docente: {max_cod_docente}")
```

0.0s Python

```
Mínimo código de docente: 4
Máximo código de docente: 4682
```

```python
# Análise dos registos com Motivo_Falta
frequencia_motivo_falta = aulas_estg['motivo_falta'].dropna().value_counts()
df_motivos = frequencia_motivo_falta.reset_index()
df_motivos.columns = ['Motivo', 'Frequência']

# Adiciona uma coluna com a percentagem calculada
df_motivos['Percentagem'] = round ( (df_motivos['Frequência'] / df_motivos['Frequência'].sum()) * 100, 2)


print(df_motivos)
```

0.0s Python

	Motivo	Frequência	Percentagem
0	Feriados e Tolerâncias	1179	61.06
1	Interrupção atividades letivas - Receção ao Ca...	332	17.19
2	Outros	224	11.60
3	Sumário em atraso	119	6.16
4	Dispensas	44	2.28
5	Atestados Médicos	21	1.09
6	Artigo 134º, n.º2 e 3, LGTFP	7	0.36
7	Greve	2	0.10
8	Licença Parental	2	0.10
9	Não Aplicável	1	0.05

```python
#Qunatidade de UCs, de acordo com o o cod_uc
qtd_cod_uc = aulas_estg['cod_uc'].nunique()
print(f"Quantidade de unidades curriculares distintas: {qtd_cod_uc}")
```

✓ 0.0s Python

```
Quantidade de unidades curriculares distintas: 780
```

```python
# Quantidade de cursos, de acordo com o cod_curso
qtd_cod_curso = aulas_estg['curso'].nunique()

print(f"Quantidade de cursos distintos: {qtd_cod_curso}")
```

✓ 0.0s Python

```
Quantidade de cursos distintos: 67
```

```python
# Contar os turnos distintos, considerando as três colunas
turnos_distintos_totais = aulas_estg[['unidade_curricular', 'componente', 'turno']].drop_duplicates()

# Contar o número total de turnos distintos
total_turnos_distintos = turnos_distintos_totais.shape[0]

# Exibir o resultado
print(f"Total de turnos distintos: {total_turnos_distintos}")
```

✓ 0.0s Python

```
Total de turnos distintos: 1523
```

+ Código + Markdown

```python
# Filtra os registros para manter apenas os valores únicos de 'id_aulas' - A análise aqui deixa de ser por registo e passa a ser por aula
aulas_unicas = aulas_estg.drop_duplicates(subset='id_aulas').reset_index()

# converte carga para float
aulas_unicas['carga'] = aulas_unicas['carga'].astype(str).str.replace(',', '.').astype(float)

```

✓ 0.0s Python

```python
qtd_aulas = aulas_unicas['id_aulas'].nunique()
sum_carga=aulas_unicas['carga'].sum()
```

✓ 0.0s Python

```python
# Média do número de aulas/docente
média_aulas_docente=round(qtd_aulas/qtd_docentes,2)
print(f"Media de Aulas/Docente: {média_aulas_docente}")
```

✓ 0.0s Python

```
Media de Aulas/Docente: 27.36
```

```python
# Média de Carga horária/Aula
media_carga_aula=round(sum_carga/qtd_aulas,2)
print(f"Media de Carga Horária/Aula: {media_carga_aula}")
```

✓ 0.0s Python

```
Media de Carga Horária/Aula: 2.28
```

```python
# Média de Carga Horária/Docente
media_carga_docente=round(sum_carga/qtd_docentes,2)
print(f"Media de Carga Horaria/Docente: {media_carga_docente}")
```

✓ 0.0s Python

```
Media de Carga Horaria/Docente: 62.42
```

Anexo B – Código Dimensão Calendário (M)

```
let
    // Fontes: fato_aula e dim_ano_letivo
    fatoAula = Fato_Aula,
    dimAnoLetivo = tab_ano_letivo,

    //  datas mínima e máxima
    dataMin = List.Min(fatoAula[data_aula]),
    dataMax = List.Max(fatoAula[data_aula]),

    // Cria tabela de datas
    listaDatas = List.Dates(dataMin, Duration.Days(dataMax - dataMin) + 1, #duration(1, 0, 0, 0)),
    calendario = Table.FromList(listaDatas, Splitter.SplitByNothing(), {"Data"}),

    // Calendario com Ano Letivo
    TabelaComAnoLetivo = Table.AddColumn(
        calendario,
        "ano_letivo",
        each let
            DataAtual = [Data], // A data da linha atual da tabela Calendário
            AnoProximo =
                List.First(
                    List.Sort( // Ordena a lista de anos letivos em ordem decrescente
                        List.Select(
                            dimAnoLetivo[ano_letivo_cod], // Seleciona a coluna ano_letivo
                            each dimAnoLetivo[semestre_1_inicio]{List.PositionOf(dimAnoLetivo[ano_letivo_cod], _)} <= DataAtual // Condição: Menor ou igual à data atual
                        ),
                        Order.Descending // Ordenação decrescente
                    ),
                    null // Valor padrão caso a lista esteja vazia
                )
        in
            AnoProximo

    ),

    // Join entre a tabela Calendário e dim_ano_letivo usando a coluna ano_letivo_cod
    TabelaComAnoLetivoJoined = Table.Join(
        TabelaComAnoLetivo, // Tabela à esquerda (Calendário com Ano Letivo)
        "ano_letivo", // Coluna de junção na tabela à esquerda
        dimAnoLetivo, // Tabela à direita (dim_ano_letivo)
        "ano_letivo_cod", // Coluna de junção na tabela à direita
        JoinKind.LeftOuter // Tipo de junção (Left Outer, mantém todas as linhas da tabela Calendário)
    ),

    // Formata o ano letivo pro padrão ano/ano
   // TabelaAnoLetivoFormatado = Table.AddColumn(TabelaComAnoLetivoJoined, "Ano_Letivo", each
    //    Text.Start(Text.From([Ano_Letivo_temp]), 4) & "/" & Text.End(Text.From([Ano_Letivo_temp]), 4)
   // ),

    TabelaComAnoLetivoReduzida= Table.SelectColumns(TabelaComAnoLetivoJoined ,
        {"Data", "ano_letivo", "semestre_1_inicio", "semestre_1_fim", "semestre_2_inicio", "semestre_2_fim"}),

    // Adiciona coluna 'semestre_letivo'
    TabelaComSemestreLetivo = Table.AddColumn(TabelaComAnoLetivoReduzida, "semestre_letivo", each
        let
            DataAtual = [Data]
        in
            if DataAtual >= [semestre_1_inicio] and DataAtual <= [semestre_1_fim] then "S1"
            else if DataAtual >= [semestre_2_inicio] and DataAtual <= [semestre_2_fim] then "S2"
            else if DataAtual > [semestre_1_fim] and DataAtual <= [semestre_2_inicio] then "S1-Int-Let"
            else "S2-Int_Let"
    ),

    TabelaComDataInicioPeriodo = Table.AddColumn(TabelaComSemestreLetivo, "data_inicio_periodo", each
        if [semestre_letivo] = "S1" then [semestre_1_inicio]
        else if [semestre_letivo] = "S2" then [semestre_2_inicio]
        else if [semestre_letivo] = "S1-Int-Let" then [semestre_1_fim] + #duration(1, 0, 0, 0)
        else [semestre_2_fim] + #duration(1, 0, 0, 0)
    ),

    // Adiciona a coluna 'semana_letiva'
    TabelaComSemanaLetiva = Table.AddColumn(TabelaComDataInicioPeriodo, "semana_letiva", each
        if [data_inicio_periodo] <> null then
            Number.RoundDown(Duration.Days([Data] - [data_inicio_periodo]) / 7) + 1
        else
            null
     ),


     TabelaCalendar_reduzida1= Table.SelectColumns(TabelaComSemanaLetiva,
     {"Data", "ano_letivo", "semestre_letivo", "data_inicio_periodo", "semana_letiva"}),


     #"Tipo Alterado" = Table.TransformColumnTypes(TabelaCalendar_reduzida1,{{"Data", type date}, {"semana_letiva", Int64.Type}}),
     #"Coluna Ano_Semestre_Letivo Inserida" = Table.AddColumn(#"Tipo Alterado", "ano_semestre_letivo", each Text.Combine({[ano_letivo], "-", [semestre_letivo]}), type text),


     datas_especiais=Tab_Datas_Especiais,
      TabelaComDatasEspeciais= Table.Join(
      #"Coluna Ano_Semestre_Letivo Inserida", // Tabela à esquerda (Calendário com Ano Letivo)
      "Data", // Coluna de junção na tabela à esquerda
      datas_especiais, // Tabela à direita (dim_ano_letivo)
      "data", // Coluna de junção na tabela à direita
      JoinKind.LeftOuter // Tipo de junção (Left Outer, mantém todas as linhas da tabela Calendário)
      ),
      TabelaCalendar_reduzida2= Table.SelectColumns(TabelaComDatasEspeciais,
          {"Data", "ano_letivo", "semestre_letivo", "data_inicio_periodo", "semana_letiva","tipo"}),

```

```
    tabela_datas_especiais_com_data = Table.AddColumn(
    TabelaCalendar_reduzida2,
    "data_especial",
    each if [tipo] = null
        then "Não"
        else [tipo] & " (" & Text.From([Data]) & ")",
    type text
),
    data_especial_com_data = Table.SelectColumns(tabela_datas_especiais_com_data ,
        {"Data", "ano_letivo", "semestre_letivo", "data_inicio_periodo", "semana_letiva","data_especial"}),
    #"Coluna Mesclada Inserida" = Table.AddColumn(data_especial_com_data, "ano_semestre_letivo", each Text.Combine({[ano_letivo], "-", [semestre_letivo]}), type text),
    //#"Colunas Renomeadas" = Table.RenameColumns(#"Coluna Mesclada Inserida",{{"Mesclado", "ano_semestre_letivo"}}),

    TabelaComDia_semana = Table.AddColumn(#"Coluna Mesclada Inserida", "dia da semana (número)", each Date.DayOfWeek([Data]) + 1, Int64.Type),

    TabelaComDiaSemanaExtenso = Table.AddColumn(TabelaComDia_semana, "dia da semana (por extenso)", each Date.ToText([Data], "dddd"), type text),
    TabelaComDiaSemanaAbreviado = Table.AddColumn(
                                  TabelaComDiaSemanaExtenso,
                                  "dia da semana (abreviado)",
                                  each Text.Start(Date.ToText([Data], "dddd"), 3),
                                  type text
                                  ),
    #"Colunas Renomeadas" = Table.RenameColumns(TabelaComDiaSemanaAbreviado,{{"Data", "data"}, {"semestre_letivo", "semestre_letivo(calc)"}, {"ano_semestre_letivo", "ano_semestre_letivo(concat)"},
            {"dia da semana (número)", "dia_semana (número)(calc)"}, {"dia da semana (por extenso)", "dia _semana (extenso)(calc)"}, {"dia da semana (abreviado)", "dia da semana (abreviado)(calc)"}})

in
    #"Colunas Renomeadas"
```

Anexo C – Interatividade *dashboard* assiduidade

O objetivo deste anexo é apresentar visualmente algumas das interações possíveis no *dashboard* de assiduidade, bem como a forma como os alertas visuais são gerados. As seleções efetuadas pelo utilizador estão assinaladas com retângulos vermelhos, enquanto os resultados obtidos são destacados com retângulos azuis.

A Figura 40 ilustra uma imagem do *dashboard* de assiduidade com a configuração dos parâmetros de filtro:

- **Escola**: ESTG;
- **Ciclo de Estudo:** Mestrado
- **Ano Letivo**: 2023/2023, **Semestre Letivo:** S1
- **Mínimo de Alunos:**10
- **Nome do Curso:** Mestrado em Ciência de Dados
- **UC:** Análise Exploratória

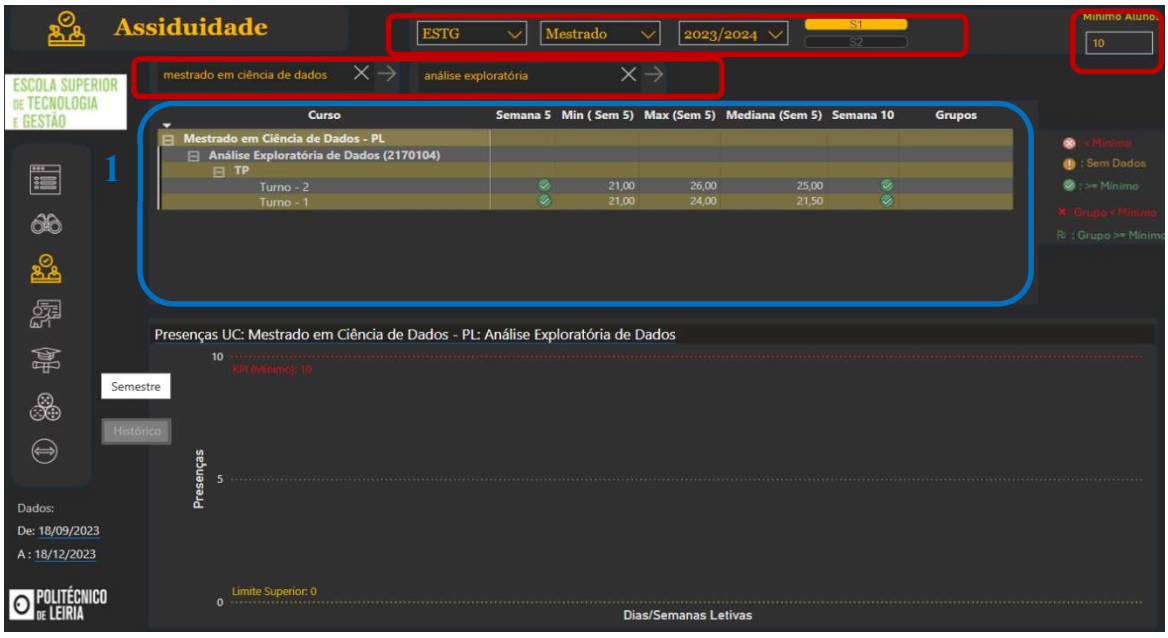


Figura 40 – Anexo C: *Dashboard* Assiduidade com marcação.

Após a definição dos parâmetros, a matriz (1) apresenta os dados de todas as UCs que satisfazem os critérios de filtragem. A lista encontra-se organizada numa hierarquia composta por curso, UC, componente e turnos.

A matriz inclui alertas visuais relativos ao cumprimento da meta de assiduidade até à semana crítica 5 e à semana crítica 10. Adicionalmente, são exibidas informações sobre os valores

mínimo, máximo e mediano de presenças até à semana 5, considerando todos os turnos filtrados. A matriz fornece ainda dados sobre a participação de cada turno nos grupos disciplinares.

Para uma análise mais detalhada, é possível visualizar graficamente a evolução da assiduidade das diferentes componentes da UC selecionada, bastando escolher uma das componentes diretamente na matriz. A Figura 41 ilustra essa interação.

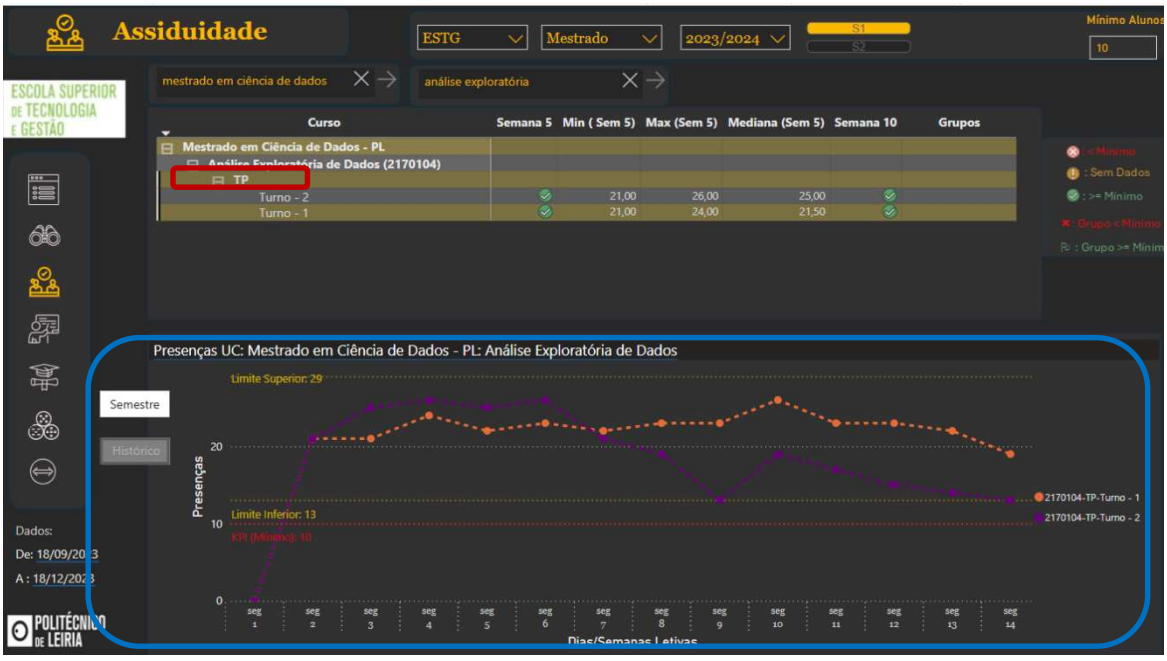


Figura 41 – Anexo C: *Dashboard* Assiduidade - Componente selecionada.

Dessa forma, um gráfico de linhas (2) é gerado automaticamente, representando a evolução da assiduidade ao longo do semestre letivo para todos os turnos da componente selecionada. Este gráfico permite uma análise visual comparativa da evolução da assiduidade entre os turnos da mesma componente da UC.

Além disso, é possível analisar individualmente cada turno, selecionando-o diretamente na matriz, conforme ilustrado na próxima figura.

Figura 42 – Anexo C: *Dashboard* Assiduidade - Gráfico linha para um turno.

Ao posicionar o rato sobre um dos pontos do gráfico de linhas, é apresentado um alerta com informações sobre a aula realizada nesse dia, permitindo uma análise de uma data em específico. Esta interação pode ser observada na figura seguinte.

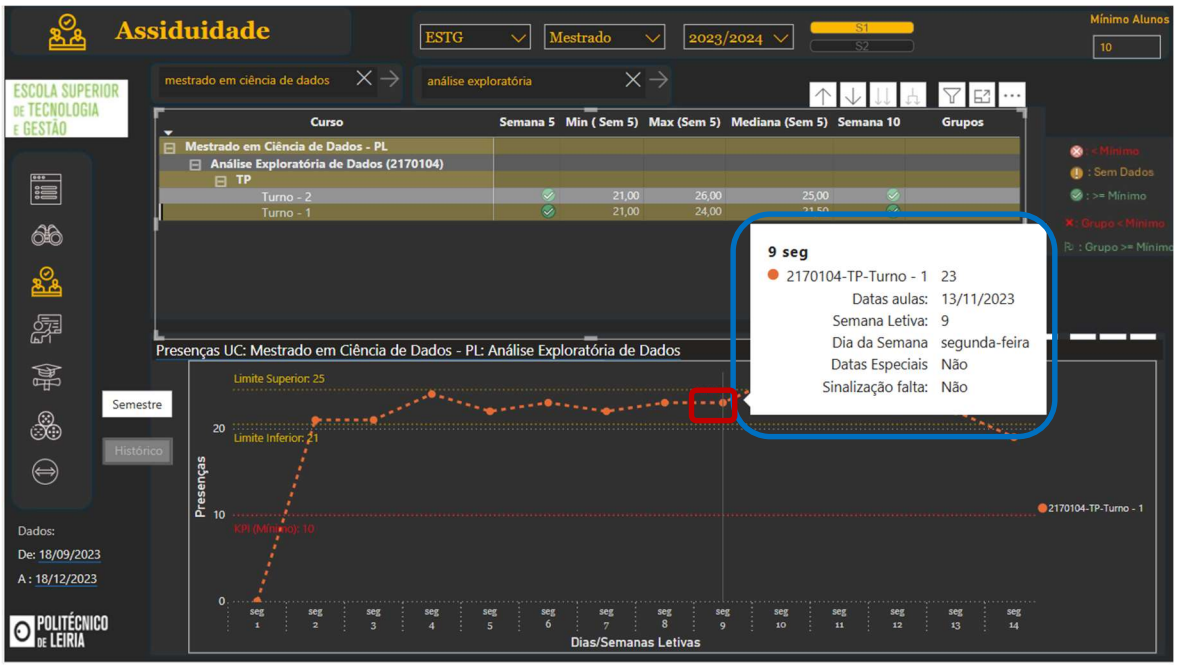


Figura 43 – Anexo C: *Dashboard* Assiduidade - Informação sobre aula.

O histórico da assiduidade da componente da respetiva UC ao longo de todo o período pode ser obtido ao selecionar a componente e, em seguida, a opção de histórico, conforme ilustrado na Figura 44.

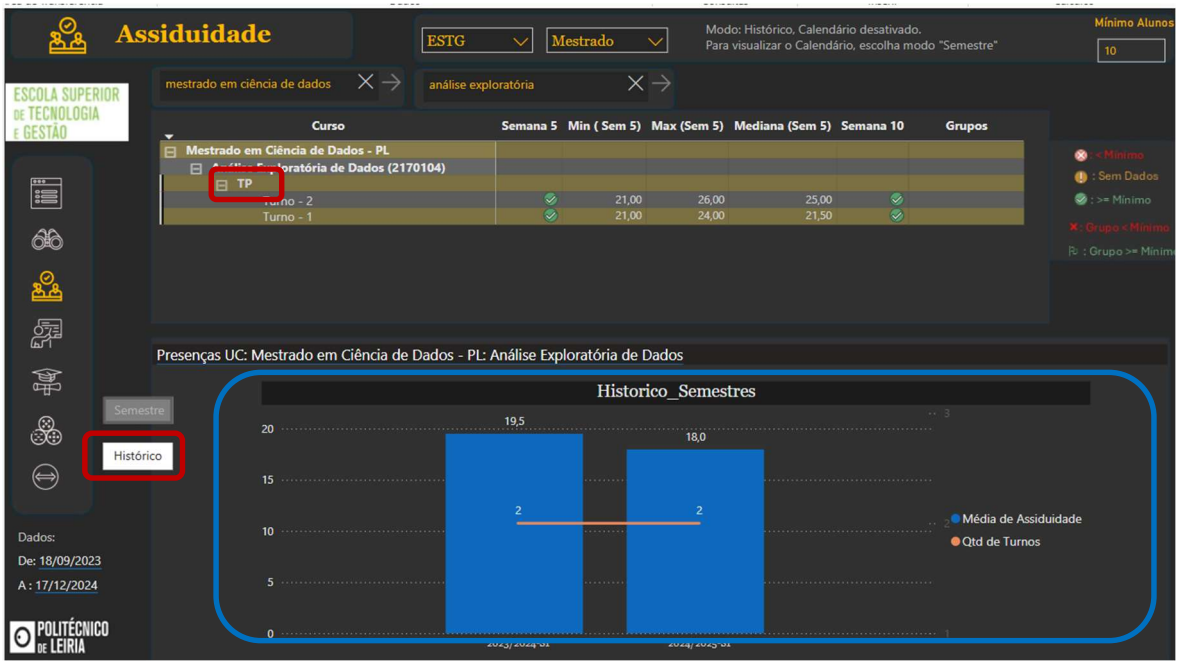


Figura 44 - Anexo C: *Dashboard* Assiduidade – Histórico.

A qualquer momento, o KPI do número mínimo de alunos pode ser ajustado pelo utilizador, resultando na atualização automática das informações relativas ao cumprimento da meta. Na Figura 45, observa-se a alteração do mínimo de alunos de 10 para 25.

Figura 45 - Anexo C: *Dashboard* Assiduidade - Alteração mínimo alunos.

Além dos alertas visuais (1) e (2), a linha de KPI mínimo no gráfico de linhas (3) é igualmente atualizada para refletir o novo valor definido.

As interações apresentadas neste anexo demonstram a flexibilidade e a capacidade analítica do *dashboard* de assiduidade, permitindo ao utilizador explorar os dados de forma dinâmica e personalizada. Através da seleção de componentes, turnos e ajustes nos parâmetros, é possível obter uma visão detalhada da evolução da assiduidade ao longo do semestre letivo e do histórico de semestres, facilitando a identificação de padrões e auxiliando na tomada de decisões informadas.