



# Painel ESG - Gestão Sustentável

DevScope Lda.

2024/2025

1220683 | Matilde Xavier Varela

**ISEP** INSTITUTO SUPERIOR  
DE ENGENHARIA DO PORTO



# **Painel ESG - Gestão Sustentável**

DevScope Lda.

**2024/2025**

**1220683 | Matilde Xavier Varela**



**Licenciatura em Engenharia Informática**

**Julho 2025**

**Orientador ISEP: Paulo Proença**

**Supervisor Externo: David Mota**



# Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, 2024/2025



# Dedicatória

Dedico este projeto a todos que me permitiram chegar a este capítulo da minha vida académica.



# Agradecimentos

Gostaria de iniciar esta secção do relatório a expressar a minha profunda gratidão às pessoas que me acompanharam ao longo deste percurso e contribuíram para a conceção deste projeto.

Em primeiro lugar, um sincero agradecimento ao professor Paulo Proença, cujo papel como orientador foi fundamental. A sua disponibilidade para rever o relatório inúmeras vezes permitiu-me apresentar uma versão mais refinada e estruturada deste trabalho.

Agradeço também ao David Mota, supervisor dos estágios na Devscope, a quem tive o prazer de conhecer na edição de *FallStack 2024*. Foi ele quem me selecionou para este estágio e acolheu os meus interesses em áreas específicas da informática, que mais tarde se integrariam na minha proposta de estágio.

Um agradecimento especial ao meu *buddy*, André Reis, pela orientação técnica e apoio especializado ao longo do projeto.

A todos os estagiários da Devscope, expresso a minha gratidão por tornarem esta experiência mais acolhedora e enriquecedora.

Quero também agradecer aos meus colegas de universidade, Rita Barbosa, Ana Guterres e Afonso Santos, que sempre me incentivaram e apoiaram durante a licenciatura. A sua amizade e motivação foram essenciais para o meu desenvolvimento académico e profissional.

Um agradecimento à Direção Geral de Ensino Superior, pelo apoio financeiro concedido através da bolsa de estudo durante os três anos da licenciatura, e à Câmara Municipal de Portimão, pelo apoio adicional durante dois anos. O contributo destas instituições foi crucial para que eu pudesse prosseguir os meus estudos.

Por fim, e com um carinho especial, quero agradecer aos meus pais, que, apesar dos desafios e dificuldades, sempre se esforçaram para que eu tivesse acesso ao ensino superior. Agradeço também ao meu namorado e à sua família, que têm sido um pilar fundamental de apoio para esta menina deslocada de casa.



# Resumo

Os temas ESG (*Environment, Social e Governance*) têm vindo a ganhar relevância nos últimos vinte anos, sobretudo no que respeita à divulgação de dados empresariais associados a estes domínios e às suas implicações. A questão do *greenwashing*, as discrepâncias nas classificações ESG fornecidas por várias agências, as abordagens para reduzir o impacto das empresas nestas dimensões e os avanços tecnológicos associados são, atualmente, temas de grande relevância. Este projeto visou a criação de um protótipo de aplicação que permite monitorizar o desempenho ESG de uma empresa.

O presente relatório, elaborado no âmbito da unidade curricular Projeto/Estágio do curso de Licenciatura em Engenharia Informática do Instituto Superior de Engenharia do Porto, apresenta o processo de desenvolvimento da plataforma ESG, realizado para a empresa Devscope, entidade responsável pela orientação do projeto. A ferramenta permite que as organizações carreguem conjuntos de dados, avaliem o desempenho ao nível ESG de várias formas, façam corresponder métricas específicas ao seu contexto, misturem métricas com base na estrutura SASB e estabeleçam objetivos em relação às mesmas métricas.

O desenvolvimento do projeto baseou-se em várias etapas. A primeira fase consistiu na análise do estado da arte, na qual se estuda o conceito de ESG e as suas implicações. Seguiu-se o processo de engenharia de *software*, na qual se procedeu à recolha e análise de requisitos, tanto funcionais como não funcionais. Com os requisitos delineados, procedeu-se à conceção do sistema, definindo a sua arquitetura com base nos modelos C4 e 4+1, bem como as suas funções. Por fim, foram efetuadas as fases de implementação e teste, bem como as evidências e a avaliação geral do que foi feito.

A solução criada cumpre os requisitos e objectivos definidos e é uma aplicação funcional como um *hub* centralizado de apoio à gestão e visualização de temas aplicáveis à ESG para a Devscope.

**Palavras-chave (Tema):** ESG (*Environment, Social, Governance*), Métricas, Matriz de Materialidade, SASB (*Sustainability Accounting Standards Board*), Conjuntos de Dados

**Palavras-chave (Tecnologias):** React, Next.js, ApexCharts, Railway, AWS S3, Jest



# Abstract

ESG issues (Environment, Social and Governance) have become increasingly important over the last twenty years, especially with regard to the disclosure of corporate data associated with these areas and their implications. The issue of greenwashing, discrepancies in ESG classifications provided by various agencies, approaches to reducing the impact of companies in these dimensions and the associated technological advances are currently highly relevant topics. This project aimed to create a prototype application to monitor a company's ESG performance.

This report, produced as part of the Project/Internship course of the Degree in Computer Engineering at the Instituto Superior de Engenharia do Porto, presents the process of developing the ESG platform for the company Devscope, the organisation responsible for guiding the project. The tool allows organisations to upload data sets, assess performance at ESG level in various ways, match specific metrics to their context, mix metrics based on the SASB framework and set objectives in relation to the same metrics.

The project was developed in several stages. The first stage consisted of analysing the state of the art, studying the concept of ESG and its implications. This was followed by the software engineering process, in which both functional and non-functional requirements were collected and analysed. Once the requirements had been outlined, the system was designed, defining its architecture based on the C4 and 4+1 models, as well as its functions. Finally, the implementation and testing phases were carried out, as well as the evidence and general evaluation of what had been done.

The solution created meets the requirements and objectives defined and is an application that functions as a centralised hub to support the management and visualisation of ESG issues for Devscope.

**Keywords (Theme):** ESG (Environment, Social, Governance), Metrics, Materiality Matrix, SASB (Sustainability Accounting Standards Board), Data Sets

**Keywords (Technologies):** React, Next.js, ApexCharts, Railway, AWS S3, Jest



# Conteúdo

<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xix</b>
<b>Lista de Código</b>	<b>xxi</b>
<b>Lista de Acrónimos</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento/Contexto . . . . .	1
1.2 Descrição do Problema . . . . .	2
1.2.1 Objetivos . . . . .	2
1.2.2 Abordagem . . . . .	2
1.2.3 Contributos . . . . .	3
1.2.4 Planeamento do trabalho . . . . .	3
1.3 Uso de Inteligência Artificial . . . . .	4
1.4 Estrutura do relatório . . . . .	4
<b>2 Estado da Arte</b>	<b>5</b>
2.1 ESG - História do Conceito e Aplicação . . . . .	5
2.1.1 <i>Who Cares Wins</i> . . . . .	5
2.1.2 PRI: Princípios de Investimento Responsável . . . . .	7
2.2 Trabalhos relacionados . . . . .	7
2.2.1 Práticas e Riscos Environmental, Social, and Governance (ESG): Impacto no Valor e Rentabilidade das Empresas . . . . .	8
2.2.2 Desafios e Abordagens no Reporte e Análise ESG . . . . .	9
2.2.3 Tendências atuais em ESG e Tecnologia . . . . .	10
2.3 Tecnologias existentes . . . . .	11
2.3.1 Frameworks de Relato ESG . . . . .	11
GRI (Global Reporting Initiative) . . . . .	11
SASB (Sustainability Accounting Standards Board) . . . . .	12
GRI e SASB: Diferenças-Chave e Complementaridade . . . . .	12
2.3.2 Softwares ESG e Soluções no Mercado . . . . .	13
Workiva . . . . .	13
SAP Sustainability Control Tower . . . . .	13
IBM Environmental Intelligence Suite . . . . .	14
Comparações entre as Soluções no Mercado . . . . .	14
2.3.3 Bibliotecas e Ferramentas de Desenvolvimento . . . . .	16
React.js . . . . .	16
Angular . . . . .	16
Vue.js . . . . .	16

<i>Vuexy</i> . . . . .	16
Comparação entre Bibliotecas e <i>Frameworks</i> de <i>Frontend</i> . . . . .	16
<b>3 Análise do Problema</b>	<b>19</b>
3.1 Contexto . . . . .	19
3.2 Domínio do problema . . . . .	19
3.2.1 Visão Geral da Aplicação . . . . .	22
3.3 Requisitos Funcionais e Não Funcionais . . . . .	23
3.3.1 Requisitos Funcionais . . . . .	23
UC-01   Visualização de KPI's . . . . .	24
UC-02   Filtragem de Métricas . . . . .	24
UC-03   Mapa de Materialidade . . . . .	25
UC-04   Pontuação ESG . . . . .	25
UC-05   Detalhes de Métricas . . . . .	25
UC-06   Métricas Customizáveis . . . . .	26
UC-07   Importação de Dados . . . . .	26
3.3.2 Requisitos Não Funcionais . . . . .	26
3.3.3 <i>Usabilidade (U)</i> . . . . .	27
3.3.4 <i>Confiabilidade (R)</i> . . . . .	27
3.3.5 <i>Performance (P)</i> . . . . .	27
3.3.6 <i>Suportabilidade (S)</i> . . . . .	27
3.3.7 + (Outros) . . . . .	27
<b>4 Desenho da solução</b>	<b>29</b>
4.1 Arquitetura do Sistema . . . . .	29
4.1.1 <i>Clean Architecture</i> . . . . .	29
4.1.2 Modelo C4 . . . . .	30
4.1.3 Modelo de Vistas 4+1 . . . . .	31
4.1.4 Vista de Processos . . . . .	32
4.1.5 Vista de Cenários . . . . .	36
4.1.6 Vista Lógica . . . . .	36
4.1.7 Vista de Implementação . . . . .	39
4.1.8 Vista Física . . . . .	39
4.2 Alternativa Arquitetural: DDD com <i>Clean Architecture</i> . . . . .	41
<b>5 Implementação da Solução</b>	<b>43</b>
5.1 Descrição da implementação . . . . .	43
5.1.1 Tecnologias Usadas . . . . .	43
Tecnologias de Pré-desenvolvimento . . . . .	43
Tecnologias de Desenvolvimento . . . . .	44
5.2 Funcionalidades . . . . .	45
5.2.1 Página Inicial   <i>Dashboard</i> . . . . .	45
5.2.2 Página das Métricas . . . . .	48
5.2.3 Mapa de Materialidade . . . . .	50
5.2.4 Página dos Conjuntos de Dados . . . . .	51
5.2.5 Página dos Objetivos . . . . .	54
5.3 Inicialização da Base de Dados . . . . .	56
5.4 Testes . . . . .	57
5.5 Avaliação da solução . . . . .	59

5.6	Implementação Alternativa . . . . .	61
<b>6</b>	<b>Conclusões</b>	<b>63</b>
6.1	Objetivos concretizados . . . . .	63
6.2	Trabalho Futuro e Limitações . . . . .	63
6.3	Considerações Finais . . . . .	64
6.3.1	Perspetiva Crítica do Problema definido e Solução proposta . . . . .	64
6.3.2	Considerações Pessoais . . . . .	65
	<b>Bibliografia</b>	<b>67</b>
<b>A</b>	<b>Cronograma do Projeto de Estágio</b>	<b>71</b>
<b>B</b>	<b>Mapeamento entre vistas de Implementação e Lógica (Nível 3)</b>	<b>73</b>
<b>C</b>	<b>Designs do Figma</b>	<b>75</b>
<b>D</b>	<b>Quadro Kanban do Desenvolvimento da Solução no Github Projects</b>	<b>83</b>



# Listas de Figuras

2.1	Principais intervenientes envolvidos na integração de temáticas ESG segundo o relatório <i>Who Cares Wins</i> (onValues 2005) . . . . .	6
2.2	Painel da Plataforma <i>SAP Sustainability Control Tower</i> (SAP 2025) . . . . .	14
3.1	Modelo de Domínio (autoria própria) . . . . .	21
3.2	Mapa de Materialidade segundo o setor SASB da Devscope ( <i>Software e serviços IT</i> ) (autoria própria) . . . . .	22
3.3	Vista de Cenários (autoria própria) . . . . .	24
4.1	Modelo Arquitetural <i>Clean Architecture</i> (Martin 2012) . . . . .	30
4.2	Modelo Arquitetural C4 (Brown 2025) . . . . .	31
4.3	Modelo de Vistas "4+1" (Kruchten 1995) . . . . .	32
4.4	Vista de processos dos casos de uso de consulta/visualização e filtragem (Nível 1) (autoria própria) . . . . .	32
4.5	Vista de processos dos casos de uso de consulta/visualização e filtragem (Nível 2) (autoria própria) . . . . .	33
4.6	Vista de processos dos casos de uso de consulta (Nível 3) (autoria própria) . . . . .	33
4.7	Vista de processo do caso de uso de consulta com acesso ao armazenamento na nuvem (Nível 3) (autoria própria) . . . . .	34
4.8	Vista de processo do caso de uso de filtragem (Nível 3) (autoria própria) . . . . .	34
4.9	Vista de processo dos casos de uso de criação e importação (Nível 1) (autoria própria) . . . . .	35
4.10	Vista de processo dos casos de uso de criação e importação (Nível 2) (autoria própria) . . . . .	35
4.11	Vista de processo do caso de uso de criação de métricas (Nível 3) (autoria própria) . . . . .	36
4.12	Vista de processo dos casos de uso de importação de dados (Nível 3) (autoria própria) . . . . .	36
4.13	Vista Lógica (Nível 1) (autoria própria) . . . . .	37
4.14	Vista Lógica (Nível 2) (autoria própria) . . . . .	37
4.15	Vista Lógica (Nível 3) (autoria própria) . . . . .	38
4.16	Vista de Implementação (Nível 3) (autoria própria) . . . . .	39
4.17	Diagrama de Nível 2 da vista física da solução (autoria própria) . . . . .	40
4.18	Diagrama de Nível 3 da vista lógica do <i>container Frontend</i> segundo uma arquitetura de camadas concêntricas (autoria própria) . . . . .	41
5.1	Página Inicial com pontuação ESG (autoria própria) . . . . .	45
5.2	Página das Métricas (autoria própria) . . . . .	48
5.3	Formulário de Criação de Métrica (autoria própria) . . . . .	49
5.4	Cartão com informação detalhada da Métrica (autoria própria) . . . . .	50
5.5	Página da Matriz da Materialidade (autoria própria) . . . . .	51
5.6	Página dos Conjuntos de Dados (autoria própria) . . . . .	52

5.7	Formulário de Importação de um Conjunto de Dados com Múltipla Renderização de Diagramas (autoria própria) . . . . .	52
5.8	Filtragem de Conjunto de Dados (autoria própria) . . . . .	53
5.9	Informação de um Conjunto de Dados (autoria própria) . . . . .	54
5.10	Página de Gestão de Objetivos (autoria própria) . . . . .	54
5.11	Cartão com Informação Detalhada de um Objetivo (autoria própria) . . . . .	55
5.12	Formulário de Criação de Objetivo (autoria própria) . . . . .	55
5.13	Pontuação ESG segundo o <i>script seedGood</i> (autoria própria) . . . . .	57
5.14	Pontuação ESG segundo o <i>script seedBad</i> (autoria própria) . . . . .	57
5.15	Testes desenvolvidos para a solução (autoria própria) . . . . .	58
5.16	Resumo do relatório de cobertura dos testes desenvolvidos da solução (autoria própria) . . . . .	59
5.17	Compilação das páginas em menos de 30 segundos (autoria própria) . . . . .	59
5.18	Execução do <i>script</i> de avaliação do tempo de compilação inicial (autoria própria) . . . . .	60
A.1	Cronograma do projeto de estágio (autoria própria) . . . . .	71
B.1	Mapeamento entre vistas de Implementação e Lógica (Nível 3) (autoria própria)	73
C.1	Página Inicial (Dashboard) da Plataforma ESG (autoria própria) . . . . .	75
C.2	Página das Métricas Customizáveis e de SASB (autoria própria) . . . . .	76
C.3	Painel de Criação de Métricas Customizáveis (autoria própria) . . . . .	77
C.4	Painel de Informação da Métrica (autoria própria) . . . . .	78
C.5	Página dos Conjuntos de Dados (autoria própria) . . . . .	79
C.6	Painel de Importação de Conjuntos de Dados (autoria própria) . . . . .	79
C.7	Matriz de Materialidade segundo o setor de Software e Serviços IT da Sustainability Accounting Standards Board (SASB) (autoria própria) . . . . .	80
C.8	Página dos Objetivos (autoria própria) . . . . .	80
C.9	Painel de Criação de Objetivos (autoria própria) . . . . .	81
D.1	Painel com quadro <i>kanban</i> usado no planeamento do projeto (autoria própria)	83

# **Lista de Tabelas**

2.1	Comparação entre GRI e SASB (autoria própria) . . . . .	13
2.2	Comparação entre soluções ESG digitais (autoria própria) . . . . .	15
2.3	Comparação entre bibliotecas e <i>frameworks</i> de <i>frontend</i> (autoria própria) . . . . .	17
3.1	Glossário do domínio do problema (autoria própria) . . . . .	20
3.2	Lista de Casos de Uso (autoria própria) . . . . .	23
3.3	Categorias FURPS+ (autoria própria) . . . . .	27
5.1	Categorias de pontuação ESG (autoria própria) . . . . .	46
6.1	Objetivos do Projeto e o seu Estado de Concretização (autoria própria) . .	63



# Listas de Código

5.1	Função responsável por obter os dados do ESG Score e do dashboard no carregamento do componente . . . . .	47
5.2	Componente mostrado no carregamento da página . . . . .	47
5.3	Componente mostrado no caso de erro no carregamento da página . . . . .	48
5.4	Excerto de Código com Filtragem das Métricas . . . . .	49
5.5	Função de <i>Upload</i> de uma imagem para o AWS S3 . . . . .	50
5.6	Classificação do progresso de uma métrica . . . . .	51
5.7	Chamada à API de Importação de um Conjunto de Dados . . . . .	53
5.8	Função de Filtragem de Conjunto de Dados . . . . .	53
5.9	Pedido POST para criar um novo Objetivo . . . . .	55
5.10	<i>Script</i> de limpeza da Base de Dados . . . . .	56
5.11	<i>Script</i> de <i>seeding</i> da Base de Dados . . . . .	56
5.12	Teste unitário ao componente <code>SwiperControls</code> . . . . .	58
5.13	Teste de integração entre <code>ChartErrorBoundary</code> e <code>ChartRenderer</code> . . . . .	58
5.14	<i>Script</i> de avaliação de performance da iniciação da aplicação desenvolvida .	60



# **Lista de Acrónimos**

<b>Sigla</b>	<b>Definição</b>
API	Application Programming Interface
CRS	Corporate Social Responsibility
ESG	Environmental, Social, and Governance
GEE	Gases de Efeito Estufa
GRI	Global Reporting Initiative
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IA	Inteligência Artificial
ISEP	Instituto Superior de Engenharia do Porto
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LEI	Licenciatura em Engenharia Informática
ODS	Objetivos de Desenvolvimento Sustentável
ONU	Organização das Nações Unidas
PESTI	Projecto / Estágio
PNL	Processamento de Linguagem Natural
PRI	Princípios para Investimento Responsável
RFID	Radio-Frequency Identification
SaaS	Software as a Service
SASB	Sustainability Accounting Standards Board
SPA	Single Page Application
UI	User Interface
UNEP FI	UNEP Finance Initiative
URL	Uniform Resource Locator



# Capítulo 1

## Introdução

Este Capítulo tem como objetivo apresentar o projeto, o trabalho realizado e a sua contextualização, abordando o problema em estudo e os objetivos a serem alcançados. Também será feita uma descrição das principais metodologias utilizadas ao longo do trabalho, bem como a identificação dos contributos e dos aspetos inovadores da solução desenvolvida. O Capítulo conclui com a apresentação da estrutura do presente documento.

### 1.1 Enquadramento/Contexto

Este documento descreve o trabalho realizado na unidade curricular de Projecto / Estágio (PESTI) da Licenciatura em Engenharia Informática (LEI) do Instituto Superior de Engenharia do Porto (ISEP) e constitui um dos elementos mais importantes do curso. Através dele, é atribuído um problema a ser resolvido, esperando-se uma solução documentada neste relatório. O principal objetivo deste projeto é desenvolver competências pessoais e profissionais, aplicando os conhecimentos adquiridos ao longo da licenciatura e preparando o estudante para a inserção no mercado de trabalho.

O presente projeto foi desenvolvido durante um estágio em ambiente empresarial na DevScope, uma empresa sediada no Porto, fundada em 2003, com mais de vinte anos de experiência em consultoria e desenvolvimento de software. A empresa especializa-se em tecnologias da Microsoft, nomeadamente *Power Platform, Portals* (Office 365 & SharePoint), *Web & App Development, AI & Machine Learning, Business Intelligence, Enterprise Integration, Cloud & DevOps* e *Training & Education* (DevScope 2025). Os seus serviços são aplicados em diversas áreas, incluindo saúde, retalho e setor imobiliário.

Nos últimos anos, as empresas têm demonstrado uma preocupação crescente com questões dos tipos *ESG*. Este foco tem levado as organizações a recolher e analisar dados que auxiliem numa tomada de decisão mais consciente e estratégica. No entanto, a falta de centralização no tratamento e visualização desses dados pode comprometer a performance empresarial e expô-la a riscos, como falhas de conformidade regulatória ou problemas éticos.

Assim, torna-se fundamental dispor de uma plataforma que compile e organize estes dados de forma acessível e estruturada, permitindo às empresas obter *insights* mais claros e estratégicos sobre as suas iniciativas ESG.

O presente projeto tem como objetivo o desenvolvimento do *frontend* desta plataforma, tornando-se o primeiro passo para a criação de uma ferramenta interna da DevScope. Além de contribuir para o desenvolvimento de um protótipo funcional, o projeto permite ao estudante aprender e aplicar ferramentas e tecnologias não abordadas no currículo académico, enriquecendo assim as suas competências técnicas e práticas.

## 1.2 Descrição do Problema

Nos últimos anos, a DevScope tem vindo a crescer exponencialmente, passando de uma empresa com um ambiente mais familiar para uma estrutura de maior dimensão. Com esse crescimento, surgiu a necessidade de uma plataforma que permita monitorizar e gerir de forma estruturada as iniciativas ESG.

Atualmente, a empresa enfrenta desafios na centralização e análise de métricas ESG, dificultando a identificação de padrões e a tomada de decisões informadas. Questões como consumo de eletricidade, temperatura do escritório e outras métricas ambientais, assim como indicadores sociais e de governança, precisam de ser monitorizados de forma mais acessível e eficiente.

A implementação desta plataforma permitirá um acompanhamento mais estruturado das métricas ESG, facilitando auditorias, identificando comportamentos que possam gerar custos desnecessários e garantindo uma maior transparência. Sem uma solução eficaz, a empresa pode enfrentar perdas financeiras devido a desperdícios operacionais, além de possíveis riscos reputacionais e regulatórios caso não consiga demonstrar conformidade com boas práticas ESG.

O projeto visa desenvolver a interface do utilizador para esta plataforma, garantindo uma experiência intuitiva e eficiente na visualização e gestão dos dados ESG, ao utilizar *frameworks* reconhecidas como a *Global Reporting Initiative (GRI)* ou *Sustainability Accounting Standards Board (SASB)* para a definição das métricas.

### 1.2.1 Objetivos

O foco do projeto é o desenvolvimento de um protótipo *frontend* de uma ferramenta interna da DevScope com o objetivo de rastrear e avaliar métricas ESG para levar a uma tomada de decisão mais informada, em alinhamento com as necessidades específicas de cada organização.

A nível técnico, o projeto tem como objetivos:

- **Consolidar dados de múltiplas fontes:** Combinar conjuntos relevantes de dados no mesmo sistema, fornecendo uma visão unificada e acessível.
- **Desenvolver uma aplicação configurável:** Permitir que as organizações configurem a ferramenta de acordo com as suas prioridades, sectores e objetivos estratégicos.
- **Visualização clara e interactiva dos dados ESG:** Conceber uma interface intuitiva que simplifique a análise e interpretação dos dados relacionados com ESG.
- **Definir e monitorizar objetivos baseados em métricas:** Facilitar o estabelecimento de objetivos personalizados de acordo com a informação ESG disponível, e simplificar a monitorização ao longo do tempo.

### 1.2.2 Abordagem

O desenvolvimento do projeto seguiu uma abordagem ágil, caracterizada pela iteratividade do *software* desenvolvido, pela compreensão das prioridades e pela capacidade de introduzir mudanças ao longo do processo. A flexibilidade perante os pedidos do cliente (DevScope) foi um dos principais fatores, promovendo uma comunicação próxima e frequente (Patel 2025).

O projeto fará ainda uso da *agile framework Scrum*, sendo o projeto dividido em iterações de uma semana (*sprints*) e diferentes fases, tais como: pré-planeamento do *sprint* (*release backlog* e *sprint goals*), planeamento do *sprint*/iteração (*sprint backlog*), implementação e demonstração das funcionalidades desenvolvidas (D. Cohen, Lindvall e Costa 2004).

Além das metodologias mencionadas, foi utilizado o *Microsoft OneNote*, uma ferramenta adotada pela DevScope como base de conhecimento colaborativa, desenvolvida pelos próprios colaboradores e acessível a toda a empresa. O *GitHub* foi empregado não apenas como repositório da solução desenvolvida, mas também como *hub* central do projeto, integrando o quadro *Kanban*. Esta abordagem, alinhada às metodologias ágeis, visa minimizar o tempo ocioso, promover um fluxo de trabalho mais eficiente e garantir um gerenciamento de tarefas estruturado, por meio de cartões detalhados que acompanham o progresso de cada etapa (Wakode, Raut e Talmale 2015).

As funções de *product owner* e *scrum master* foram exercidas por supervisores da empresa.

O estagiário teve flexibilidade horária, mas manteve comunicação contínua com a empresa. As reuniões diárias (*daily stand-ups*, 30 min) serviram para atualizar a equipa, enquanto as reuniões semanais garantiram acompanhamento mais detalhado com o supervisor, incluindo demonstrações e *feedback* para orientar o desenvolvimento.

### 1.2.3 Contributos

A solução desenvolvida oferece à DevScope uma ferramenta que centraliza e processa informação importante e permite todo um conjunto de benefícios, contribuindo para a mesma de diversas formas:

- **Aumento da produtividade:** Facilitar o acesso e análise das métricas ESG, agilizando o processo de tomada de decisões.
- **Tomadas de decisão mais informadas e estruturadas:** Fornecer dados confiáveis e organizados que suportem decisões empresariais baseadas em métricas claras.
- **Economia financeira:** Identificar comportamentos ineficientes que possam gerar custos desnecessários, promovendo maior eficiência.
- **Avaliação contínua de comportamentos empresariais:** Permitir a monitorização constante de práticas ESG, incentivando um comportamento mais responsável e sustentável.

Para o estudante, a solução desenvolvida representa o culminar de uma experiência enriquecedora, que exigiu a aquisição e o aprimoramento de novas competências. Além disso, proporcionou a sua integração num ambiente empresarial real, com as suas regras, processos, políticas e exposição à cultura organizacional.

### 1.2.4 Planeamento do trabalho

O projeto iniciou-se com uma fase de *Ramp Up* de três semanas, durante a qual são realizados *workshops* sobre as tecnologias utilizadas na DevScope. Após essa etapa inicial, o estágio seguiu um cronograma definido pela empresa, dividido em três fases principais: pesquisa e estado da arte, implementação da solução e refinamento do relatório e da solução. A escrita deste documento teve início após o período de *Ramp Up*.

O cronograma do estágio foi organizado da seguinte forma:

- **Ramp Up:** 24 de fevereiro a 14 de março
- **Estado da Arte:** 17 de março a 4 de abril
- **Implementação:** 14 de abril a 9 de maio
- **Refinamento e Revisão:** 19 de maio a 6 de junho

No Apêndice A, apresenta-se um diagrama de *Gantt* detalhado com o planeamento definido.

Devido ao aumento de complexidade do projeto desenvolvido, surgiram desvios do planeamento inicial, levando a que a submissão final fosse apontada para 6 de julho, de modo a concluir a implementação, testagem, escrita e refinamento do documento.

### 1.3 Uso de Inteligência Artificial

Nesta secção procura-se esclarecer, de forma transparente, a utilização da Inteligência Artificial (IA) durante a elaboração deste documento. A aplicação desta ferramenta foi restrita, esporádica e sempre em busca de dar clareza e coerência ao texto, sem comprometer o rigor analítico ou a autoria do texto.

As áreas em que foram utilizadas a inteligência artificial foram:

- As Tabelas 2.1, 2.2 e 2.3: toda a análise, recolha de informação e conclusões foram feitas manualmente. A inteligência artificial foi utilizada apenas para reorganizar o texto, melhorar a coesão dos resumos e facilitar a adaptação ao espaço disponível nas tabelas, sem alterar o conteúdo fundamental.
- A Secção 3.3.1, aborda a utilização de inteligência artificial na normalização da escrita dos requisitos funcionais. A informação e as especificações de cada requisito foram determinadas antecipadamente, sendo a inteligência artificial utilizada apenas para garantir que existia um formato e estilo uniformes nos vários casos de utilização.

### 1.4 Estrutura do relatório

O presente documento é composto por seis capítulos que descrevem o desenvolvimento da solução pretendida.

A **Introdução** contextualiza o projeto, apresenta o problema e as abordagens adotadas, destaca os contributos para os *stakeholders* e expõe o cronograma.

O **Estado da Arte** analisa trabalhos e tecnologias relevantes, explorando soluções existentes relacionadas com o problema em questão.

Os Capítulos de **Análise e Desenho da Solução** detalha a metodologia, as tecnologias e os algoritmos utilizados, além da modelação da solução e especificação de requisitos.

A **Implementação da Solução** descreve aspectos técnicos, como *software*, ferramentas e sistemas utilizados, bem como os testes realizados e a avaliação da solução.

Por fim, as **Conclusões** apresentam um balanço dos resultados obtidos, analisam as limitações encontradas e incluem uma reflexão crítica sobre o trabalho desenvolvido.

## Capítulo 2

# Estado da Arte

O estado da arte refere-se ao conhecimento atual sobre um determinado tema em análise ou estudo. Este Capítulo apresenta uma visão geral das origens do termo ESG, destacando os benefícios da adoção destas práticas, os desafios da sua implementação e monitorização, além de discutir o papel da tecnologia na gestão ESG.

Em seguida, é realizada uma análise dos trabalhos relacionados com a temática do projeto, acompanhada de uma compilação das tecnologias já existentes para a avaliação e gestão de métricas ESG.

### 2.1 ESG - História do Conceito e Aplicação

O conceito de ESG surgiu na década de **1990**, quando a Organização das Nações Unidas (ONU) passou a adotar uma postura mais aberta em relação ao setor corporativo. Durante esse período, o então Secretário-Geral da ONU, Kofi Annan, lançou as bases da iniciativa que levaria à criação do conceito de ESG. No contexto da missão das Nações Unidas de promover a paz e o desenvolvimento, aliada aos objetivos do setor empresarial de gerar riqueza e prosperidade, a organização começou a estabelecer parcerias estratégicas, reconhecendo o papel fundamental das empresas no avanço desses objetivos (Pollman 2024).

Em **1999**, durante o Fórum Econômico Mundial de Davos, Kofi Annan apresentou a proposta *Global Compact* (Pollman 2024), que se tornou operacional em **2000**, de interesse mútuo da ONU e do setor corporativo. De acordo com a ONU, trata-se de "um apelo às empresas para alinharem as suas estratégias e operações com princípios universais de direitos humanos, trabalho, meio ambiente e combate à corrupção, e tomem medidas que promovam objetivos sociais" (ONU 2025). Esta iniciativa tem por base dez princípios fundamentados nos direitos humanos, trabalho, meio ambiente e combate à corrupção <sup>1</sup>.

#### 2.1.1 Who Cares Wins

Em **2004**, Kofi Annan convidou algumas das principais instituições financeiras do mundo para se unirem à ONU numa nova iniciativa, como extensão do *Global Compact*: o projeto *Who Cares Wins*.

<sup>1</sup>Os princípios que constituem a iniciativa *Global Compact* estão disponíveis no site oficial *UN Global Compact*: <https://unglobalcompact.org/what-is-gc/mission/principles> (acesso a 31/03/2025).

Esta iniciativa reuniu, pela primeira vez, investidores institucionais, gestores de ativos, analistas de pesquisa *buy-side*<sup>2</sup> e *sell-side*<sup>3</sup>, consultores globais, órgãos governamentais e reguladores, com o objetivo de examinar o papel dos fatores ESG na gestão de ativos e na pesquisa financeira (Pollman 2024).

Como resultado, chegou-se a um consenso sobre o impacto dos fatores ESG no contexto de investimentos de longo prazo, e foi elaborado um relatório<sup>4</sup>, no qual o termo "ESG" foi introduzido oficialmente. O documento também apresentou recomendações para diferentes atores do mercado sobre como integrar questões ambientais, sociais e de governança corporativa na gestão de ativos, nos serviços de corretagem de valores mobiliários<sup>5</sup> e nas funções de pesquisa associadas (onValues 2005). Na Figura 2.1 estão representados os principais intervenientes envolvidos na integração de temáticas ESG segundo o relatório *Who Cares Wins* (onValues 2005).

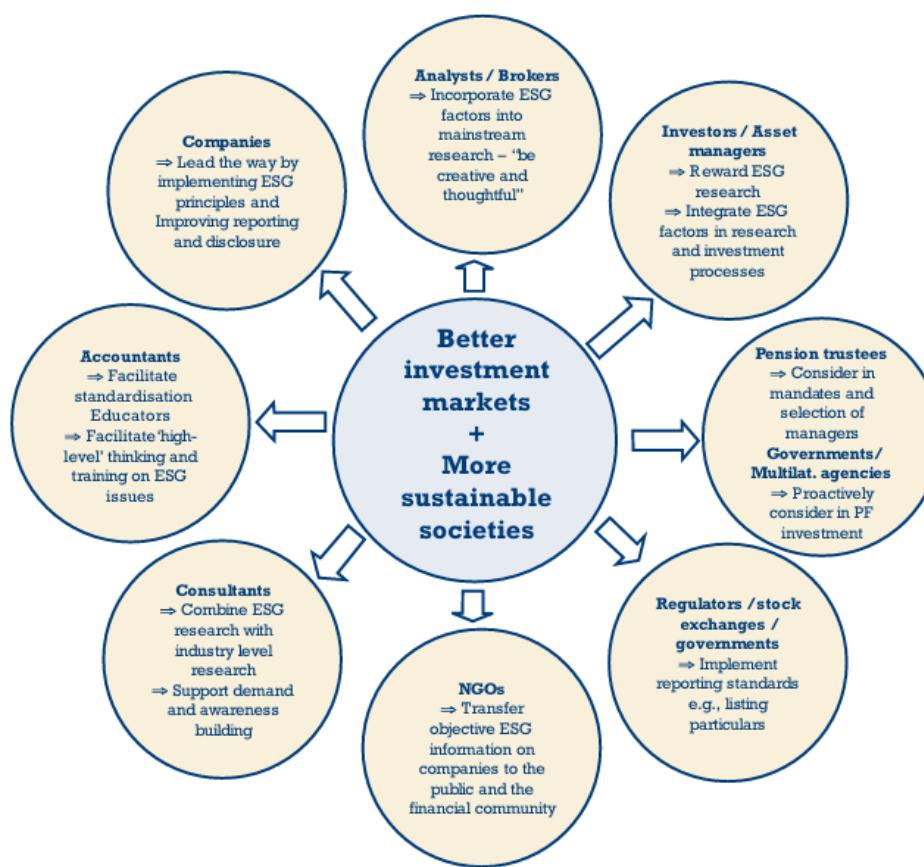


Figura 2.1: Principais intervenientes envolvidos na integração de temáticas ESG segundo o relatório *Who Cares Wins* (onValues 2005)

<sup>2</sup>Buy-side refere-se a instituições que compram ativos para investimento próprio, como fundos de investimento e seguradoras.

<sup>3</sup>Sell-side refere-se a instituições que vendem ativos e oferecem recomendações de investimento, como corretoras e bancos de investimento.

<sup>4</sup>Uma versão em PDF do relatório de 2004 está disponível em <https://www.ifc.org/content/dam/ifc/doc/mgrt/whocareswins-2005conferencereport.pdf> (acesso a 31/03/2025).

<sup>5</sup>Corretagem de valores mobiliários refere-se à intermediação na compra e venda de ativos financeiros, normalmente realizada por corretoras.

Segundo Pollman 2024, "embora o termo ESG tenha sido mencionado em menos de 1% das conferências de resultados financeiros nos anos seguintes ao relatório *Who Cares Wins*, em 2021 a sua presença cresceu significativamente, sendo citado em quase um quinto dessas conferências. Além disso, um estudo conduzido por *MarketWatch* (Carlson 2021), revelou que 72% dos investidores institucionais passaram a incorporar fatores ESG em suas estratégias".

### 2.1.2 PRI: Princípios de Investimento Responsável

Com a crescente conscientização dos problemas ESG, surgiram outras iniciativas, como os Princípios para Investimento Responsável (PRI). O PRI foi iniciado em **2005** por Kofi Annan, que convidou um grupo internacional de investidores institucionais a desenvolver iniciativas que refletissem a crescente importância das questões ESG nas práticas de investimento (Kim e Yoon 2022).

Os PRI mantêm uma forte ligação à ONU através dos seus dois parceiros fundadores: o *Global Compact* e a *UNEP Finance Initiative (UNEP FI)*. A iniciativa assenta em seis princípios<sup>6</sup> concebidos para reforçar a relação entre o investimento responsável e o desenvolvimento sustentável, alinhando o trabalho do PRI com os Objetivos de Desenvolvimento Sustentável (ODS)<sup>7</sup> e incentivando os seus signatários a adotar a mesma abordagem (PRI 2017).

No seu *website* oficial<sup>8</sup>, os PRI enumeram uma vasta gama de desafios, temáticas e externalidades ESG que as empresas devem considerar cada vez mais. Estes desafios abrangem três grandes dimensões:

- **Ambiente** – conservação da natureza, transição para uma economia circular, gestão sustentável da água, impactos do fraturamento hidráulico e emissões de metano.
- **Sociedade** – promoção da diversidade, equidade e inclusão, bem como condições de trabalho dignas.
- **Governança** – justiça tributária, engajamento político responsável, segurança cibernetica, remuneração executiva, propósito corporativo, combate à corrupção, proteção de denunciantes e nomeações de diretores.

## 2.2 Trabalhos relacionados

Esta secção tem como objetivo apresentar o estado da arte sobre ESG, abordando tanto os desafios e impactos desses fatores nas empresas quanto as soluções já desenvolvidas para sua análise e reporte.

<sup>6</sup>Os seis princípios para investimento responsável podem ser consultados no site oficial: <https://www.unpri.org/about-us/what-are-the-principles-for-responsible-investment> (acesso em 31/03/2025).

<sup>7</sup>As Nações Unidas disponibilizam um *website* com informações detalhadas sobre os ODS: <https://ods.pt/ods/> (acesso em 31/03/2025).

<sup>8</sup>O site oficial do PRI disponibiliza informações detalhadas sobre as várias medidas, práticas e desafios ESG. As **questões ambientais** podem ser consultadas em <https://www.unpri.org/sustainability-issues/environmental-social-and-governance-issues/environmental-issues>, as **questões sociais** em <https://www.unpri.org/sustainability-issues/environmental-social-and-governance-issues/social-issues> e as **questões de governança** em <https://www.unpri.org/sustainability-issues/environmental-social-and-governance-issues/governance-issues> (acesso em 31/03/2025).

Primeiramente, serão exploradas as práticas ESG e os seus efeitos no valor, rentabilidade e riscos das empresas, com base em estudos recentes sobre a relação entre desempenho financeiro e fatores ESG. Em seguida, serão analisadas as principais metodologias e abordagens utilizadas para reporte e análise ESG, destacando a diversidade de métricas e *frameworks* existentes e os desafios relacionados à sua padronização.

Posteriormente, serão discutidas as tendências atuais, com ênfase no impacto das novas tecnologias, como inteligência artificial e análise de dados em tempo real, na otimização da gestão.

### **2.2.1 Práticas e Riscos ESG: Impacto no Valor e Rentabilidade das Empresas**

O movimento por detrás da sigla ESG é bastante complexo, incorporando elementos de responsabilidade social, ambiental e governança corporativa para avaliar empresas. Os fatores de ESG são importantes para medir a sustentabilidade dos agentes econômicos e expandem o escopo do desempenho corporativo. Fatores externos, como o mercado e a indústria, e fatores internos, como a estrutura de propriedade e o conselho de administração, influenciam as práticas ESG na criação de valor (Wang et al. 2023).

Segundo uma análise conduzida por Whelan et al. 2021, em colaboração com a *Rockefeller Asset Management* e o *NYU Stern Center for Sustainable Business*, examinou-se mais de 1000 estudos publicados entre 2015 e 2020 sobre a relação entre fatores ESG e desempenho financeiro. Os resultados indicaram que 58% dos estudos identificaram uma relação positiva, 8% uma relação negativa, 13% nenhuma relação e 21% apresentaram resultados mistos. Embora a maioria dos estudos sugira um impacto positivo dos fatores ESG no desempenho financeiro, os autores concluem que os resultados refletem um debate contínuo sobre o tema.

Os estudos que identificaram uma relação positiva entre práticas ESG e desempenho financeiro apontam uma forte correlação entre a qualidade dos relatórios ESG e o valor da empresa. Isso sugere que a transparência, a confiança e a responsabilização dos *stakeholders* exercem uma influência positiva na valorização da empresa (Aydoğmuş, Gülay e Ergun 2022).

Por outro lado, os estudos que encontraram efeitos negativos destacam diferentes fatores. Entre eles, está o impacto adverso no desempenho financeiro devido à realocação de recursos dos acionistas (*shareholders*) para outras partes interessadas (*stakeholders*). Além disso, há evidências de que, em mercados emergentes, a relação entre pontuações ESG e retorno financeiro tende a ser negativa, possivelmente devido a desafios estruturais e regulatórios nesses mercados (Aydoğmuş, Gülay e Ergun 2022).

Os estudos sobre efeitos mistos mostram que relatar questões ambientais geralmente prejudica o desempenho financeiro da empresa. No entanto, a participação dos *stakeholders* na gestão está associada positivamente à dimensão social, enquanto a governança tem um impacto favorável no desempenho financeiro. Em geral, há uma correlação positiva entre os níveis de ESG e o valor da empresa, embora isso não se reflita diretamente na rentabilidade (Aydoğmuş, Gülay e Ergun 2022).

Segundo Aydoğmuş, Gülay e Ergun 2022, "a pontuação combinada de métricas ESG tem uma relação positiva e altamente significativa com o valor da empresa", porém a dimensão ambiental não acompanha esse mesmo padrão. O estudo sugere que, ao contrário dos componentes sociais e de governança, as iniciativas ambientais levam mais tempo, por vezes

anos, para gerar resultados concretos para a empresa. Além disso, os elevados custos de investimento associados às práticas ambientais podem representar um obstáculo, tornando esta dimensão menos atrativa em termos de impacto financeiro imediato. Estatísticas descritivas do estudo indicam que a média da pontuação ambiental tende a ser inferior às pontuações de governança e social, o que reforça a ideia de que esta métrica evolui de forma mais lenta e exige investimentos mais significativos.

De acordo com G. Cohen 2023, a pontuação ESG indica que empresas de maior dimensão tendem a ser menos poluentes. No entanto, apresentam menor preocupação com as implicações sociais de suas operações e enfrentam desafios crescentes em governança. O estudo destaca que, à medida que expandem as suas operações globalmente, os conglomerados passam a priorizar questões ambientais. Contudo, a governança torna-se mais complexa, devido às dificuldades no controlo corporativo de empresas internacionais de grande porte, evidenciando os desafios inerentes à sua gestão.

### 2.2.2 Desafios e Abordagens no Reporte e Análise ESG

Com o crescimento das preocupações em torno das métricas ESG, surgiram diversas classificações amplamente utilizadas, desenvolvidas por provedores de dados<sup>9</sup> ESG para auxiliar investidores na comparação e avaliação do desempenho das empresas nesses critérios. Inicialmente, os dados ESG eram obtidos a partir de fontes públicas, como relatórios financeiros e sites corporativos. No entanto, com o aumento das exigências de transparência, um número crescente de empresas passou a publicar relatórios anuais de *Corporate Social Responsibility (CRS)*, o que contribui para a ampliação da disponibilidade de dados ESG. Apesar deste avanço, a qualidade e confiabilidade das informações continuam a ser motivo de preocupação. Os indicadores divulgados frequentemente apresentam inconsistências entre as diferentes empresas, o que dificulta comparações diretas e resulta em divergências entre as agências de classificação ESG (Rau e Yu 2024).

A forma como as agências de classificação ESG avaliam as empresas depende das informações divulgadas e dos critérios adotados para a medição. No entanto, há divergências significativas em relação ao escopo, à ponderação dos fatores e aos métodos de avaliação utilizados por cada agência. O escopo e o peso determinam quais os aspectos que uma classificação ESG busca medir, enquanto a medição define a forma como esses aspectos são avaliados. A principal fonte de discrepância entre as classificações ESG está na divergência dos critérios de medição, refletindo diferenças nas perspetivas das agências sobre quais as categorias que devem ter maior relevância na avaliação do desempenho ESG de uma empresa (Berg, Kölbel e Rigobon 2022).

Em resposta à crescente demanda por informações não financeiras mais confiáveis, mensuráveis e transparentes, diversas *frameworks* de sustentabilidade foram desenvolvidas e implementadas. O objetivo comum destas estruturas é padronizar a divulgação de informações ambientais, sociais e de governança (ESG), permitindo maior comparabilidade entre as empresas. Embora a maioria destas *frameworks* seja de adoção voluntária e tenha um foco seletivo em alguns aspectos, elas buscam oferecer padrões de alta qualidade que diferenciem empresas genuinamente comprometidas com a melhoria de seu desempenho sustentável daquelas que praticam *greenwashing* (Cruz e Matos 2023).

---

<sup>9</sup>Entidades que atuam como intermediários na coleta, armazenamento e distribuição de dados, como provedores de serviços de *internet*, empresas de telecomunicações ou plataformas online.

O *greenwashing*<sup>10</sup> surge como um efeito colateral das preocupações das empresas com a sua imagem, ocorrendo quando estas tentam projetar uma reputação pró-sustentabilidade e afirmam adotar práticas ESG, mas falham em cumprir efetivamente as suas responsabilidades. Esta prática pode manifestar-se de diversas formas, incluindo narrativas ou divulgações seletivas e enganosas, alegações ambientais sem fundamento, certificações e rótulos duvidosos, entre outras estratégias que induzem os consumidores e investidores a percepções distorcidas sobre o real impacto da empresa (Rau e Yu 2024).

Segundo Schiemann e Tietmeyer 2022, uma maior divulgação ESG por parte das empresas está associada a níveis mais elevados de controvérsias ESG. Empresas envolvidas em controvérsias ESG enfrentam maior incerteza na precisão das previsões analíticas, afetando a confiança dos investidores. No entanto, a divulgação ESG desempenha um papel moderador nesta relação: uma comunicação transparente e detalhada pode mitigar a incerteza gerada pelas controvérsias, reduzindo os impactos negativos na percepção do mercado. Estes resultados possuem implicações práticas tanto para investidores, que buscam informações mais confiáveis, quanto para as empresas, que podem utilizar a divulgação ESG como estratégia para fortalecer a sua credibilidade.

### 2.2.3 Tendências atuais em ESG e Tecnologia

Um estudo realizado por Krueger et al. 2024 observou os efeitos de divulgação ESG na liquidez de ações das empresas, sendo a sua amostra constituída por 17 680 empresas únicas em 65 países no período entre 2002 e 2020. O artigo conclui que mandatos de divulgação ESG obrigatórios têm um efeito positivo significativo na liquidez de ações, particularmente quando implementados por instituições governamentais numa base de conformidade total e aplicados por instituições informais.

O crescimento da relevância deste tópico ao longo das últimas duas décadas ocorreu em paralelo com uma revolução tecnológica contínua. A digitalização das empresas tem proporcionado diversos benefícios às práticas ESG no contexto corporativo, resultando em melhorias significativas nas suas pontuações ESG. Entre as vantagens, destaca-se a redução dos custos de agência e o aumento das pontuações de governança e sociais. No entanto, não se observa uma correlação entre a digitalização das empresas e a melhoria das suas pontuações ambientais (Fang, Nie e Shen 2023).

A Indústria 5.0, ao focar na digitalização das empresas, oferece ferramentas para uma divulgação ESG mais eficaz. Diferente da Indústria 4.0, que priorizava aspectos económicos e técnicos, a Indústria 5.0 enfatiza a centralidade no ser humano, sustentabilidade e resiliência, alinhando-se diretamente aos princípios ESG. Ela melhora a autenticidade e abrangência das divulgações ESG, transformando relatórios retrospetivos em prospetivos e em tempo real, personalizando-os e expandindo o escopo para cadeias de suprimentos, além de reduzir custos e aumentar a eficácia das divulgações (Asif, Searcy e Castka 2023).

No âmbito técnico, a Indústria 5.0 engloba diversas tecnologias que otimizam o fluxo e o compartilhamento de informações, tais como sensores, *Radio-Frequency Identification*

---

<sup>10</sup>Top-Performing Singapore Firm Accused of Greenwashing in India Coal Sale"<https://www.bloomberg.com/news/articles/2022-11-09/top-performing-singapore-stock-with-temasek-backing-is-accused-of-greenwashing>

(RFID)<sup>11</sup> e *Internet of Things (IoT)*<sup>12</sup>; transações descentralizadas via *blockchain*<sup>13</sup>; processamento de grandes volumes de dados utilizando *big data* e computação em nuvem; tomada de decisão inteligente com apoio de *machine learning*, Inteligência Artificial (IA) e simulações; além da automação de processos com robôs e drones que podem ser aplicados a questões ESG (Asif, Searcy e Castka 2023).

A IA cresce rapidamente e é usada para resolver problemas de previsão, otimização e classificação, incluindo desafios ESG. Segundo Burnaev et al. 2023, a IA pode acelerar o processamento de dados e melhorar a compreensão das informações, contribuindo para ações ambientais, sociais e de governança. Exemplos incluem: IA a ajudar a atingir metas dos ODS, como ações climáticas e redução da poluição; uso em cidades inteligentes e redes elétricas; deteção de evasão fiscal e *greenwashing*; e, para investidores, a IA analisa o desempenho das empresas a partir de relatórios públicos através de Processamento de Linguagem Natural (PNL)<sup>14</sup>, que também automatiza tarefas como verificar a consistência de documentos e gerar relatórios.

## 2.3 **Tecnologias existentes**

A presente secção do relatório tem como objetivo documentar as tecnologias existentes. Inicia-se com uma análise das *frameworks* de relato ESG mencionadas na descrição do projeto, e inclui uma comparação entre estas. De seguida, realiza-se uma pesquisa sobre os principais *softwares* ESG disponíveis no mercado. Por fim, é feita uma breve menção ao *React* e às bibliotecas utilizadas na implementação do projeto.

### 2.3.1 **Frameworks de Relato ESG**

Diversas *frameworks* de sustentabilidade/ESG foram criadas para padronizar a divulgação de informações não financeiras, tornando-as mais fiáveis e acessíveis aos investidores. Apesar de serem maioritariamente voluntárias, estas estruturas facilitam a avaliação do impacto da sustentabilidade nas empresas e ajudam a distinguir compromissos reais de *greenwashing* (Cruz e Matos 2023).

#### **GRI (Global Reporting Initiative)**

A GRI é uma *framework* cujo público-alvo são os grupos de partes interessadas (*stakeholders*) (GRI e SASB 2021), e foi a primeira das iniciativas a surgir, tornando-se a principal referência para a elaboração de relatórios a nível mundial (Luque-Vilchez et al. 2023).

<sup>11</sup>RFID (*radio frequency identification*) é uma tecnologia sem fio que usa ondas de rádio para identificar de forma exclusiva objetos, animais ou pessoas.

<sup>12</sup>Internet das Coisas (IoT) refere-se a uma rede de dispositivos físicos, veículos, aparelhos e outros objetos incorporados com sensores, *software* e recursos de conectividade, permitindo-lhes coletar e trocar dados pela *Internet*.

<sup>13</sup>*Blockchain* é uma tecnologia de registro digital descentralizado, onde informações são armazenadas em blocos interligados, formando uma cadeia. Cada bloco contém dados e uma referência ao bloco anterior, tornando difícil alterar as informações sem afetar toda a cadeia, garantindo segurança e transparência nas transações.

<sup>14</sup>Processamento de Linguagem Natural (PNL) é um campo da inteligência artificial que se concentra na interação entre computadores e a linguagem humana. Isto envolve a capacidade dos sistemas computacionais de entender, interpretar e gerar texto ou fala de forma que seja útil para os utilizadores, aplicando técnicas como a análise de sentimentos, a tradução automática e o reconhecimento de voz.

A *framework* GRI consiste num sistema modular de normas que distingue entre requisitos ("shall"), recomendações ("should") e orientações (Adams et al. 2022). A GRI organiza-se em três tipos de normas<sup>15</sup>: normas universais, normas setoriais e normas para tópicos específicos (GRI 2025a). As organizações começam com as normas universais, utilizam depois as normas setoriais aplicáveis para determinar os tópicos materiais e relatam-nos utilizando as normas temáticas relevantes (GRI 2025b).

A estrutura modular da *framework* proporciona maior flexibilidade e uma abordagem mais detalhada na divulgação ESG das empresas. O seu objetivo principal é promover transparência e responsabilidade sobre os impactos empresariais, além de facilitar um diálogo mais informado sobre sustentabilidade corporativa. A GRI busca, assim, estabelecer um idioma comum para relatórios de impacto, auxiliando na construção de um futuro sustentável (Adams et al. 2022).

A GRI é uma *framework* em constante expansão. O seu programa *GRI Sector Program* tem como objetivo desenvolver normas específicas para 40 setores<sup>16</sup>, começando pelos que mais contribuem para as necessidades básicas e, posteriormente, expandindo-se para setores adjacentes (GRI 2025b).

### **SASB (Sustainability Accounting Standards Board)**

A SASB é uma *framework* cujo público-alvo são os investidores (GRI e SASB 2021). Embora seja de caráter voluntário, foi adotada globalmente e é composta por conjuntos de normas que padronizam 77 indústrias<sup>17</sup> (SASB 2025a). O foco da SASB é estabelecer e fornecer normas para a divulgação de informações sobre questões de sustentabilidade, facilitando a comunicação de dados não financeiros relevantes entre as empresas e os investidores (Goswami, Islam e Evers 2023). A *framework* considera os impactos no curto, médio e longo prazo sobre o valor das empresas.

Segundo Cruz e Matos 2023, um dos componentes mais importantes desta *framework* é o "Mapa de Materialidade"<sup>18</sup>, que oferece informações claras e acessíveis sobre quais as questões de sustentabilidade que são materiais para setores específicos. Isto permite que os investidores tenham acesso a informações diretamente, sem a necessidade de realizar uma análise extensa para avaliar a materialidade financeira de uma empresa nas questões ESG.

### **GRI e SASB: Diferenças-Chave e Complementaridade**

Como mencionado nas subsecções anteriores, tanto a GRI quanto a SASB são *frameworks* cujo objetivo é compilar informações de forma normalizada, visando melhorar a divulgação e a pontuação ESG das empresas, assim como avaliar seu valor, rentabilidade e performance consoante questões ESG.

---

<sup>15</sup>O site oficial da GRI detalha as normas da *framework*: [https://www.globalreporting.org/media/s4cp0oth/gri-gristandards-visuals-fig1\\_family-2021-print-v19-01.png](https://www.globalreporting.org/media/s4cp0oth/gri-gristandards-visuals-fig1_family-2021-print-v19-01.png) e <https://www.globalreporting.org/how-to-use-the-gri-standards/gri-standards-english-language/> (acesso em 04/04/2025)

<sup>16</sup>A GRI disponibilizou um documento sobre o *GRI Sector Program*, onde cataloga e agrupa setores de acordo com a sua prioridade: <https://www.globalreporting.org/media/mqznr5mz/gri-sector-program-list-of-prioritized-sectors.pdf> (acesso em 04/04/2025)

<sup>17</sup>O documento indicado pelo URL refere-se às normas SASB de divulgação ESG para a indústria de software e serviços *Information Technology (IT)*, onde se enquadra a Devscope. [https://d3f1raxduht3gu.cloudfront.net/latest\\_standards/software-and-it-services-standard\\_en-gb.pdf](https://d3f1raxduht3gu.cloudfront.net/latest_standards/software-and-it-services-standard_en-gb.pdf) (acesso em 04/04/2025)

<sup>18</sup><https://sasb.ifrs.org/standards/materiality-map/>

Em 2021, ambas as organizações por detrás das *frameworks* uniram-se para estudar o uso das suas normas no mercado, assim como averiguar as diferenças entre si e a sua complementaridade (GRI e SASB 2021). De acordo com GRI e SASB 2021, Pizzi, Principale e Nuccio 2023 e Antolín-López e Ortiz-de-Mandojana 2023, as diferenças as diferenças identificadas estão elencadas na Tabela 2.1.

Critério	GRI	SASB
<b>Aplicação da Materialidade</b>	Prioriza a divulgação de tópicos com impactos económicos, ambientais, sociais e de direitos humanos significativos	Foca em informações financeiramente relevantes que possam influenciar decisões de investimento e crédito
<b>Tipo e Âmbito da Divulgação</b>	Aborda os impactos económicos, ambientais e sociais das atividades da empresa no desenvolvimento sustentável global	Considera como os fatores sociais e ambientais afetam o valor da empresa, mas não o impacto da empresa no mundo
<b>Público-Alvo</b>	<i>Stakeholders</i> (partes interessadas)	Investidores ( <i>shareholders</i> )
<b>Processo de Definição das Normas</b>	Desenvolvidas por grupos de peritos representando diversos interesses globais, com transparência total e consulta pública.	Baseadas em pesquisa com participação de empresas, investidores e especialistas, com avaliação baseada em evidências e processo transparente.

Tabela 2.1: Comparação entre GRI e SASB (autoria própria)

Referenciando novamente o estudo conduzido por GRI e SASB 2021, diversos inquiridos consideram que ambos os conjuntos de normas são complementares, pois possibilitam uma divulgação mais abrangente das questões ESG a partir de diferentes perspetivas, abordagens à materialidade e grupos de interessados. Desta forma, as empresas conseguem atender tanto às necessidades dos investidores quanto das demais partes interessadas (*stakeholders*).

### 2.3.2 Softwares ESG e Soluções no Mercado

A presente subsecção explora soluções no mercado que tratam de métricas e relatos ESG.

#### Workiva

A Workiva oferece uma plataforma *cloud* para relatórios financeiros, ESG, auditoria e conformidade, permitindo a recolha, gestão e comunicação de dados de forma integrada e segura. A solução centraliza *workflows*, automatiza processos e assegura a rastreabilidade e conformidade com diversas *frameworks* e normas (Workiva 2025).

#### SAP Sustainability Control Tower

O *SAP Sustainability Control Tower* é uma solução *Software as a Service (SaaS)*<sup>19</sup> que permite às empresas registar, reportar e agir sobre métricas ESG com dados fiáveis e prontos para auditoria. Automatiza a integração de dados de múltiplas fontes, fornece cálculos avançados de emissões e incorpora *insights* de sustentabilidade nos processos empresariais

<sup>19</sup>O SaaS, ou *Software as a Service*, é um modelo de fornecimento de *software* baseado na nuvem (*cloud*) em que os utilizadores accedem às aplicações através da *Internet*, em vez de as instalarem localmente.

para uma gestão estratégica e transparente (SAP 2025). A Figura 2.2 ilustra o painel da Plataforma *SAP Sustainability Control Tower*.

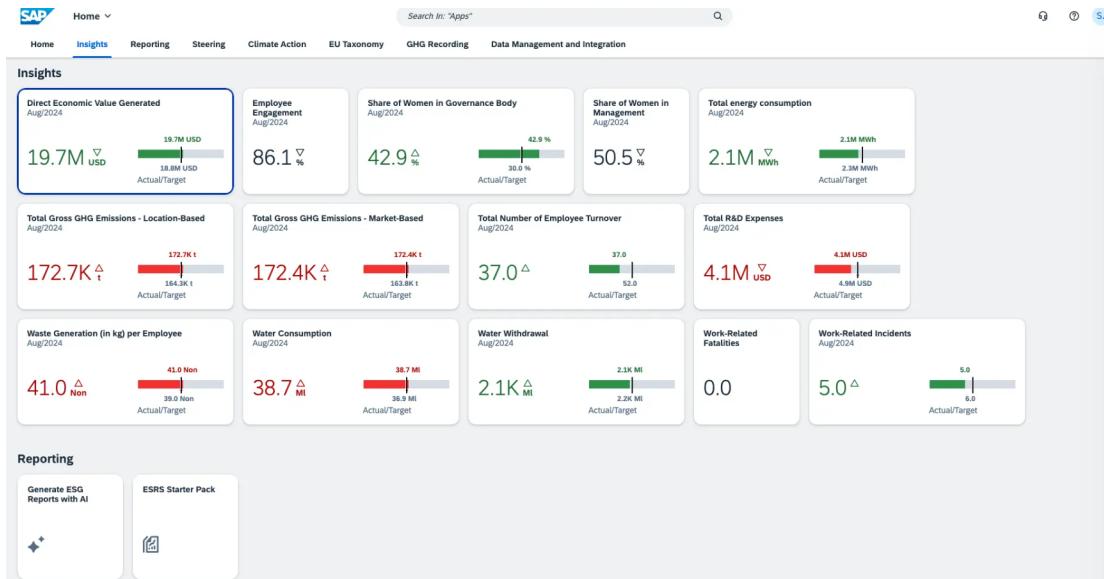


Figura 2.2: Painel da Plataforma *SAP Sustainability Control Tower* (SAP 2025)

## IBM Environmental Intelligence Suite

O *IBM Environmental Intelligence Suite* oferece várias *Application Programming Interface (API)* para integrar, analisar e visualizar dados ambientais, climáticos e de emissões de Gases de Efeito Estufa (GEE). Ao usar *machine learning* e *AI-driven insights*, a plataforma permite extrair informações estratégicas, prever impactos e garantir conformidade com normas de sustentabilidade, adaptando-se às necessidades específicas de cada empresa (IBM 2025).

## Comparações entre as Soluções no Mercado

Após a descrição individual das três soluções indicadas, é pertinente passar a uma análise comparativa direta entre elas, a partir da qual é possível evidenciar as suas diferenças e complementariedades. A Tabela 2.2 sintetiza essa comparação, com base em critérios como o tipo de plataforma, funcionalidades, nível de automatização, cumprimento de normas, capacidade de integração de dados e tecnologias utilizadas. Esta visão comparativa permite uma melhor compreensão da posição de cada solução no atual ambiente de soluções digitais para a gestão e reporte ESG.

Característica	Workiva	SAP Sustainability Control Tower	IBM Environmental Intelligence Suite
<b>Tipo de Plataforma</b>	Plataforma <i>cloud</i> integrada	Plataforma <i>SaaS</i>	Plataforma com múltiplas <i>APIs</i> e serviços modulares
<b>Funcionalidades Principais</b>	Relatórios financeiros, ESG, auditoria e conformidade	Registo, reporte e ação sobre métricas ESG, com cálculos de emissões e integração de dados	Análise, visualização e previsão de dados ambientais, climáticos e de emissões de GEE
<b>Automatização de Processos</b>	<b>Sim</b> , automatiza <i>workflows</i> e garante rastreabilidade	<b>Sim</b> , integração de dados automática e cálculos avançados de emissões	<b>Sim</b> , usa <i>machine learning</i> para previsões e <i>insights</i>
<b>Foco em Conformidade</b>	<b>Alto</b> , com várias <i>frameworks</i> e normas	<b>Alto</b> , foco em auditoria e conformidade normativa	<b>Alto</b> , foco em conformidade ambiental e climática
<b>Integração de Dados</b>	<b>Sim</b> , recolha e gestão integradas	<b>Sim</b> , múltiplas fontes e automatização	<b>Sim</b> , integração via <i>APIs</i> climáticas e ambientais
<b>Analises Avançadas / Insights</b>	<b>Limitado</b> , foco em reporting	<b>Médio</b> , oferece <i>insights</i> estratégicos	<b>Elevado</b> , previsão de impactos baseada em IA
<b>Tecnologias Empregues</b>	<i>Cloud</i> , <i>workflows</i> , gestão documental	<i>Cloud</i> , cálculos de emissões, integração com processos de negócio	<i>Machine learning</i> , IA, <i>APIs</i> ambientais
<b>Utilização Indicada Para</b>	Consolidação de relatórios e auditoria ESG e financeira	Estratégia empresarial com foco em sustentabilidade e métricas automatizadas	Adaptação baseada em previsões ambientais em tempo real

Tabela 2.2: Comparação entre soluções ESG digitais (autoria própria)

Com base na Tabela 2.2, observa-se que as três soluções analisadas apresentam um elevado grau de automatização, forte alinhamento com normas de conformidade e capacidade de integração de dados. No entanto, cada uma posiciona-se de forma distinta no mercado: a *Workiva* foca-se na consolidação de relatórios e auditoria, a *SAP* destaca-se pela visão estratégica e integração com processos empresariais, e a *IBM* oferece recursos avançados de análise preditiva e resposta a riscos ambientais em tempo real.

Esta análise comparativa revelou que, apesar da maturidade e robustez destas plataformas, existe uma lacuna em soluções que ofereçam maior flexibilidade e personalização de acordo com as necessidades específicas de cada organização. Assim, os resultados desta comparação serviram como base para orientar o desenvolvimento do protótipo proposto, que visa colmatar essas limitações através de uma abordagem centrada na modularidade, usabilidade e adaptação à realidade operacional de diferentes empresas.

### 2.3.3 Bibliotecas e Ferramentas de Desenvolvimento

#### React.js

O *React.js* é uma das bibliotecas<sup>20</sup> *JavaScript* de *frontend* mais populares (Schwarzmueller 2022). O seu uso é focado na criação de interfaces de utilizador a partir de excertos de código individuais chamados componentes e das suas combinações em telas inteiras, páginas e aplicativos (React 2025).

#### Angular

O *Angular* é uma *framework* de desenvolvimento *frontend* em *TypeScript*, mantida pela *Google*. Ao contrário do *React* e *Vue*, que são mais flexíveis, o *Angular* proporciona uma estrutura completa e fortemente opinativa, seguindo o padrão *MVC* (*Model-View-Controller*) e tem funcionalidades integradas como injeção de dependências, roteamento e comunicação com serviços externos (Angular 2025).

#### Vue.js

O *Vue.js* é uma *framework* progressiva do *JavaScript* direcionada à construção das interfaces de utilizador. Desenvolvido por Evan You, a sua arquitetura permite uma adoção gradual, uma vez que pode ser utilizada tanto em componentes simples quanto em aplicações inteiras Single Page Application (SPA) (Vue 2025). O *Vue* destaca-se pela curva de aprendizagem que é acessível, integrabilidade simplificada com projetos existentes e uma clara demarcação entre a lógica e a apresentação visual, através da utilização dos arquivos *.vue* (Mastery 2025).

O facto de ser suportado por uma comunidade ativa e um vasto ecossistema, incluindo ferramentas como o *Vue Router* e o *Pinia*<sup>21</sup>, tornam-no uma alternativa sólida ao *React* e o *Angular*, particularmente em cenários que envolvem prototipagem rápida e em aplicações moderadamente complexas.

#### Vuexy

*Vuexy* é um *template* de *User Interface (UI)* concebido para o desenvolvimento de aplicações *web* rápidas e escaláveis, compatível com diversas *frameworks* como *HyperText Transfer Protocol (HTTP)*, *Next.js*, *.NET Core* e *Figma* (Pixelnvent 2025). A plataforma oferece duas versões: uma mais simples, adequada para iniciantes, e outra mais avançada, que inclui *routing*, páginas modelo e integração com bibliotecas como *ApexCharts* e *i18next* (suporte multilingue). Além disso, fornece uma base de dados falsa e estrutura preparada para ligação com uma base de dados dinâmica e funcional (Vuexy 2025).

### Comparação entre Bibliotecas e Frameworks de Frontend

Embora a empresa tenha solicitado explicitamente o uso de *React.js* em conjunto com o *template* *Vuexy*, é relevante compreender o panorama tecnológico atual e comparar com outras soluções populares de desenvolvimento *frontend*. Esta comparação permite não só

<sup>20</sup>No contexto do *React.js*, uma biblioteca é um conjunto de funcionalidades que se pode utilizar para obter resultados que normalmente exigiriam mais código e trabalho da parte do utilizador.

<sup>21</sup>Site oficial do *Pinia*: <https://pinia.vuejs.org/>

justificar a escolha adotada, como também evidenciar as vantagens e limitações face a alternativas amplamente utilizadas, como o *Vue.js* e o *Angular*.

A Tabela 2.3 sintetiza os principais aspectos descritos nas secções anteriores relativamente a estas três tecnologias.

<b>Critério</b>	<b>React.js</b>	<b>Vue.js</b>	<b>Angular</b>
<b>Tipo</b>	Biblioteca UI	Framework progressiva	Framework completa
<b>Curva de Aprendizagem</b>	Moderada	Baixa a moderada	Alta
<b>Arquitetura</b>	Flexível (liberdade na estruturação)	Estrutura recomendada, mas flexível	Estrutura rigidamente definida (MVC)
<b>Popularidade e Comunidade</b>	Muito alta, grande ecossistema	Alta, especialmente em projetos individuais/ <i>startups</i>	Alta, especialmente em ambientes corporativos
<b>Integração com Bibliotecas Externas</b>	Muito fácil e comum	Fácil, com ecossistema próprio ou externo	Mais rígida, mas com muitas funcionalidades nativas
<b>Desempenho</b>	Elevado (virtual DOM)	Elevado (virtual DOM otimizado)	Bom, mas com maior peso inicial
<b>Casos de Uso Típicos</b>	SPAs, dashboards, aplicações escaláveis	Projetos rápidos, UIs interativas, apps pequenas a médias	Grandes sistemas empresariais, aplicações complexas

Tabela 2.3: Comparação entre bibliotecas e *frameworks* de *frontend* (autoria própria)

A análise comparativa entre *React.js*, *Vue.js* e *Angular* permite contextualizar a escolha da stack *frontend* adotada no projeto, destacando não só as capacidades da biblioteca *React.js*, como também as características que a diferenciam de outras soluções populares.

*React.js* destaca-se pela sua elevada popularidade, desempenho eficiente com virtual DOM, e flexibilidade arquitetural, o que o torna especialmente indicado para o desenvolvimento de interfaces ricas, *dashboards* interativos e aplicações escaláveis, como é o caso da plataforma ESG proposta. A integração com bibliotecas externas e a vasta comunidade de suporte reforçam ainda mais essa escolha.

Embora *Vue.js* apresente uma curva de aprendizagem mais baixa e seja ideal para protótipos rápidos e UIs leves, e *Angular* ofereça uma estrutura completa adequada a projetos de larga escala corporativa, *React.js* equilibra bem flexibilidade, desempenho e escalabilidade. Assim, a escolha desta tecnologia, em conformidade com os requisitos da empresa, mostra-se adequada ao contexto e aos objetivos do projeto.



# Capítulo 3

## Análise do Problema

Neste próximo Capítulo aborda-se a análise do problema em questão. Inicia-se com a descrição do domínio do problema através do glossário de termos usados, do modelo de domínio e do mapa de materialidade da Devscope de acordo com as métricas indicadas pela SASB para o setor de *software* e serviços IT. Por fim, serão elencados os requisitos funcionais e não funcionais do sistema.

### 3.1 Contexto

A fase de análise é uma das mais importantes no ciclo de vida do desenvolvimento de *software*, pois é nela que ocorre a recolha dos requisitos, funcionalidades e necessidades do cliente, bem como das especificações do sistema. O objetivo final é definir de forma clara os recursos e funcionalidades que compõem a solução a ser desenvolvida.

Qualquer informação incorreta, ou não obtida, pode comprometer o desenvolvimento do produto, tanto em termos de qualidade como de cumprimento dos prazos.

A proposta de desenvolvimento de uma plataforma ESG surgiu da necessidade identificada pela Devscope em centralizar e otimizar o controlo de diversas métricas relacionadas com os pilares ambiental, social e de governança (ESG). Esta necessidade prende-se com a crescente importância de monitorizar e reportar práticas sustentáveis, promovendo uma gestão mais transparente, eficiente e alinhada com os objetivos de responsabilidade corporativa da empresa. Ao centralizar estes dados numa única plataforma, a Devscope pretende não só melhorar a sua capacidade de análise e tomada de decisões, como também reforçar o seu compromisso com a sustentabilidade e o impacto positivo na sociedade.

### 3.2 Domínio do problema

A presente secção visa descrever o domínio do problema através de vários artefactos de complexidades diversas, nomeadamente conceitos e diagramas.

A Tabela 3.1 apresenta os conceitos de domínio do problema e que serão usados ao longo do desenvolvimento.

Conceito (EN)	Descrição
<b>ESG</b>	<b>ESG (Ambiente, Social e Governança):</b> Conjunto de critérios que avaliam o desempenho ambiental, social e de governança de uma organização, com foco na sustentabilidade e responsabilidade corporativa.
<b>Dashboard</b>	<b>Painel :</b> Resumo gráfico de várias informações importantes, normalmente utilizado para dar uma visão geral de um negócio.
<b>Materiality Map</b>	<b>Mapa de Materialidade :</b> Ferramenta que destaca as métricas ESG mais relevantes para cada setor, segundo as normas da SASB.
<b>Metric</b>	<b>Métrica:</b> Unidade de medida usada para quantificar aspectos específicos do desempenho ESG, como emissões de carbono, diversidade na força de trabalho ou políticas anticorrupção.
<b>Dataset</b>	<b>Conjunto de Dados :</b> Coleção estruturada de dados ESG, normalmente em formato tabular, que contém informações como categorias, datas e valores associados.
<b>URL</b>	<b>Localizador Uniforme de Recursos (URL):</b> Endereço que identifica e localiza um recurso na internet, como páginas web, APIs ou arquivos.
<b>API</b>	<b>Interface de Programação de Aplicações (API):</b> Conjunto de regras que permite a comunicação entre diferentes sistemas ou componentes de software. Utilizada para integrar funcionalidades ou aceder a dados.
<b>KPI</b>	<b>Indicador-Chave de Desempenho (KPI):</b> Métrica usada para avaliar o sucesso de uma organização ou de uma atividade específica na conquista de objetivos estratégicos ou operacionais.
<b>SPA</b>	<b>Aplicação de Página Única (SPA):</b> Tipo de aplicação web que carrega uma única página HTML e atualiza dinamicamente o conteúdo conforme o utilizador interage, sem recarregar completamente a página.

Tabela 3.1: Glossário do domínio do problema (autoria própria)

A Figura 3.1 ilustra uma visualização do modelo de domínio, traçando as relações entre os principais conceitos da solução pretendida. Cada uma das entidades principais foi especificada como uma raiz agregada, uma escolha motivada por preocupações de consistência transnacional, concorrência, escalabilidade e clareza semântica.

Um bom exemplo é a distinção entre **Empresa** e **Setor**. Enquanto o **Setor** caracteriza a empresa no contexto da indústria, também organiza o sistema agrupando as **Subáreas** e, por extensão, as **Métricas** relevantes para a avaliação ESG com base nas normas SASB. Ao representá-lo como um agregado autônomo, esta abordagem localiza a lógica subjacente, minimizando as redundâncias e promovendo a consistência.

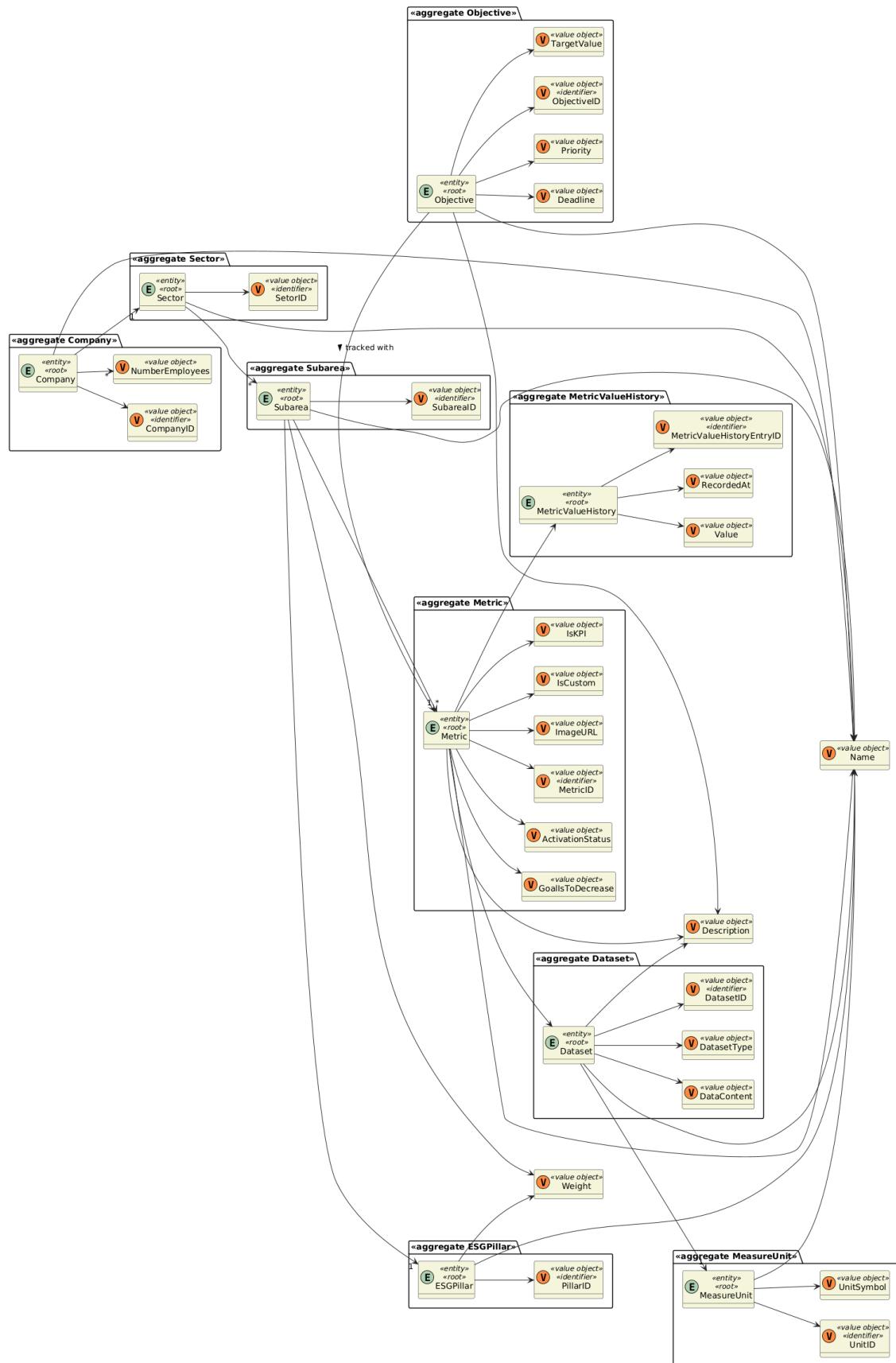


Figura 3.1: Modelo de Domínio (autoria própria)

### 3.2.1 Visão Geral da Aplicação

O *dashboard* apresenta a pontuação ESG consolidada da empresa, oferecendo uma visão geral do seu desempenho em sustentabilidade. Além disso, destaca as métricas mais relevantes, assim como aquelas que apresentaram melhorias ou quedas em relação ao último registro.

Outra seção da plataforma disponibiliza uma visão detalhada de todas as métricas existentes: métricas customizáveis (criadas pelos próprios utilizadores) e métricas orientadas pelos padrões definidos pela SASB.

Cada métrica possui atributos específicos, incluindo o pilar e subárea ESG a que pertence, o seu código identificador, os conjunto de dados que lhe estão associados, a unidade de medida associada aos mesmos e a classificação do seu progresso.

De acordo com a SASB, adotado pela Devscope neste projeto, existem normas específicas para cada setor, como é o caso do setor de Software e Serviços de TI, que determinam quais as métricas devem ser reportadas (SASB 2025b). É com base nestas diretrizes que se constrói o Mapa de Materialidade, indicado na Figura 3.2.

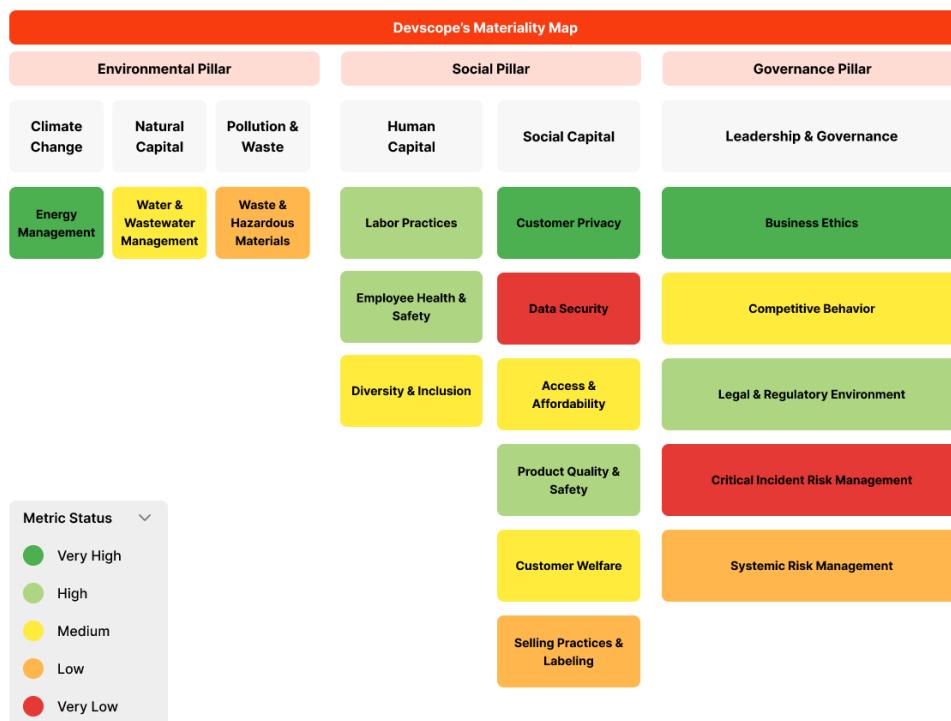


Figura 3.2: Mapa de Materialidade segundo o setor SASB da Devscope (Software e serviços IT) (autoria própria)

Neste mapa, um dos elementos centrais da plataforma, reflete as métricas ativas não só definidas pela SASB, consoante o setor da empresa, bem como as métricas personalizadas. Este mapa organiza as métricas segundo os pilares ESG e respectivas subáreas, e atribui uma cor correspondente ao seu progresso, proporcionando uma visão estruturada das prioridades de sustentabilidade da organização.

Por fim, outra página da plataforma é dedicada à gestão de objetivos, permitindo acompanhar os objetivos definidos, as métricas a eles associadas, o progresso na sua concretização ao saber se se encontram dentro ou fora do plano estabelecido.

### 3.3 Requisitos Funcionais e Não Funcionais

Esta secção compila os requisitos obtidos com o cliente e categoriza-os entre funcionais e não funcionais.

#### 3.3.1 Requisitos Funcionais

Os requisitos funcionais correspondem a funcionalidades presentes no *software* a ser desenvolvido, normalmente representadas por casos de uso e acompanhados por diagramas. Os requisitos funcionais deste projeto estão organizados na Tabela 3.2, onde cada requisito é considerado um caso de uso e identificado por um código alfa-numérico.

Código	Descrição Curta	Descrição
<b>UC-01</b>	Visualização de KPI's	Como utilizador da plataforma, quero visualizar rapidamente os principais KPIs ambientais, sociais e de governação, para poder avaliar o desempenho da organização.
<b>UC-02</b>	Filtragem de Métricas	Como utilizador da plataforma, quero poder filtrar os dados por pilar, de modo a facilitar a navegação.
<b>UC-03</b>	Mapa de Materialidade	Como utilizador da plataforma, quero visualizar a matriz de materialidade dividida por categorias ESG com um sistema de cores intuitivo, para identificar rapidamente os temas mais críticos e, ao clicar em cada indicador, aceder a detalhes explicativos que me ajudem a compreender o seu progresso e evolução.
<b>UC-04</b>	Pontuação ESG	Como utilizador da plataforma, quero aceder a uma pontuação ESG agregada, para ter uma visão geral do desempenho da empresa em responsabilidade social, ambiental e de governança.
<b>UC-05</b>	Detalhes de Métricas	Como utilizador da plataforma, quero clicar numa métrica e ver mais detalhes, para compreender a progressão de cada indicador.
<b>UC-06</b>	Métricas Customizáveis	Como utilizador da plataforma, quero poder criar métricas ESG específicas à realidade da empresa e definir a sua fonte de dados, para garantir que o painel reflete os indicadores mais relevantes para a nossa estratégia.
<b>UC-07</b>	Importação de Dados	Como utilizador da plataforma, quero poder importar dados para associar às métricas.

Tabela 3.2: Lista de Casos de Uso (autoria própria)

A Figura 3.3 é um diagrama que compila os casos de usos principais do projeto, que serão apresentados de forma detalhada.

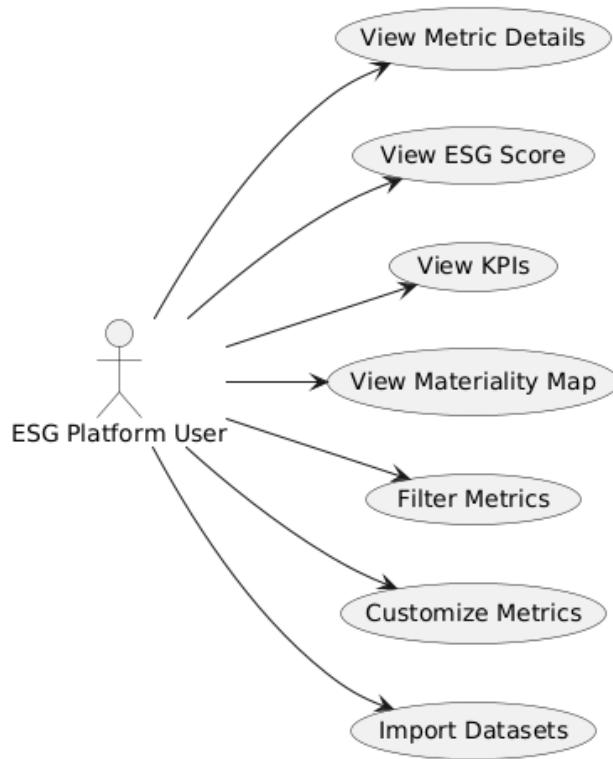


Figura 3.3: Vista de Cenários (autoria própria)

### UC-01 | Visualização de KPI's

**Objetivo:** Permitir ao utilizador consultar rapidamente os principais *Key Performance Indicator (KPI)* ESG da organização.

**Autor Principal:** Utilizador da plataforma

**Descrição Geral:** Este caso de uso disponibiliza uma visão geral de métricas dos três pilares (Ambiental, Social e de Governação). A interface mostra indicadores como emissões de CO<sub>2</sub>, diversidade na força de trabalho, entre outros.

**Fluxo Principal:**

1. O utilizador acede ao painel principal da plataforma.
2. A aplicação apresenta os KPIs.
3. O utilizador pode visualizar valores atuais e tendências.

**Resultado Esperado:** O utilizador obtém uma percepção imediata do desempenho ESG da empresa.

### UC-02 | Filtragem de Métricas

**Objetivo:** Permitir ao utilizador filtrar os dados por pilar ESG (*Environmental, Social ou Governance*).

**Autor Principal:** Utilizador da plataforma

**Descrição Geral:** Este caso de uso oferece um mecanismo de filtragem que facilita a

navegação por métricas, de acordo com o pilar de interesse.

**Fluxo Principal:**

1. O utilizador seleciona o pilar desejado através de controlos de filtragem.
2. A plataforma atualiza os dados apresentados para refletir apenas as métricas relevantes.

**Resultado Esperado:** O utilizador consegue concentrar-se nos dados ESG mais relevantes ao seu objetivo de análise.

### **UC-03 | Mapa de Materialidade**

**Objetivo:** Exibir uma matriz de materialidade com base no setor da empresa, permitindo a priorização de temas ESG.

**Ator Principal:** Utilizador da plataforma

**Descrição Geral:** A matriz destaca os temas ESG mais críticos segundo o modelo SASB, com categorização por pilar e subárea, e códigos de cor para facilitar a interpretação.

**Fluxo Principal:**

1. O utilizador acede à secção de materialidade.
2. É apresentada a matriz correspondente ao setor da empresa.
3. O utilizador pode clicar em indicadores para visualizar descrições detalhadas.

**Resultado Esperado:** Facilita a identificação de prioridades estratégicas ESG para o setor em análise.

### **UC-04 | Pontuação ESG**

**Objetivo:** Fornecer uma pontuação ESG agregada para a empresa.

**Ator Principal:** Utilizador da plataforma

**Descrição Geral:** A aplicação calcula e exibe uma pontuação ESG com base nas métricas existentes e ativas, representada por cores e uma escala numérica.

**Fluxo Principal:**

1. O utilizador acede ao *dashboard* principal.
2. A plataforma mostra a pontuação global da empresa.
3. Podem existir dicas ou justificações para a pontuação.

**Resultado Esperado:** Oferece uma visão sintética do desempenho ESG.

### **UC-05 | Detalhes de Métricas**

**Objetivo:** Permitir a exploração detalhada de cada métrica ESG.

**Ator Principal:** Utilizador da plataforma

**Descrição Geral:** O utilizador pode selecionar uma métrica para visualizar dados históricos, a classificação do seu progresso, o conjunto de dados de origem e o pilar e subárea a que pertence.

**Fluxo Principal:**

1. O utilizador clica numa métrica apresentada em *dashboards* ou gráficos.

2. A aplicação apresenta uma vista detalhada com gráficos temporais, dados e metadados.

**Resultado Esperado:** Maior compreensão do desempenho de cada indicador específico.

### UC-06 | Métricas Customizáveis

**Objetivo:** Permitir ao utilizador definir novas métricas ESG alinhadas com a realidade da empresa.

**Autor Principal:** Utilizador da plataforma

**Descrição Geral:** Funcionalidade que permite criar métricas personalizadas, associando-as a fontes de dados e subáreas ESG específicas.

**Fluxo Principal:**

1. O utilizador acede à área de configuração de métricas.
2. Define os dados pedidos.
3. A métrica é validada e integrada nos *dashboards*.

**Resultado Esperado:** Flexibilidade para adaptar o sistema ESG a diferentes contextos empresariais.

### UC-07 | Importação de Dados

**Objetivo:** Permitir ao utilizador importar conjuntos de dados em formato CSV para análise ESG.

**Autor Principal:** Utilizador da plataforma

**Descrição Geral:** O utilizador pode carregar ficheiros de dados externos, que serão associados a métricas e utilizados em gráficos e cálculos.

**Fluxo Principal:**

1. O utilizador escolhe o ficheiro CSV a importar.
2. O sistema valida o conteúdo.
3. As informações passam a estar disponíveis para visualização e análise.

**Resultado Esperado:** Capacidade de integração de dados externos no ecossistema ESG da aplicação.

### 3.3.2 Requisitos Não Funcionais

Requisitos não funcionais correspondem a todas as propriedades do sistema que não se referem diretamente a funcionalidades específicas, mas sim a qualidades e restrições sobre como o sistema se deve comportar. Incluem aspetos como desempenho, usabilidade, confiabilidade, manutenibilidade e outros fatores de qualidade da aplicação.

Uma forma amplamente utilizada de categorizar estes requisitos é através do modelo **FURPS+**, detalhado na Tabela 3.3, que cataloga os requisitos não funcionais em cinco categorias principais, com um grupo adicional representado pelo símbolo +.

<b>Letra</b>	<b>Categoria</b>	<b>Descrição</b>
<b>F</b>	<b>Functionality</b>	Funcionalidades esperadas, segurança, interoperabilidade e adequação funcional
<b>U</b>	<b>Usability</b>	Facilidade de uso, aparência, acessibilidade, e interação com o utilizador
<b>R</b>	<b>Reliability</b>	Confiabilidade, tolerância a falhas, disponibilidade, e recuperação
<b>P</b>	<b>Performance</b>	Tempo de resposta, eficiência de recursos e escalabilidade
<b>S</b>	<b>Supportability</b>	Facilidade de manutenção, extensibilidade, modularidade, e testabilidade
<b>+</b>	<b>Others</b>	Restrições de design, implementação, normas, tecnologias ou frameworks

Tabela 3.3: Categorias FURPS+ (autoria própria)

Os tópicos seguintes categorizam os requisitos não funcionais obtidos consoante o modelo FURPS+.

### 3.3.3 Usabilidade (U)

- *Feedback* imediato ao utilizador através de alertas indicando sucesso ou erro das ações realizadas.
- Interface do utilizador em inglês para facilitar o acesso global.

### 3.3.4 Confiabilidade (R)

- O sistema deve apresentar mensagens de erro claras e específicas, indicando a origem do problema, em pelo menos 95% dos casos de falha.

### 3.3.5 Performance (P)

- As interações principais devem ser eficientes e não exceder 30 segundos por operação.
- O tempo total para utilizar a prova de conceito (incluindo *seeding* de dados e carregamento da interface) não deve ultrapassar 3 minutos.

### 3.3.6 Suportabilidade (S)

- Conformidade com o padrão SASB para estruturação das métricas ESG.
- A arquitetura do sistema deve permitir a adição de novas funcionalidades com impacto mínimo na base de código existente e sem degradação da usabilidade.

### 3.3.7 + (Outros)

- Seguir boas práticas de programação e arquitetura coesa.
- Utilizar fonte de dados simulada através de scripts de seeding.
- Comunicação entre os diferentes módulos da aplicação através de REST APIs utilizando HTTP/S para garantir interoperabilidade e segurança.

- Devem ser implementados testes unitários e de integração cobrindo, no mínimo, 80% das funcionalidades críticas do sistema.
- Código modular e bem estruturado, seguindo princípios de separação de responsabilidades.
- Arquitetura baseada em boas práticas, como *Clean Architecture* e uso de camadas (*Controller*, *Service*, *Repository*).

# Capítulo 4

## Desenho da solução

Este capítulo aborda a fase de *design* no processo de desenvolvimento de *software*, onde, após a recolha de requisitos e análise das funcionalidades, são delineados os elementos do sistema através de diagramas que refletem a arquitetura e os modelos escolhidos, tendo em conta os requisitos e restrições definidos. Conforme descrito em Tsui, Karam e Bernal 2022, o *design* é dividido em duas etapas principais:

- **Design arquitetural** — uma visão macro e de alto Nível do sistema, onde são identificados os principais componentes, as suas propriedades externas e as relações entre eles. Esta fase é orientada pelos requisitos funcionais e não funcionais, bem como por considerações técnicas;
- **Design detalhado** — uma decomposição mais pormenorizada dos componentes, que detalha como cada módulo cumpre os requisitos funcionais definidos. Esta fase representa uma visão micro do sistema, guiada pela arquitetura estabelecida e pelos requisitos funcionais.

### 4.1 Arquitetura do Sistema

As seguintes subsecções elaboram os diversos tipos de modelos usados para representar a arquitetura da solução desenvolvida, acompanhados com diagramas referentes à aplicação dos mesmos.

#### 4.1.1 Clean Architecture

A *Clean Architecture* é um modelo arquitetural de *software* que privilegia a manutenibilidade e a escalabilidade do sistema, ao restringir as dependências entre os componentes de diferentes camadas. Neste modelo, diz-se que um componente A depende de um componente B se A conhece diretamente o nome de B, por exemplo, se o importa ou referencia. O objetivo da *Clean Architecture* é limitar estas dependências para que alterações em uma parte do sistema não causem efeitos colaterais noutras. Para isso, aplica-se o princípio da inversão de dependências: o componente A define uma interface, e o componente B implementa essa interface. Mesmo que A chame B em tempo de execução, B é quem depende de A, pois precisa de conhecer e implementar a interface definida por A (Lano e Tehrani 2023).

Segundo Lano e Tehrani 2023, este princípio está na base da chamada regra de dependência, que afirma que componentes dependentes de plataformas externas (como bases de dados, frameworks ou interfaces gráficas) podem depender da lógica de negócio, mas nunca o contrário. Assim, a lógica central do sistema permanece isolada e protegida de alterações

tecnológicas periféricas, garantindo maior estabilidade e facilidade de evolução ao longo do tempo.

A Figura 4.1 ilustra o modelo *Clean Architecture* e as suas camadas concêntricas.

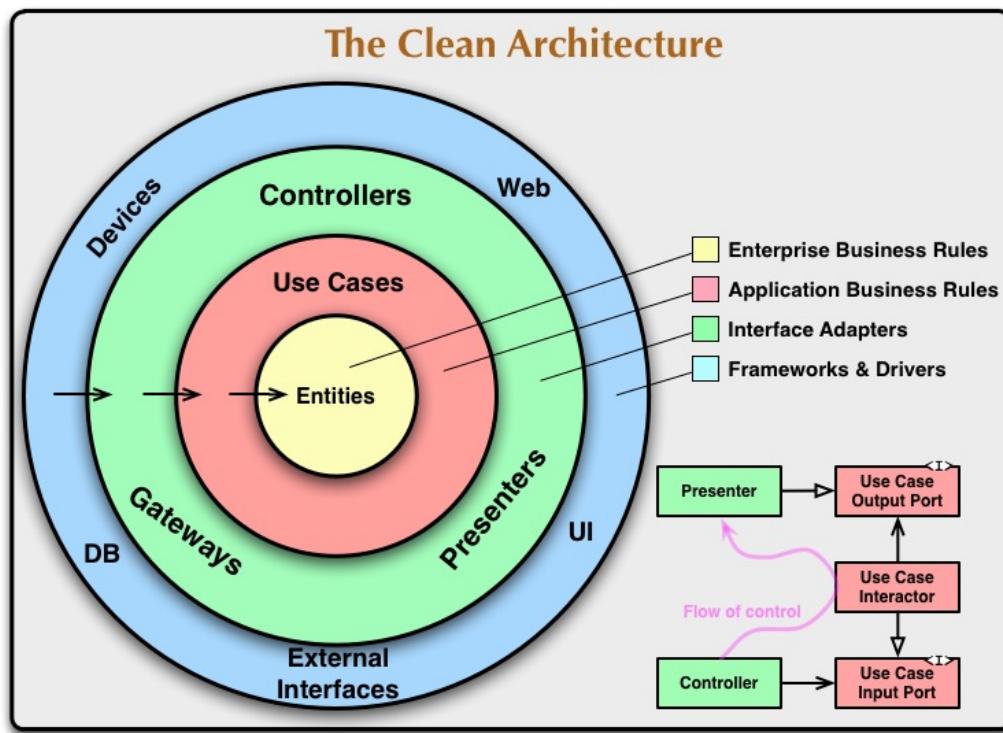


Figura 4.1: Modelo Arquitetural *Clean Architecture* (Martin 2012)

Embora o projeto se inspire nos princípios da *Clean Architecture*, nomeadamente a separação clara entre as camadas de apresentação, lógica de negócio e acesso a dados, não foi realizada uma implementação rigorosa deste modelo (por exemplo, com injeção de dependências ou uso sistemático de interfaces). Esta decisão deve-se ao âmbito do projeto e à sua natureza prática, que não exigem este Nível de abstração.

Em alternativa à implementação deste modelo, foram aplicados os modelos C4 (Figura 4.1.2) e “4+1” (Figura 4.1.3) para representar graficamente a arquitetura da solução, proporcionando uma visão clara e compreensível da estrutura do sistema e das suas interações.

#### 4.1.2 Modelo C4

O modelo C4 descreve a arquitetura de um *software* em quatro níveis de detalhe (*Context*, *containers*, *components* e *code*). De modo a criar os diferentes diagramas é necessário diferentes graus de abstração (Brown e Betts 2018):

- **(Nível 1) System context diagram** - Dá uma vista geral do contexto em que o sistema se insere, isto é, quem usa o *software* criado e como é que outros sistemas inseridos no mesmo contexto comunicam com ele.

- **(Nível 2) Container diagram** - Detalha o sistema a um nível macro, decompondo-o em "containers", tais como aplicações (monolíticas ou microserviços), armazenamentos de dados, entre outros.
- **(Nível 3) Component diagram** - Decompõe um *container* em elementos que constituem abstrações reais no código a ser produzido.
- **(Nível 4) Code** - Nível de detalhe final, define a implementação de um componente.

A Figura 4.2 hierarquiza a granularidade pretendida no detalhe das diferentes visões do sistema numa representação com o modelo C4.

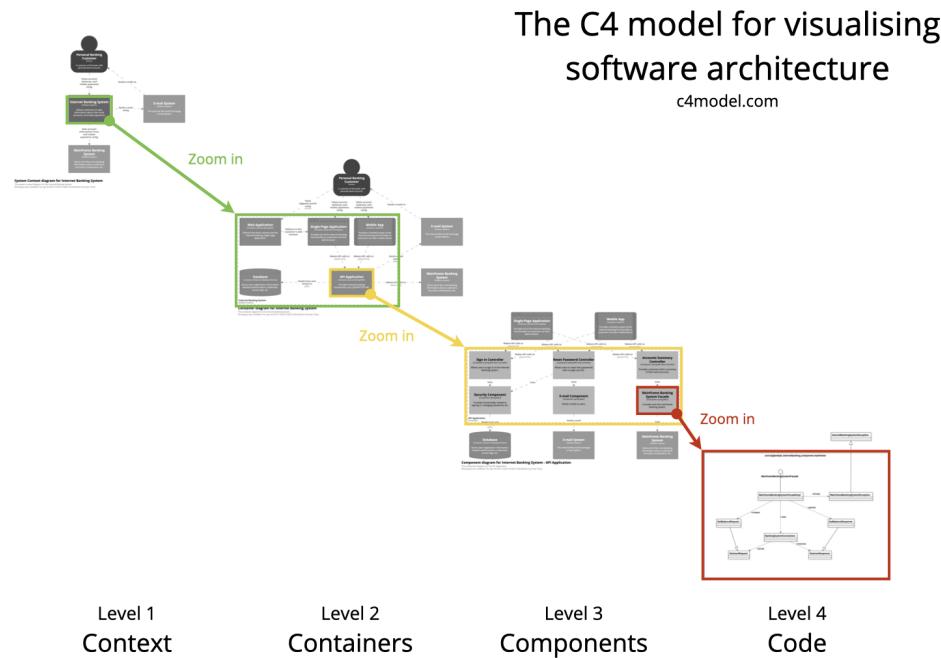


Figura 4.2: Modelo Arquitetural C4 (Brown 2025)

A granularidade do Nível 4 revela-se excessiva e retira o foco aos diagramas que melhor ilustram a arquitetura do sistema, portanto, não será abordado neste relatório.

### 4.1.3 Modelo de Vistas 4+1

Segundo Kruchten 1995, o modelo arquitetural "4+1" descreve a arquitetura de um sistema de *software* a partir de cinco perspectivas complementares, cada uma focada em diferentes preocupações e públicos-alvo:

- **Vista Lógica** - Foca-se na funcionalidade do sistema, pois representa a estrutura dos principais componentes do domínio.
- **Vista de Processos** - Descreve os aspectos dinâmicos do sistema, como a comunicação entre processos concorrentes e a forma como as tarefas são sincronizadas e distribuídas.
- **Vista Física** - Mapeia os componentes de *software* para os recursos de *hardware* e ilustra como o sistema está distribuído.

- **Vista de Implementação/Desenvolvimento** - Mostra a estrutura estática do código, organização em módulos/pacotes e o ambiente de desenvolvimento.
- **Vista de Cenários** - Representa os principais casos de uso ou fluxos de interação do sistema.

A Figura 4.3 ilustra as diferentes vistas, o respetivo público-alvo para cada uma, e as suas interações.

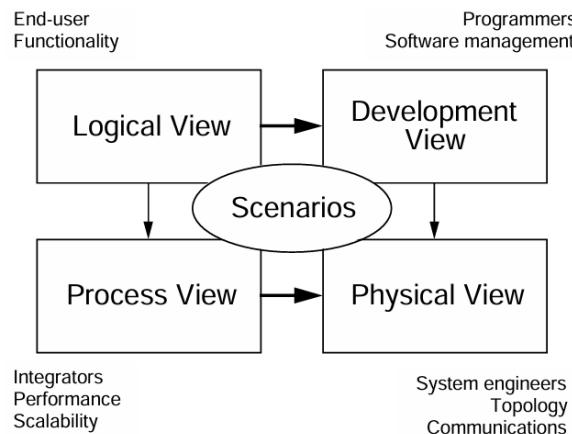


Figura 4.3: Modelo de Vistas "4+1"(Kruchten 1995)

#### 4.1.4 Vista de Processos

Alguns casos de uso partilham entre si a mesma estrutura de interação representada nos diagramas da vista de processos. Exemplos disso são os casos de uso UC-01, UC-02, UC-03, UC-04 e UC-05, que seguem a sequência ilustrada na Figura 4.4, variando apenas no conceito de negócio específico que cada um aborda.

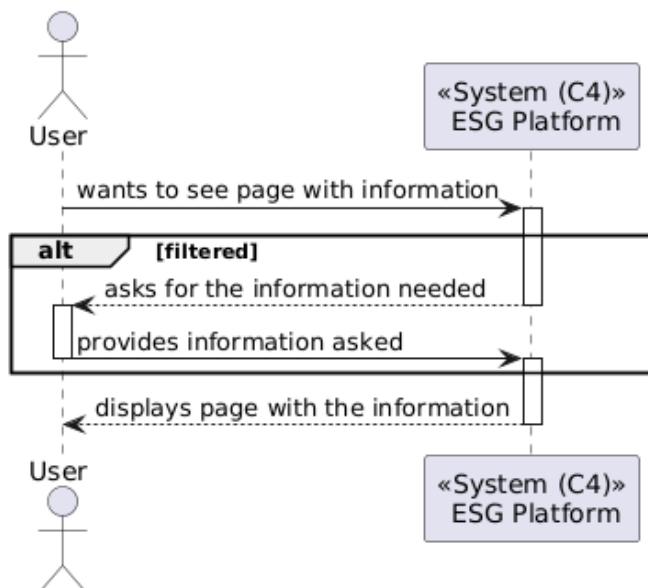


Figura 4.4: Vista de processos dos casos de uso de consulta/visualização e filtragem (Nível 1) (autoria própria)

De igual forma, os respetivos diagramas de Nível 2 também partilham uma estrutura comum, diferenciando-se apenas na lógica de negócio subjacente. Esta semelhança é evidenciada na Figura 4.5, que exemplifica uma dessas variações mantendo a arquitetura processual de base.

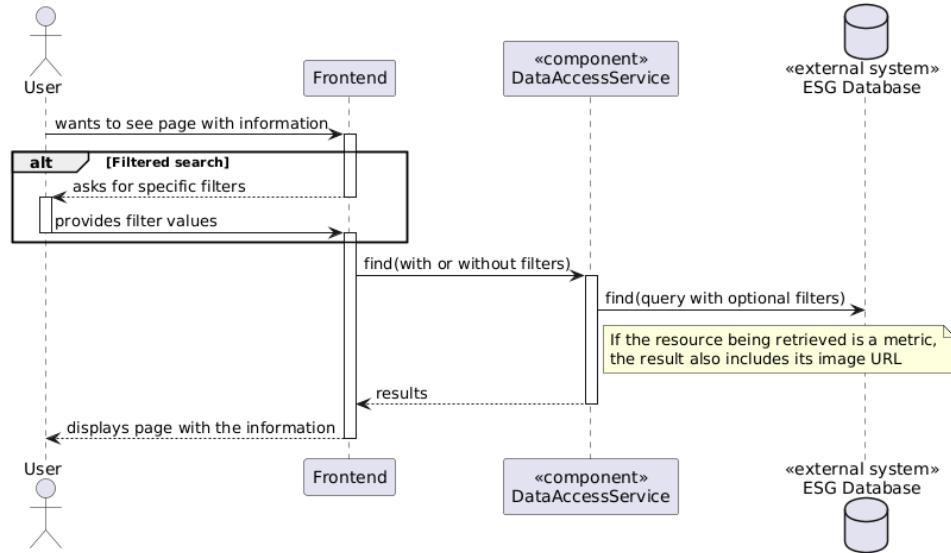


Figura 4.5: Vista de processos dos casos de uso de consulta/visualização e filtragem (Nível 2) (autoria própria)

É no Nível 3 que se tornam evidentes as diferenças nos fluxos de interação, consoante os requisitos de cada funcionalidade. Como ilustrado na Figura 4.6, os casos de uso UC-01, UC-03 e UC-04 envolvem apenas o acesso à base de dados.

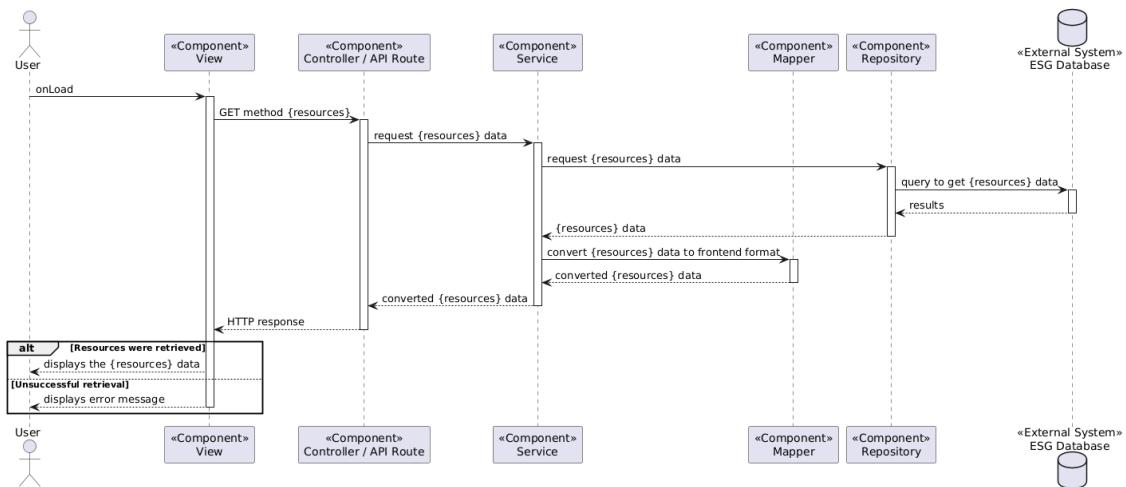


Figura 4.6: Vista de processos dos casos de uso de consulta (Nível 3) (autoria própria)

Já o caso UC-05 (Figura 4.7) inclui também comunicação com o serviço de armazenamento na nuvem.

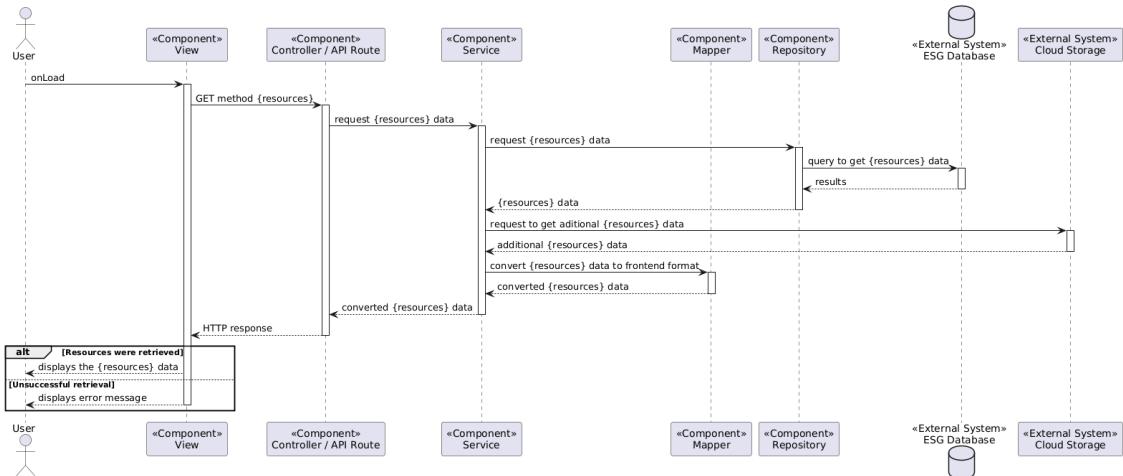


Figura 4.7: Vista de processo do caso de uso de consulta com acesso ao armazenamento na nuvem (Nível 3) (autoria própria)

A UC-02 (Figura 4.8) distingue-se ainda por incluir uma etapa de filtragem adicional antes da obtenção dos dados.

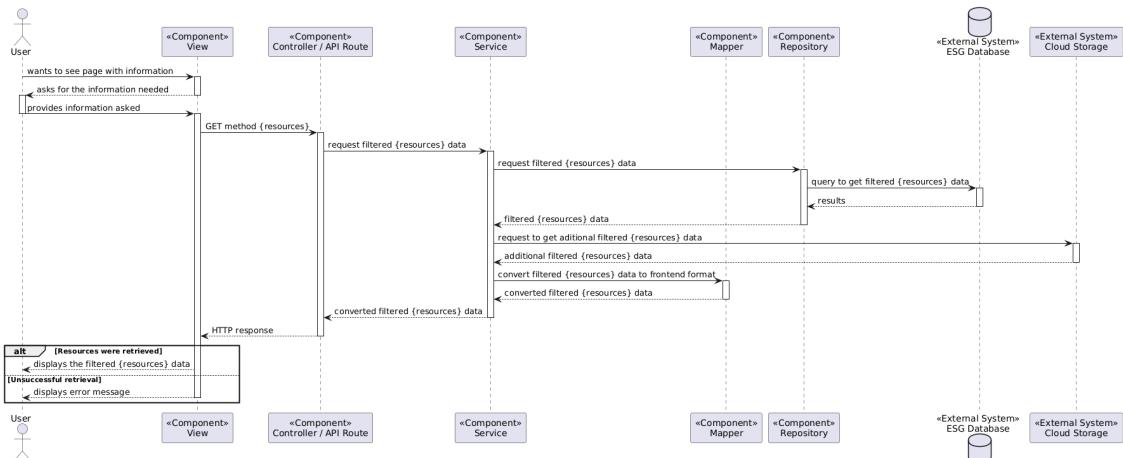


Figura 4.8: Vista de processo do caso de uso de filtragem (Nível 3) (autoria própria)

Da mesma forma, no Nível 1, ambas os casos de uso UC-06 e UC-07 são bastante semelhantes, mudando apenas a lógica do negócio, como é ilustrado na Figura 4.9.

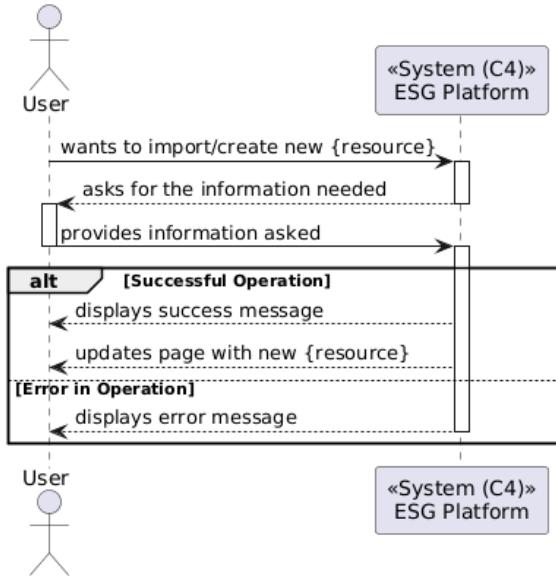


Figura 4.9: Vista de processo dos casos de uso de criação e importação (Nível 1) (autoria própria)

A mesma tendência continua no nível seguinte, como é representado pela Figura 4.10.

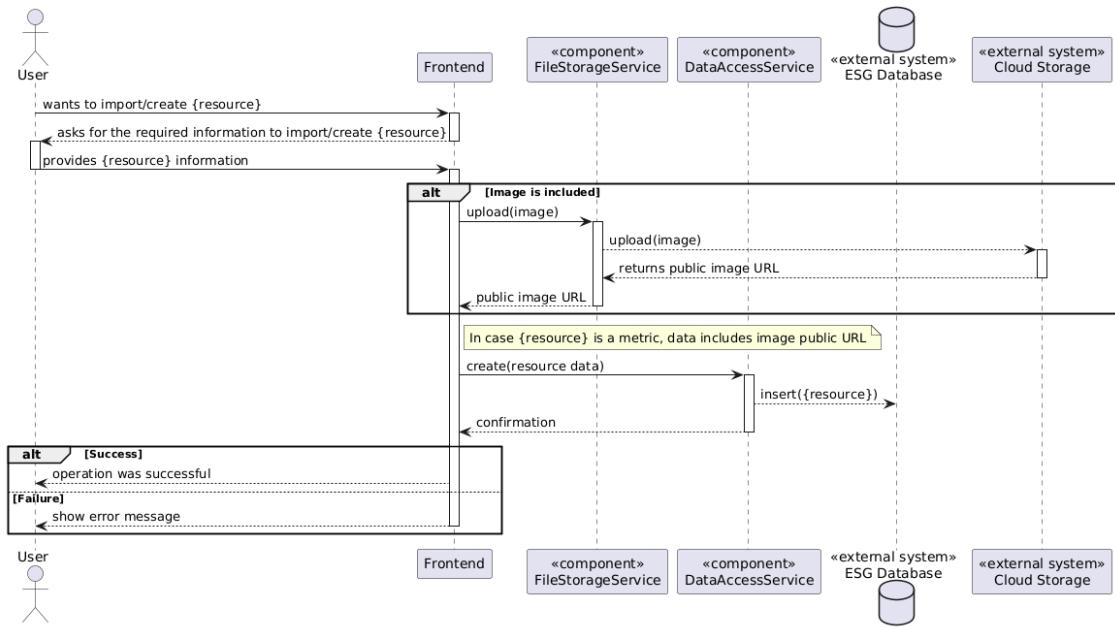


Figura 4.10: Vista de processo dos casos de uso de criação e importação (Nível 2) (autoria própria)

A diferença no caso de uso UC-06 é realçada pelo diagrama de granularidade de Nível 3, como se pode ver na Figura 4.11.

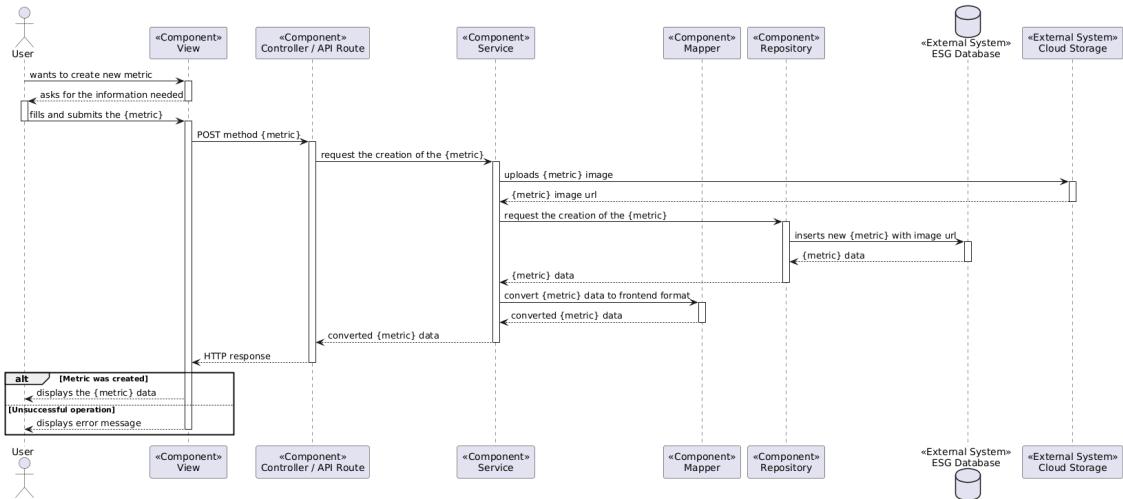


Figura 4.11: Vista de processo do caso de uso de criação de métricas (Nível 3) (autoria própria)

De forma semelhante, o caso de uso UC-07 também apresenta essa distinção, evidenciada na Figura 4.12.

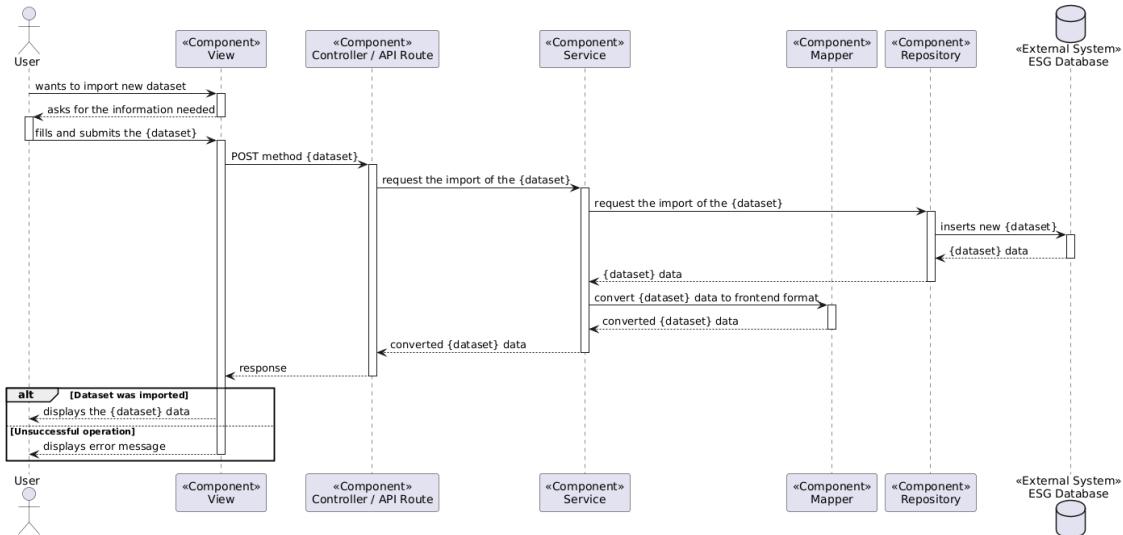


Figura 4.12: Vista de processo dos casos de uso de importação de dados (Nível 3) (autoria própria)

#### 4.1.5 Vista de Cenários

A vista de cenários, ilustrada pela Figura 3.3 e apresentada na secção 3.3.1, compila os caso de usos principais. Esta vista é redundante, daí o "+1" na designação do modelo arquitetural (Kruchten 1995).

#### 4.1.6 Vista Lógica

Esta vista será abordada por vários diagramas correspondentes a sucessivos níveis de detalhe na representação lógica do sistema, iniciando-se pelo Nível 1, com a Figura 4.13, que permite

entender o que o sistema disponibiliza, assim como que APIs consome.

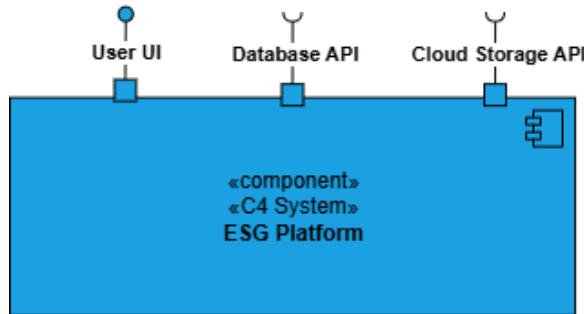


Figura 4.13: Vista Lógica (Nível 1) (autoria própria)

A Figura 4.14 apresenta o Nível 2, oferecendo uma visão mais detalhada do sistema. Este nível inclui o contentor *Frontend*, responsável por disponibilizar a *User UI*, através da qual os utilizadores interagem com a aplicação. Este contentor consome duas APIs, a *Database API* e a *Cloud Storage API*, que, respetivamente, comunicam com a base de dados e com o serviço de armazenamento na nuvem.

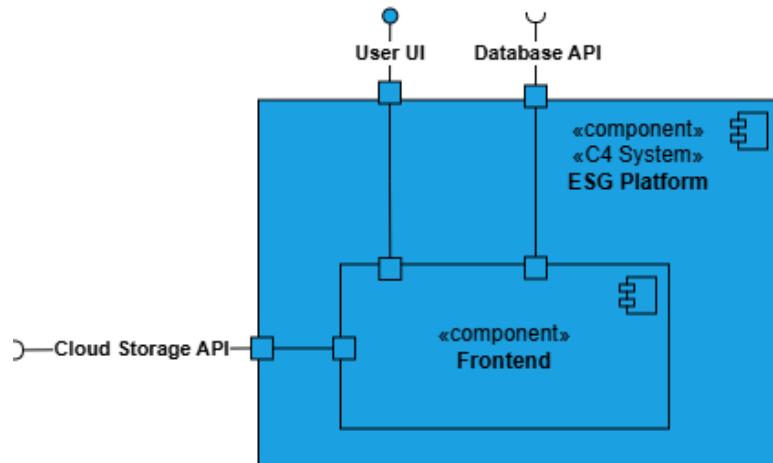


Figura 4.14: Vista Lógica (Nível 2) (autoria própria)

Já o Nível 3 (Figura 4.15) representa os componentes que compõem os contentores do sistema, tornando mais claras as interações entre eles.

Observa-se que o contentor *Frontend* consome duas APIs: uma disponibilizada pelo *Database Provider* e outra pelo *Cloud Storage Provider*.

O *Cloud Storage Provider* é utilizado para armazenar ficheiros estáticos (como imagens) e gerar os seus *Uniform Resource Locator (URL)* de acesso público. Estes URLs são posteriormente armazenados na base de dados.

O *Database Provider*, por sua vez, utiliza uma base de dados relacional. A responsabilidade de persistir os dados, incluindo os URLs gerados, cabe ao contentor *Frontend*, que interage diretamente com ambas as APIs.

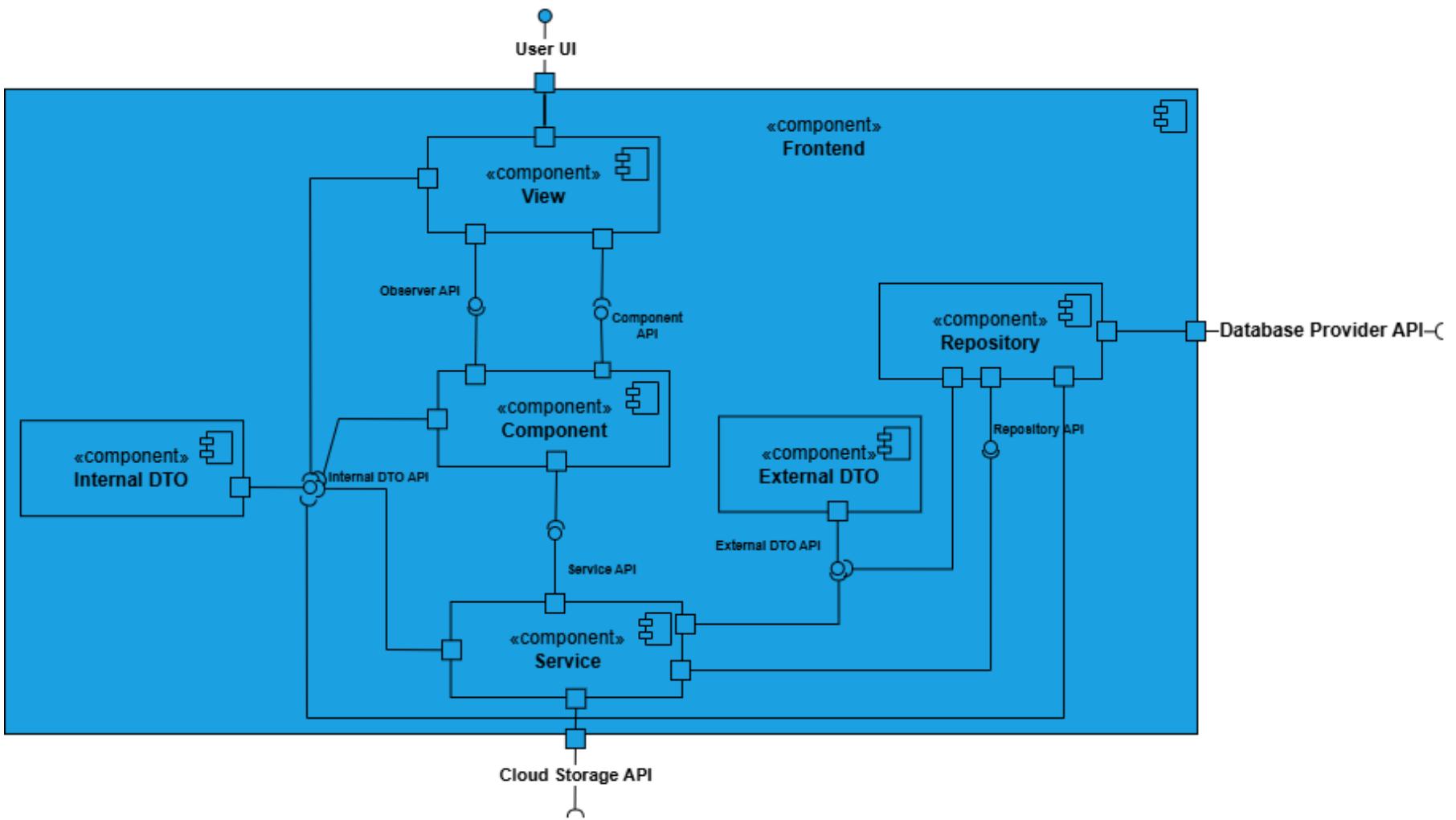


Figura 4.15: Vista Lógica (Nível 3) (autoria própria)

#### 4.1.7 Vista de Implementação

Semelhante à subseção anterior, esta vista será detalhada somente com um diagrama de Nível 3, representado pela Figura 4.16.

O Nível 1 não consta neste relatório por se tratar de uma representação demasiado abstrata, que pouco acrescentaria ao entendimento do sistema em questão. O Nível 2 não oferece nenhuma informação acrescida comparativamente ao diagrama do mesmo nível da vista lógica (Figura 4.14).

O Nível 3, de granularidade mais fina, elabora conteúdo do módulo *Frontend*, dando um maior entendimento não só na sua constituição mas também nas diferentes interações entre os seus elementos.

A aplicação Plataforma ESG é constituída por uma *React SPA* para o *Frontend*, construída para suportar a geração atual de *browsers* para *desktop*.

Todas as ações realizadas pelos utilizadores têm origem no *Frontend*, que comunica com o *Database Provider* para obter ou inserir os dados necessários. No caso específico da criação de uma nova métrica, o *Frontend* interage adicionalmente com o *Cloud Storage Provider*, onde ocorre o *upload* da imagem representativa da respetiva métrica.

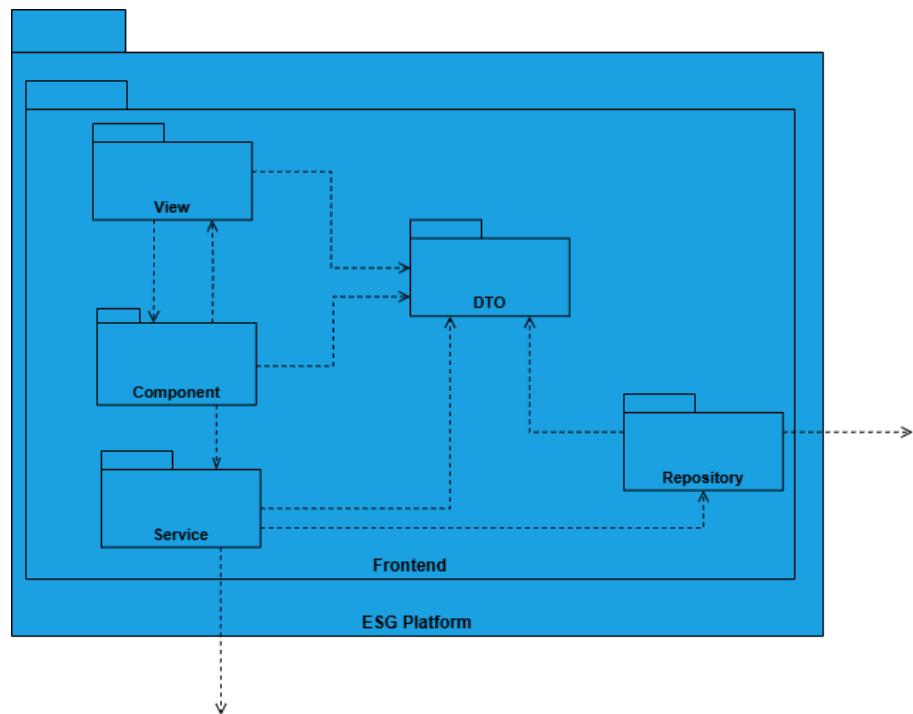


Figura 4.16: Vista de Implementação (Nível 3) (autoria própria)

O Apêndice B apresenta o mapeamento entre a vista lógica e a de implementação de Nível 3 de modo a facilitar o entendimento das relações entre as duas vistas.

#### 4.1.8 Vista Física

A vista física tem como objetivo mapear o *software* para o *hardware*, com ênfase nos requisitos não funcionais do sistema, como disponibilidade, confiabilidade (tolerância a falhas),

desempenho e escalabilidade (Kruchten 1995). A Figura 4.17 apresenta o diagrama de Nível 2 da vista física da solução desenvolvida.

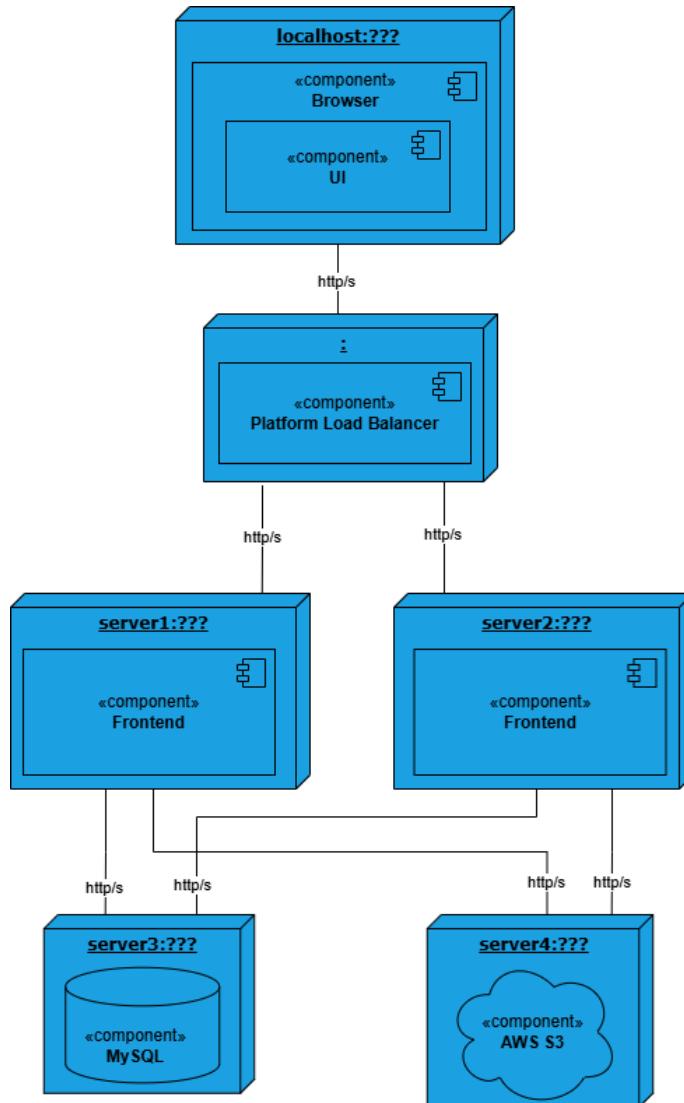


Figura 4.17: Diagrama de Nível 2 da vista física da solução (autoria própria)

Os *Load Balancers* são responsáveis por distribuir o tráfego de forma dinâmica entre os servidores que compõem as camadas de *Frontend* e *Data Management* (*backend* com base de dados). Estes atuam como intermediários entre os utilizadores e os servidores, assegurando uma distribuição equilibrada da carga, especialmente durante períodos de maior tráfego. Com isso, aumentam a disponibilidade e a resiliência do sistema: caso um servidor falhe, o tráfego é automaticamente redirecionado para outro servidor ativo (tolerância a falhas). Esta abordagem contribui ainda para a redução da latência e para evitar respostas inconsistentes da aplicação (F5 2025).

As comunicações entre os vários componentes são realizadas através de protocolo HTTP ou HTTPS.

## 4.2 Alternativa Arquitetural: DDD com Clean Architecture

Uma alternativa à arquitetura representada pelos modelos C4/4+1 seria o uso de *Domain-Driven Design* (DDD) combinado com os princípios da *Clean Architecture*. Esta abordagem também permite uma estrutura modular e escalável, alinhada aos limites naturais do domínio e preparada para futuras expansões.

Como mostra a Figura 4.18, o sistema é dividido em módulos independentes, cada um representando um conceito central da aplicação ESG: *Dataset*, *Metric*, *Objective*, *Pillar* e *Subarea*. Cada módulo encapsula as suas entidades, regras de negócio e casos de uso, comunicando-se com o exterior por meio de interfaces (*ports*), enquanto os adaptadores (como base de dados ou APIs) implementam essas interfaces.

A estrutura segue o modelo em anéis concêntricos da *Clean Architecture*, isolando o núcleo de domínio das dependências externas. Os casos de uso coordenam a lógica na camada de aplicação, e as camadas mais externas lidam com *frameworks* como *Prisma* ou *Next.js*.

Esta organização garante separação clara de responsabilidades, menor acoplamento, maior testabilidade e independência tecnológica. Estes são fatores que contribuem para um código mais limpo, coeso e sustentável.

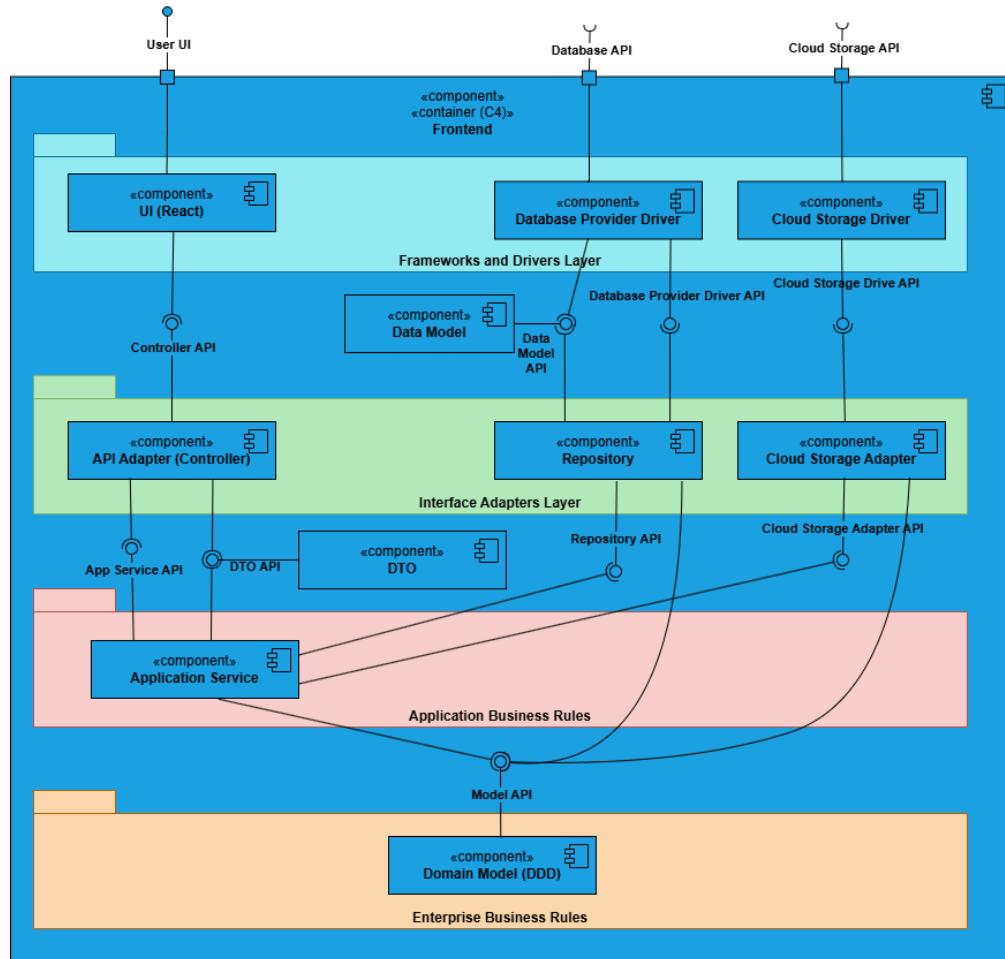


Figura 4.18: Diagrama de Nível 3 da vista lógica do *container Frontend* segundo uma arquitetura de camadas concêntricas (autoria própria)



## Capítulo 5

# Implementação da Solução

A fase de implementação/desenvolvimento de *software* segue as diretrizes definidas na análise e no *design* do sistema, traduzindo-as em código-fonte e linguagem de programação (UBIMINDS 2025). É nesta etapa que se integram sistemas externos e se utilizam ferramentas que auxiliam na programação, rastreio de código, colaboração da equipa, gestão de tarefas e cumprimento de prazos e metas estabelecidos.

Neste capítulo, será descrito o processo de implementação do projeto ao longo de várias fases, nomeadamente as fases de planeamento e de desenvolvimento. Serão também referidas as tecnologias utilizadas, incluídas imagens representativas da plataforma criada e explicadas as funcionalidades implementadas, conforme definidas nos casos de uso apresentados nos Capítulos 3 e 4. Detalhes considerados desnecessários para a compreensão do trabalho serão remetidos para os anexos.

## 5.1 Descrição da implementação

### 5.1.1 Tecnologias Usadas

Nesta subsecção são descritas as tecnologias usadas na implementação, categorizadas entre pré-desenvolvimento, referentes ao planeamento desta fase, e desenvolvimento, relativamente à implementação da solução.

#### Tecnologias de Pré-desenvolvimento

No desenvolvimento dos diagramas da vista de progresso (Subsecção 4.1.4) que servem de guia para a implementação das funcionalidades foi usado o *PlantUML*<sup>1</sup> compilado no *Visual Studio Code*<sup>2</sup>. Os restantes diagramas foram feitos pelo programa *Draw.io*<sup>3</sup>.

O protótipo visual da plataforma ESG, referido no Apêndice C, foi realizado no *Figma*<sup>4</sup>, a divisão de tarefas e planeamento do projeto foi realizado no *Github Projects*<sup>5</sup> e o repositório foi alogado no *Github*<sup>6</sup>. A aplicação tem como referência visual o *kit* de marca da Devscope<sup>7</sup>.

O Apêndice D refere-se ao quadro *kanban* usado durante o desenvolvimento da solução.

<sup>1</sup>Site do *Plantuml*: <https://plantuml.com/>

<sup>2</sup>Site do *Visual Studio Code*: <https://code.visualstudio.com/>

<sup>3</sup>Site do *Drawio*: <https://www.drawio.com/>

<sup>4</sup>Site do *Figma*: <https://www.figma.com/>

<sup>5</sup>Página sobre o *Github Projects*: <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>

<sup>6</sup>Site do *Github*: <https://github.com/>

<sup>7</sup>Site do *kit* de marca da Devscope: <https://brand.devscope.net/intro.html>

## Tecnologias de Desenvolvimento

Relativamente às tecnologias utilizadas no desenvolvimento do projeto, o *Visual Studio Code* foi o editor escolhido para a escrita do código-fonte.

No que diz respeito ao *frontend*, recorreu-se ao *TypeScript* como linguagem de programação de alto nível, de código aberto<sup>8</sup>. O *TypeScript* é uma versão melhorada do *JavaScript*, orientada a objetos, sendo preferida em projetos de maior escala devido à sua robustez, melhor manutenção e escalabilidade.

Para o desenvolvimento da interface, utilizaram-se as tecnologias *React*<sup>9</sup>, uma biblioteca *JavaScript* focada na construção de interfaces de utilizador, especialmente em aplicações do tipo SPA e o *Next.js*<sup>10</sup>, uma *framework* baseada em *React* que facilita a criação de aplicações *web* com funcionalidades como renderização no lado do servidor (*Server-Side Rendering*) e geração estática de páginas.

A nível de estilização, foi usada a *framework* *Tailwind CSS*<sup>11</sup>, que permite desenvolver rapidamente interfaces personalizadas através da aplicação de classes utilitárias diretamente no HTML, evitando a necessidade de escrever folhas de estilo tradicionais.

No *frontend*, foi utilizado o *Vuexy*<sup>12</sup> como base para o desenvolvimento do painel de administração. Este *template* de *dashboard* é altamente personalizável e compatível com tecnologias como *Vue.js*<sup>13</sup>, *React*, *HTML* e *Laravel*<sup>14</sup>, oferecendo componentes reutilizáveis, gráficos e estruturas de *layout*. Para a visualização interativa dos dados, foi utilizada a biblioteca *ApexCharts*<sup>15</sup>, uma solução *open source* baseada em *JavaScript*.

A escolha de todas as tecnologias mencionadas foi uma exigência da empresa onde decorreu o estágio, assegurando a compatibilidade do projeto desenvolvido com os restantes produtos e soluções da organização.

A plataforma *Railway*<sup>16</sup> foi utilizada para alojar a base de dados relacional *MySQL*<sup>17</sup>. Esta permite integração direta com repositórios *Git*, facilitando a integração contínua e a publicação. Para além disso, trata automaticamente da infraestrutura e escalabilidade, sendo útil no desenvolvimento e manutenção de aplicações *web*.

Para o armazenamento de ficheiros estáticos, como imagens associadas às métricas ESG, foi utilizado o serviço da *Amazon AWS S3*<sup>18</sup>, onde foi criado um *bucket* próprio para o projeto. As imagens são alojadas nesse *bucket*, sendo o respetivo URL público guardado no atributo `imageURL` da tabela de métricas ESG da base de dados *MySQL*.

A interação com a base de dados foi facilitada pelo uso do *Prisma*<sup>19</sup>, uma ferramenta ORM (*Object-Relational Mapping*) que permite definir o esquema da base de dados de forma declarativa, através de um ficheiro `schema.prisma`. Para além disso, o *Prisma* foi utilizado

---

<sup>8</sup>Site oficial do *TypeScript*: <https://www.typescriptlang.org/>

<sup>9</sup>Site do *React*: <https://react.dev/>

<sup>10</sup>Site do *Next.js*: <https://nextjs.org/>

<sup>11</sup>Site do *Tailwind CSS*: <https://tailwindcss.com/>

<sup>12</sup>Site oficial do *Vuexy*: <https://pixinvent.com/vuexy-bootstrap-html-admin-template/>

<sup>13</sup>Site oficial do *Vue.js*: <https://vuejs.org/>

<sup>14</sup>Site oficial do *Laravel*: <https://laravel.com/>

<sup>15</sup>Site oficial do *ApexCharts*: <https://apexcharts.com/>

<sup>16</sup>Site oficial: <https://railway.com/>

<sup>17</sup>Site oficial: <https://www.mysql.com/>

<sup>18</sup>Site do *AWS S3*: <https://aws.amazon.com/s3/>

<sup>19</sup>Site do *Prisma*: <https://www.prisma.io/>

na criação de *scripts* de *seeding*, que permitem inserir dados fictícios na base de dados, e também para a limpeza e eliminação de dados durante o processo de desenvolvimento.

## 5.2 Funcionalidades

### 5.2.1 Página Inicial | Dashboard

A página inicial, Figura 5.1, apresenta a informação mais recente e dinâmica da plataforma, incluindo a pontuação ESG e os principais KPI's. Esta implementação abrange os casos de uso UC-01 (Visualização de KPI's) e UC-04 (Pontuação ESG), destacando as métricas que registaram melhorias ou quedas em relação ao valor anterior.

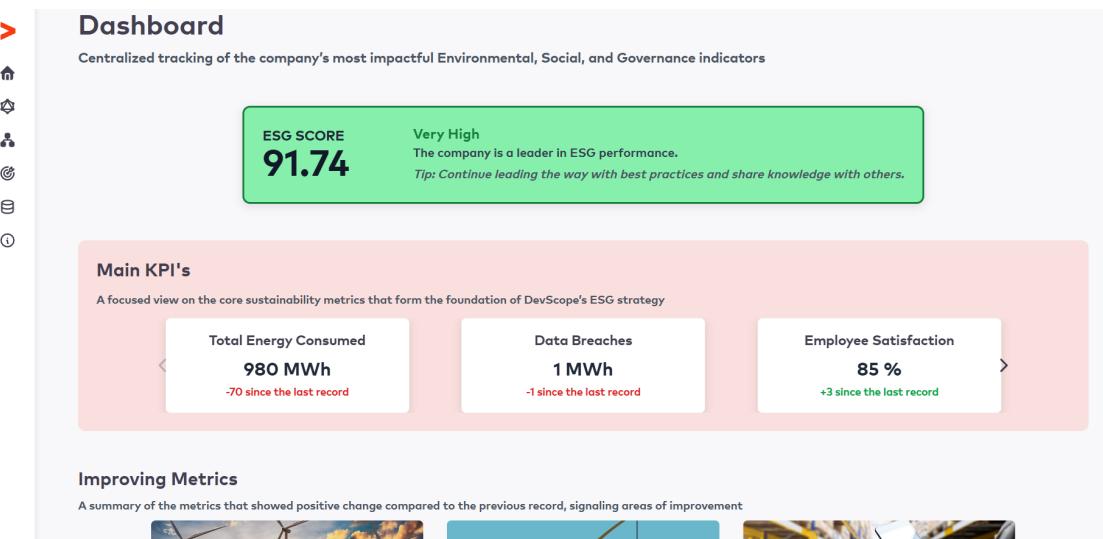


Figura 5.1: Página Inicial com pontuação ESG (autoria própria)

O valor obtido na pontuação ESG surge de uma fórmula criada com base no artigo redigido por Berg, Kölbel e Rigobon 2022, que explora as divergências das várias metodologias para classificações ESG usadas pelas agências na classificação do desempenho ESG das empresas. Tendo em conta o conteúdo do artigo, uma nova fórmula foi desenvolvida. A pontuação ESG é calculada através de uma abordagem hierárquica que agrupa métricas em subáreas, subáreas em pilares, e finalmente combina os pilares para obter a pontuação final. O processo segue as seguintes etapas:

#### Normalização de Métricas

Cada métrica é normalizada para uma escala de 0 a 100 ao utilizar os seus valores mínimo e máximo definidos:

$$\text{valor\_normalizado} = \begin{cases} \frac{\text{valor\_bruto} - \text{minValue}}{\text{maxValue} - \text{minValue}} \times 100 & \text{se minValue e maxValue existirem} \\ \text{valor\_bruto} & \text{caso contrário} \end{cases} \quad (5.1)$$

### Cálculo de Pontuação por Subárea

Para cada subárea, calcula-se a média das métricas ativas (com *activationStatus* verdadeiro) e aplica-se o peso da subárea:

$$\text{pontuação\_subarea} = \left( \frac{\sum_{i=1}^n \text{valor\_normalizado}_i}{n} \right) \times \text{peso\_subarea} \quad (5.2)$$

onde  $n$  é o número de métricas ativas na subárea.

### Cálculo de Pontuação por Pilar

A pontuação de cada pilar é calculada como a média ponderada das subáreas pertencentes a esse pilar:

$$\text{pontuação\_pilar} = \left( \frac{\sum_{j=1}^m \text{pontuação\_subarea}_j}{\sum_{j=1}^m \text{peso\_subarea}_j} \right) \times \text{peso\_pilar} \quad (5.3)$$

onde  $m$  é o número de subáreas no pilar.

### Cálculo da Pontuação ESG Final

A pontuação final ESG é obtida pela média ponderada dos pilares, limitada ao intervalo [0, 100]:

$$\text{pontuação\_final} = \min \left( 100, \max \left( 0, \frac{\sum_{k=1}^p \text{pontuação\_pilar}_k}{\sum_{k=1}^p \text{peso\_pilar}_k} \right) \right) \quad (5.4)$$

onde  $p$  é o número total de pilares considerados (Ambiental, Social e Governança).

A pontuação final é posteriormente classificada numa das categorias predefinidas, como indicado na Tabela 5.1. Esta Tabela têm uma formatação diferente das restantes, uma vez que cada linha está colorida com a cor associada à respetiva categoria.

<b>Very Low</b>	[0–20]	A empresa necessita de melhorias significativas nas suas práticas ESG.
<b>Low</b>	[21–40]	Há margem de progressão, especialmente nas áreas de responsabilidade social e sustentabilidade ambiental.
<b>Medium</b>	[41–60]	A empresa está no caminho certo, mas ainda precisa de melhorias substanciais.
<b>Good</b>	[61–80]	A empresa apresenta um bom desempenho ESG, com espaço para ajustes menores.
<b>Great</b>	[81–90]	A empresa destaca-se nas práticas ESG, estando próxima da excelência.
<b>Very High</b>	[91–100]	A empresa é uma referência no desempenho ESG e serve como modelo de boas práticas.

Tabela 5.1: Categorias de pontuação ESG (autoria própria)

Estas categorias não só representam a qualidade geral da performance ESG da empresa, como também enciam a forma como os dados são apresentados no *frontend* da aplicação, através de diferentes cores, textos e recomendações visuais para cada cenário. Esta abordagem contribui para uma demonstração mais realista da plataforma, permitindo visualizar comportamentos distintos consoante o per ESG simulado.

No *frontend*, os dados do *dashboard* são obtidos dinamicamente através de uma chamada HTTP do tipo GET, utilizando a função `fetch` (Listagem 5.1) para aceder à rota `/api/dashboard`. A resposta, em formato *JavaScript Object Notation (JSON)*, agrupa todas as métricas necessárias, evitando múltiplas chamadas à API e melhorando a performance da aplicação.

```

1  useEffect(() => {
2    document.title = 'Dashboard | ESG Platform';
3    const fetchData = async () => {
4      try {
5        const responseScore = await fetch('/api/esgScore');
6        if (!responseScore.ok) {
7          throw new Error('Failed to fetch ESG score');
8        }
9        const dataScore = await responseScore.json();
10       setEsgScore(dataScore);
11       setCategory(classifyESGScore(dataScore));
12       const responseData = await fetch('/api/dashboard');
13       if (!responseData.ok) {
14         throw new Error('Failed to fetch dashboard info');
15       }
16       const dashboardData = await responseData.json();
17       setKPIs(dashboardData.kpis);
18       setBestN(dashboardData.bestMetrics);
19       setWorstN(dashboardData.worstMetrics);
20     } catch (err: any) {
21       setError(err.message);
22     } finally {
23       setIsLoading(false);
24     }
25   };
26   fetchData();
27 }, []);

```

Listagem 5.1: Função responsável por obter os dados do ESG Score e do dashboard no carregamento do componente

De modo a melhorar a experiência do utilizador relativamente à espera do retorno da informação requisitada foi desenvolvido um excerto HTML (Listagem 5.2) que escurece levemente a plataforma e introduz um componente circular a indicar o progresso de obtenção das informações.

```

1  if (isLoading) {
2    return (
3      <Backdrop
4        open={true}
5        sx={{
6          backgroundColor: 'rgba(255, 255, 255, 0.7)',
7          zIndex: (theme) => theme.zIndex.modal + 1,
8          display: 'flex',
9          flexDirection: 'column',
10         gap: 2
11       }}
12     >
13       <CircularProgress color="primary" size={60} />
14     </Backdrop>
15   )
16 }

```

Listagem 5.2: Componente mostrado no carregamento da página

O mesmo foi feito caso acontecesse algum erro, como demonstrado na Listing 5.3.

```

1  if (error) {
2    return (
3      <Dialog open={true}>
4        <DialogTitle>Error</DialogTitle>
5        <DialogContent>
6          <div style={{ color: 'red', padding: '16px' }}>
7            Error: {error}
8          </div>
9        </DialogContent>
10       <DialogActions>
11         <Button onClick={() => setError(null)}>Close</Button>
12       </DialogActions>
13     </Dialog>
14   )
15 }

```

Listagem 5.3: Componente mostrado no caso de erro no carregamento da página

Os KPIs apresentam os valores atuais das métricas classificadas como tal, assim como a sua evolução desde o último registo. Esta variação é destacada com um código de cores: verde para evolução positiva e vermelho para desempenho negativo.

A distinção entre as melhores e piores métricas é feita com base na evolução das métricas ativas (com *activationStatus* verdadeiro), calculada entre o primeiro e o último valor registrado. O delta obtido determina a inclusão da métrica no conjunto das cinco melhores ou cinco piores.

### 5.2.2 Página das Métricas

Esta página (Figura 5.2) da plataforma agrupa várias informações e funcionalidades, nomeadamente as que são referentes aos casos de uso UC-05 (Visualização dos detalhes das métricas), UC-02 (Filtragem de métricas), e UC-06 (Métricas Customizáveis).

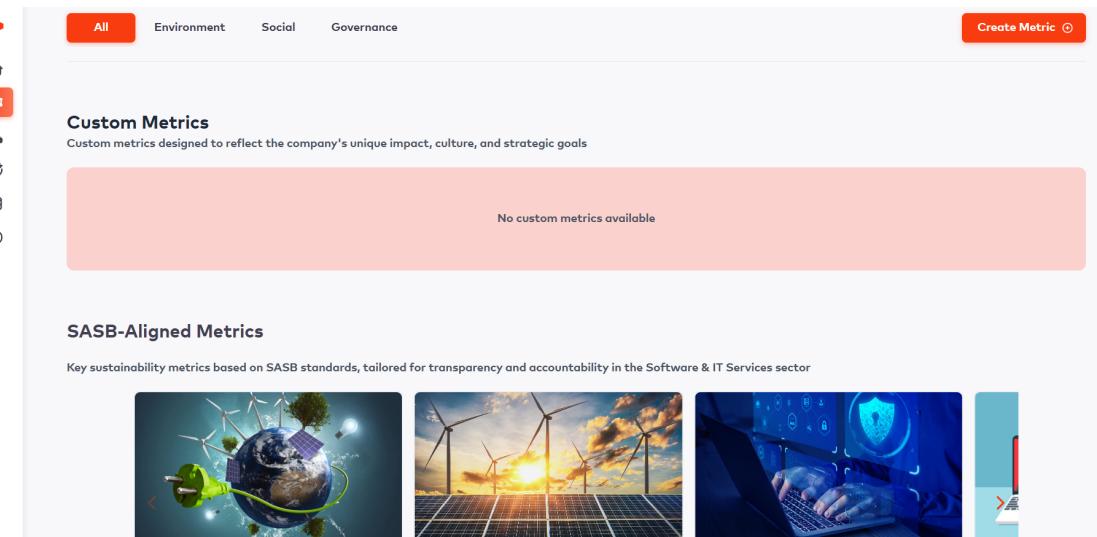


Figura 5.2: Página das Métricas (autoria própria)

A interface da aplicação apresenta um conjunto de separadores (*tabs*) que representam os diferentes pilares ESG, funcionando como mecanismo de filtragem das métricas (UC-02). Por defeito, todos os pilares estão selecionados, permitindo visualizar a totalidade das

métricas disponíveis. A Listagem 5.4 ilustra a lógica de filtragem implementada: caso o valor de `selectedPillar` seja 'All', são apresentadas todas as métricas; caso contrário, apenas são consideradas as métricas associadas ao pilar selecionado.

```

1 const filteredMetrics = useMemo(() => {
2   return selectedPillar === 'All'
3     ? metrics
4     : metrics.filter((metric) => metric.subarea?.pillar?.name === selectedPillar)
5 }, [metrics, selectedPillar])

```

Listagem 5.4: Excerto de Código com Filtragem das Métricas

Adicionalmente, é feita uma distinção clara entre as métricas definidas pela framework SASB, específicas para o setor de *Software* e *Serviços IT*, e as métricas personalizadas criadas pela empresa (UC-06). Esta criação é realizada através de um *popup*, ilustrado na Figura 5.3, permitindo adaptar a plataforma às necessidades da organização. As informações submetidas são enviadas via API e armazenadas na base de dados.

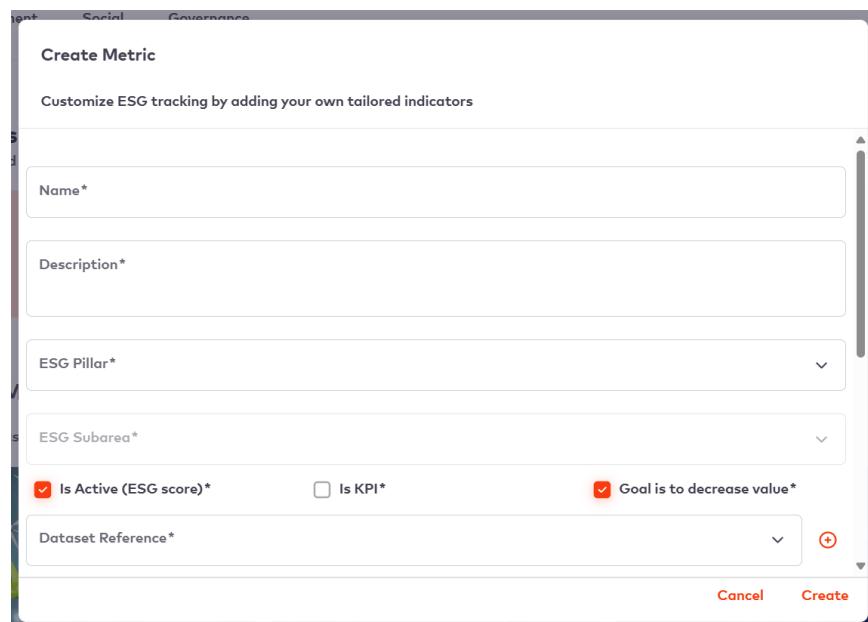


Figura 5.3: Formulário de Criação de Métrica (autoria própria)

Tendo em conta que cada métrica está relacionada com uma imagem e que o armazenamento de imagens como BLOB em bases de dados não é aconselhável, por questões de desempenho e escalabilidade, foi implementada uma solução híbrida (code-examples.net 2025).

As imagens são armazenadas no serviço AWS S3, que é perfeito para ficheiros estáticos, e o URL gerado é guardado na base de dados, com o objetivo de ser utilizado na renderização do cartão da métrica (Listagem 5.5).

```

1 export async function generateSignedUploadUrl(fileName: string, fileType: string) {
2   const key = `${folder}/${fileName}`
3   const command = new PutObjectCommand({
4     Bucket: bucket,
5     Key: key,
6     ContentType: fileType,
7   })
8   const url = await getSignedUrl(s3, command, { expiresIn: 60 })
9   return { url, key, bucket }
10 }

```

Listagem 5.5: Função de *Upload* de uma imagem para o AWS S3

Cada cartão representante de uma métrica é interativo. Quando clicado, abre uma janela com as informações da métrica (UC-05), como ilustrado pela Figura 5.4. Esta janela mostra os diferentes atributos das métricas.

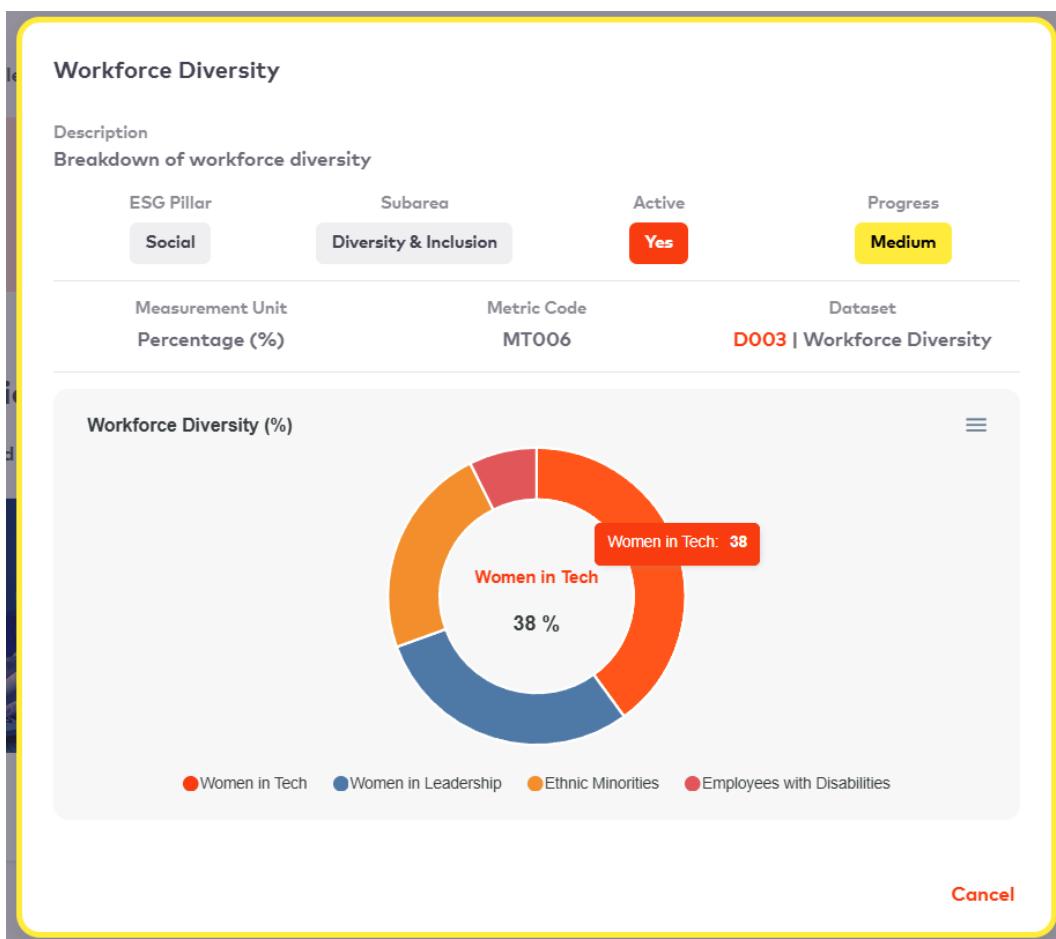


Figura 5.4: Cartão com informação detalhada da Métrica (autoria própria)

### 5.2.3 Mapa de Materialidade

O mapa de materialidade, abordado no Capítulo 2, corresponde a uma matriz que agrupa as métricas consideradas relevantes para a avaliação ESG da empresa. A plataforma apresenta este elemento numa página dedicada, Figura 5.5, organizada por pilares, subáreas e métricas ativas. O objetivo é fornecer uma visualização clara das áreas cujo progresso ainda se encontra aquém do desejável.

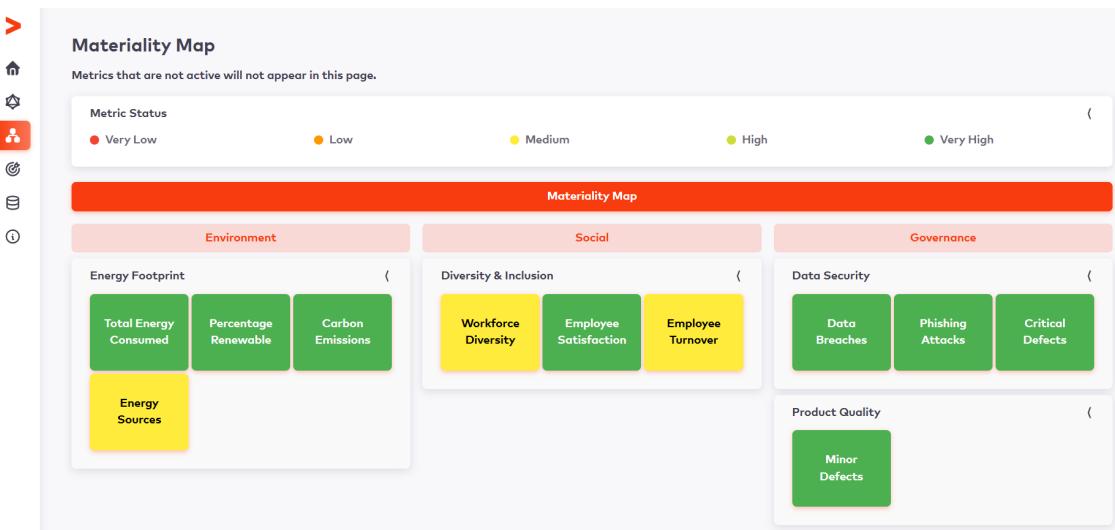


Figura 5.5: Página da Matriz da Materialidade (autoria própria)

Tal como na página das métricas, cada cartão é interativo e apresenta informações da métrica, sendo colorido conforme o seu progresso. A Listagem 5.6 mostra a função que classifica esse progresso e define a cor correspondente.

```

1 export function getMetricDevelopmentStatus(
2   values: MetricValueHistory[], goalIsDecrease: boolean,
3   options = { recentPoints: 5 }
4 ): { status: MetricStatus; color: string } {
5   if (values.length < 2) return { status: 'Medium', color: nodeColors['Medium'] };
6
7   const recent = values
8     .map(h => ({ ...h, recordedAt: new Date(h.recordedAt) }))
9     .sort((a, b) => a.recordedAt.getTime() - b.recordedAt.getTime())
10    .slice(-options.recentPoints);
11
12  const [first, last] = [recent[0].value, recent.at(-1)!.value];
13  const change = ((goalIsDecrease ? first - last : last - first) / first) * 100;
14
15  if (change >= 10) return { status: 'Very High', color: nodeColors['Very High'] };
16  if (change >= 5) return { status: 'High', color: nodeColors['High'] };
17  if (change > -5) return { status: 'Medium', color: nodeColors['Medium'] };
18  if (change > -10) return { status: 'Low', color: nodeColors['Low'] };
19  return { status: 'Very Low', color: nodeColors['Very Low'] };
20}

```

Listagem 5.6: Classificação do progresso de uma métrica

#### 5.2.4 Página dos Conjuntos de Dados

Cada métrica tem o seu conjunto de dados associado, estando estes acessíveis noutra página da plataforma (Figura 5.6), organizados como entradas numa tabela.

**Dataset Collection**

Add Dataset Filter

ID ↑	Name	ESG Pillar	Type	Imported At
D001	Energy Consumption Metrics	Environment	Time Series	24 Jun 2025
D002	Security Incident Metrics	Governance	Time Series	24 Jun 2025
D003	Workforce Diversity	Social	Single Value	24 Jun 2025
D004	Carbon Emissions	Environment	Time Series	24 Jun 2025
D005	Employee Satisfaction	Social	Time Series	24 Jun 2025

Rows per page: 5 1-5 of 8 < >

Dense padding

Figura 5.6: Página dos Conjuntos de Dados (autoria própria)

É nesta página que se observa a implementação do caso de uso UC-07 (Importação de Dados), através de uma janela ilustrada na Figura 5.7, onde são solicitadas as informações necessárias, incluindo o ficheiro CSV que contém os dados. Consoante a estrutura do ficheiro, a interface apresenta diferentes visualizações do mesmo conjunto de dados sob a forma de vários tipos de diagramas. Compete ao utilizador selecionar o tipo de diagrama que melhor representa a informação.

**Dataset Import**

Uploaded: Employee Sustainability Participation.csv  
Supported Charts: line, area, bar, pie, donut, radial

Only CSV files using commas (,) as delimiters are supported for import.

Supported Chart Types

Preferred Chart Type\*  
area

line Energy Report (MWh)  
area Energy Report (MWh)  
bar Energy Report (MWh)

MW h Jan 2023 Feb 2023  
Recycling Program  
Carpool Initiative

MW h Jan 2023 Feb 2023  
Recycling Program  
Carpool Initiative

MW h Jan 2023  
Recycling Program  
Carpool Initiative

Create

Figura 5.7: Formulário de Importação de um Conjunto de Dados com Múltipla Renderização de Diagramas (autoria própria)

Após o preenchimento do formulário de importação de dados, é realizado um pedido HTTP do tipo *POST* através da função `fetch`, com o objetivo de enviar a informação inserida para a rota da API correspondente. A informação submetida é convertida em formato *JSON* e enviada no corpo do pedido, tal como ilustrado na Listagem 5.7.

```

1 const response = await fetch('/api/datasets', {
2   method: 'POST',
3   headers: { 'Content-Type': 'application/json' },
4   body: JSON.stringify(newDataset)
5 })
6 if (!response.ok) {
7   const text = await response.text()
8   throw new Error(`Failed to import dataset: ${text}`)
9 }
10 const createdDataset: DatasetReadable = await response.json()

```

Listagem 5.7: Chamada à API de Importação de um Conjunto de Dados

A página permite ainda fazer a filtragem dos conjuntos de dados (Figura 5.8) mediante o pilar ESG a que estão associados e à sua tipologia.

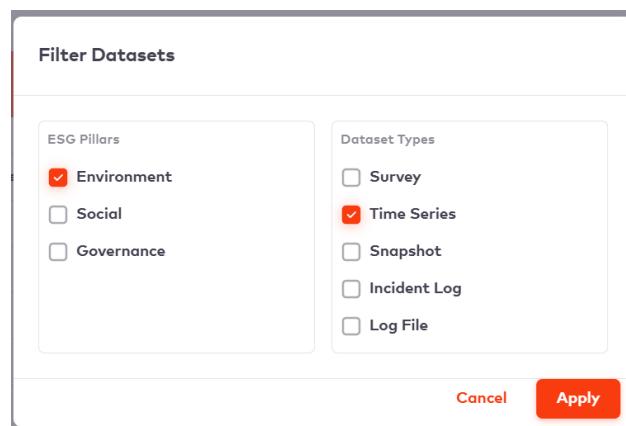


Figura 5.8: Filtragem de Conjunto de Dados (autoria própria)

A Listagem 5.8 apresenta a função que faz o pedido GET onde são aplicados os filtros aos conjuntos de dados com base nos parâmetros definidos na *query url*.

```

1 const handleApplyFilters = async (filters: { pillars: string[], types: string[] }) => {
2   const queryParams = new URLSearchParams()
3   filters.pillars.forEach(pillar => queryParams.append('pillars', pillar))
4   filters.types.forEach(type => queryParams.append('types', type))
5   const response = await fetch('/api/datasets/filter?${queryParams.toString()}', {
6     method: 'GET',
7     headers: { 'Content-Type': 'application/json' }
8   })
9   if (!response.ok) {
10     console.error('Error fetching filtered datasets')
11     return
12   }
13   const data = await response.json()
14   setDatasets(data)
15 }

```

Listagem 5.8: Função de Filtragem de Conjunto de Dados

É ainda possível visualizar a informação de cada conjunto de dados, o seu conteúdo e alternar entre renderizações de diferentes tipos de diagramas, como ilustrado na Figura 5.9.

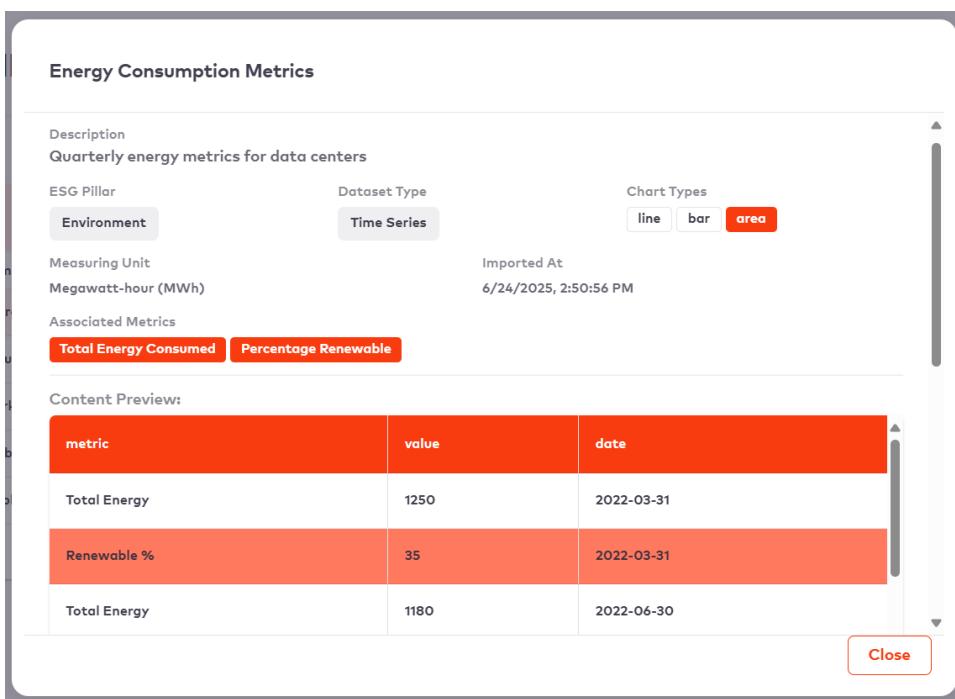


Figura 5.9: Informação de um Conjunto de Dados (autoria própria)

### 5.2.5 Página dos Objetivos

Esta página da plataforma, ilustrada na Figura 5.10, complementa os casos de uso definidos no projeto, tendo sido desenvolvida por iniciativa da empresa. Na parte superior, apresenta um resumo do estado dos objetivos, indicando quantos foram concluídos, quantos permanecem por cumprir, a percentagem que revela bom progresso e a percentagem que se encontra fora do ritmo esperado.

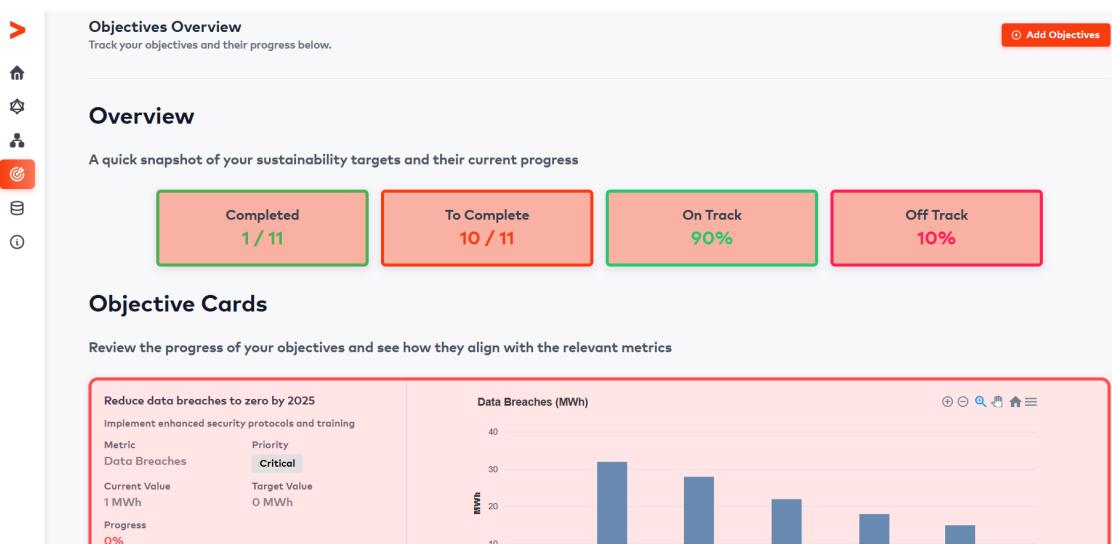


Figura 5.10: Página de Gestão de Objetivos (autoria própria)

A funcionalidade principal desta página é a criação e acompanhamento de objetivos associados a métricas, bem como a avaliação do seu progresso. Cada objetivo é representado

num cartão individual, Figura 5.11, onde se apresentam as respetivas informações, o nível de progresso e um gráfico com os dados correspondentes. Estes cartões são organizados por ordem de prioridade ou importância atribuída ao objetivo.

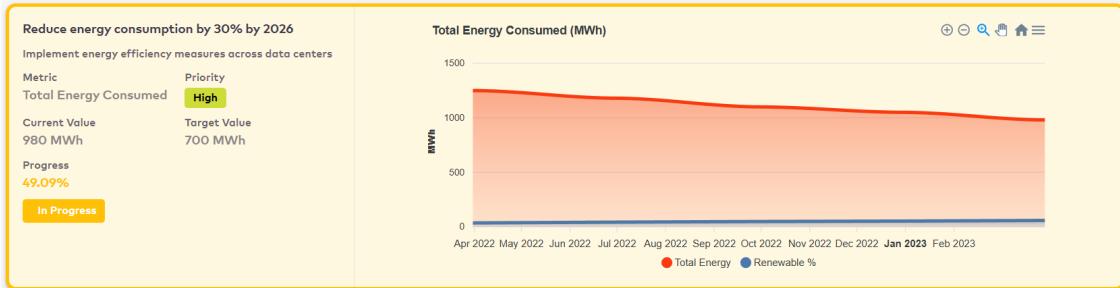


Figura 5.11: Cartão com Informação Detalhada de um Objetivo (autoria própria)

A criação de um novo objetivo é iniciada com a submissão de um formulário, que desencadeia um pedido POST à API responsável por registar o objetivo. O envio dos dados é feito em formato JSON, e a resposta da API é validada para garantir o sucesso da operação. Em caso de erro, é apresentada uma mensagem informativa, como indicado na Listagem 5.9.

```

1 const response = await fetch('/api/objectives', {
2   method: 'POST',
3   headers: { 'Content-Type': 'application/json' },
4   body: JSON.stringify(data),
5 });
6 if (!response.ok) {
7   const errorText = await response.text();
8   throw new Error(`Failed to create new objective: ${errorText}`);
9 }
10 const createdObjective = await response.json() as ObjectiveReadable;

```

Listagem 5.9: Pedido POST para criar um novo Objetivo

O formulário criado para a criação de novos objetivos é ilustrado pela Figura 5.12.

The figure shows a modal dialog titled "Create Objective". The header includes the instruction "Set custom ESG objectives to benchmark against key metrics". The form contains several input fields: "Name\*" (text input), "Description\*" (text input), "Priority\*" (dropdown menu with options like Low, Medium, High, Critical), "Comparative Metric\*" (dropdown menu), "Target Value\*" (text input), and "Deadline\*" (date picker input with placeholder "Click to select a date and time"). At the bottom right are "Cancel" and "Create" buttons.

Figura 5.12: Formulário de Criação de Objetivo (autoria própria)

### 5.3 Inicialização da Base de Dados

De forma a tornar a demonstração da plataforma mais rica e realista, foram desenvolvidos diversos *scripts* para preparar o estado da base de dados. Estes *scripts* permitem simular diferentes cenários de utilização e facilitar a visualização dos resultados obtidos no *dashboard*.

Inicialmente, foi criado um *script* de limpeza (Listagem 5.10) responsável por remover todos os registos existentes na base de dados. Este processo garante que o ambiente de demonstração começa sempre num estado limpo e previsível, evitando interferência de dados anteriores.

```

1 const prisma = new PrismaClient()
2 async function clearDatabase() {
3   console.log('Clearing database...')
4   await prisma.objective.deleteMany({})
5   await prisma.metric.deleteMany({})
6   await prisma.subarea.deleteMany({})
7   await prisma.dataset.deleteMany({})
8   await prisma.measureUnit.deleteMany({})
9   await prisma.pillar.deleteMany({})
10  await prisma.company.deleteMany({})
11  await prisma.sector.deleteMany({})
12  console.log('Database cleared!')
13}
14 clearDatabase()
15 .catch(e => {
16   console.error('Error clearing database:', e)
17   process.exit(1)
18 })
19 .finally(async () => {
20   await prisma.$disconnect()
21 })
```

Listagem 5.10: *Script* de limpeza da Base de Dados

De seguida, foi implementado um *script* principal de *seeding* (Listagem 5.11) que permite popular a base de dados com dados simulados, com base num cenário definido através da variável de ambiente `SEED_SCENARIO`. Este parâmetro pode assumir os valores `good`, `bad` ou `random`, correspondendo a diferentes perfis de desempenho ESG.

```

1 const prisma = new PrismaClient()
2 const SEED_SCENARIO = process.env.SEED_SCENARIO || 'random'
3
4 async function main() {
5   if (SEED_SCENARIO === 'good') {
6     console.log('Seeding good scenario...')
7     await seedGood(prisma)
8   } else if (SEED_SCENARIO === 'bad') {
9     console.log('Seeding bad scenario...')
10    await seedBad(prisma)
11  } else {
12    console.log('Seeding random scenario...')
13    await seedRandom(prisma)
14  }
15}
16
17 main()
18 .catch(e => {
19   console.error(e)
20   process.exit(1)
21 })
22 .finally(async () => {
23   await prisma.$disconnect()
24 })
```

Listagem 5.11: *Script* de *seeding* da Base de Dados

Ao executar o *script*, o cenário selecionado determina qual das funções de *seeding* será invocada: `seedGood`, `seedBad` ou `seedRandom`. Cada uma destas funções é responsável por gerar dados coerentes com o respetivo perfil, permitindo avaliar o comportamento da aplicação em contextos contrastantes.

A geração de dados no cenário aleatório é auxiliada pela biblioteca *Faker*<sup>20</sup>, utilizada para criar valores credíveis e variados para simular empresas, métricas e valores ESG. Esta abordagem facilita a demonstração da plataforma e permite observar os efeitos de diferentes perfis de dados no *dashboard*.

A Figura 5.13 demonstra a pontuação ESG resultante da execução do *script seedGood*.

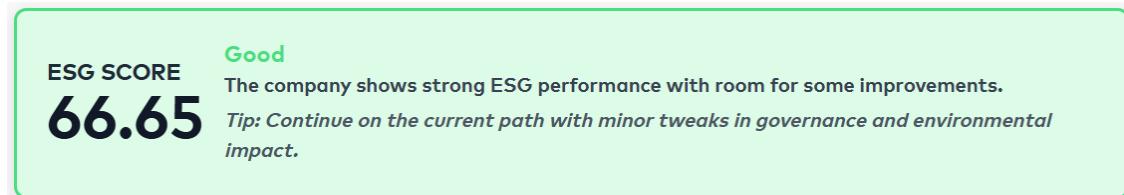


Figura 5.13: Pontuação ESG segundo o *script seedGood* (autoria própria)

A Figura 5.14 demonstra a pontuação ESG resultante da execução do *script seedBad*.



Figura 5.14: Pontuação ESG segundo o *script seedBad* (autoria própria)

A pontuação ESG resultante da execução do *script SeedRandom* varia significativamente, de forma a simular empresas com diferentes níveis de desempenho em cada um dos pilares ESG. Cada valor é atribuído aleatoriamente dentro de um intervalo entre 0 e 100, sendo a pontuação final posteriormente classificada conforme as categorias predefinidas, previamente indicadas na Tabela 5.1.

## 5.4 Testes

O objetivo dos testes de *software* é garantir a qualidade do produto, confirmando a sua funcionalidade, desempenho e conformidade com os requisitos previamente estabelecidos (softdesing 2025). Estes foram realizados com o auxílio do *Jest*<sup>21</sup> e da *React Testing Library*<sup>22</sup>. Os testes foram compilados num portátil com o *Windows 11* instalado, um processador *AMD Ryzen 7 6800HS* com *Radeon Graphics* e 16 gigabytes de memória RAM.

Para garantir a robustez e o correto funcionamento da solução desenvolvida, foram concebidos testes de várias tipologias. Foram utilizados testes unitários para garantir que cada

<sup>20</sup>Site oficial do Faker: <https://fakerjs.dev/>

<sup>21</sup>Site oficial do Jest: <https://jestjs.io/>

<sup>22</sup>Site oficial da React Testing Library: <https://testing-library.com/docs/react-testing-library/intro/>

componente isolado se comportava como esperado e que as suas funções estavam funcionais. A Listagem 5.12 refere-se a um teste unitário feito ao componente `SwiperControls`, na página inicial da aplicação, que verifica o comportamento do componente mediante um cenário onde não existem KPIs válidos.

```

1 it('renders a message when no valid KPIs exist', () => {
2   const noHistoryMetric: MetricInfoCard = {
3     ...baseMetric,
4     valueHistories: []
5   }
6   render(<SwiperControls kpis={[noHistoryMetric]} />)
7   expect(screen.getByText(/no metrics with sufficient
8     history/i)).toBeInTheDocument()
})
```

Listagem 5.12: Teste unitário ao componente `SwiperControls`

Paralelamente, foram realizados testes de integração com o objetivo de verificar a comunicação e interação entre os vários componentes e confirmar que estes funcionam de forma harmoniosa e consistente quando combinados. Um teste de integração entre os componentes `ChartErrorBoundary` e `ChartRenderer` é apresentado na Listagem 5.13. De forma a evitar que a aplicação falhe completamente, o componente `ChartErrorBoundary` é responsável por capturar os erros ocorridos durante a renderização dos gráficos no seu componente de ficheiro, neste caso o `ChartRenderer`. Adicionalmente, assegura o registo das mensagens de erro (*logging*), permitindo uma melhor monitorização e diagnóstico do problema.

```

1 describe('ChartErrorBoundary + ChartRenderer', () => {
2   it('renders fallback UI on invalid dataset', () => {
3     const invalidDataset = {
4       headers: ['value'],
5       rows: [{ value: 42 }, { value: 12 }]
6     };
7     render(
8       <ChartErrorBoundary>
9         <ChartRenderer
10           chartType="line"
11           dataset={invalidDataset}
12           datasetType="Time Series"
13           title="Invalid Test Chart"
14           unit="kg"
15         />
16       </ChartErrorBoundary>
17     );
18     expect(screen.getByText(/missing required date field/i)).toBeInTheDocument();
19   });
20});
```

Listagem 5.13: Teste de integração entre `ChartErrorBoundary` e `ChartRenderer`

Esta abordagem diversificada leva à criação de um elevado número de testes, Figura 5.15, permite identificar falhas a vários níveis do sistema, o que contribui para melhorar a qualidade contínua e a qualidade do produto final.

```

Test Suites: 25 passed, 25 total
Tests:       137 passed, 137 total
Snapshots:  6 passed, 6 total
Time:        9.857 s, estimated 40 s
Ran all test suites.
```

Figura 5.15: Testes desenvolvidos para a solução (autoria própria)

A Figura 5.16 ilustra o resumo do relatório de cobertura dos testes desenvolvidos.

File	%Stmts	%Branch	%Funcs	%Lines
All files	83.2	63.25	79.34	84.1

Figura 5.16: Resumo do relatório de cobertura dos testes desenvolvidos da solução (autoria própria)

Estes valores mostram um grau razoável de testes automatizados, especialmente na execução de instruções e cobertura de linhas, com alguma margem para melhoria na cobertura de ramos, para ajudar a garantir que todas as vias lógicas são adequadamente testadas.

## 5.5 Avaliação da solução

A aplicação apresenta tempos de carregamento eficientes, não ultrapassando os 30 segundos por página, como se pode ver na Figura 5.17, o que cumpre os critérios definidos para o projeto.

```

✓ Compiled /materiality-map in 1162ms
GET /materiality-map 200 in 1312ms
GET /api-pillars 200 in 680ms
GET /api/datasets 200 in 1096ms
GET /api/metrics 200 in 1557ms
GET /api-pillars 200 in 1013ms
GET /api/metrics 200 in 2291ms
o Compiling /objectives ...
✓ Compiled /objectives in 3.5s
GET /objectives 200 in 3643ms
o Compiling /api/objectives ...
✓ Compiled /api/objectives in 508ms
GET /api-objectives 200 in 2322ms
GET /api-objectives 200 in 1264ms
GET /api-metrics 200 in 1679ms
GET /api-metrics 200 in 1476ms
o Compiling /data ...
✓ Compiled /data in 1869ms

```

Figura 5.17: Compilação das páginas em menos de 30 segundos (autoria própria)

Para verificar se o tempo total de carregamento da solução idealizada, incluindo tanto o processo de *seeding* da base de dados quanto a compilação da página inicial, estava dentro do intervalo de 3 minutos, foi criado um *script* especial, como pode ser observado na Listagem 5.14.

```

1 #!/bin/bash
2
3 echo "Seeding and setup started at: $(date '+%Y-%m-%d %H:%M:%S')"
4 START=$(date +%s)
5
6 npm run db:soft-reset
7 npm run seed:good
8
9 END=$(date +%s)
10 echo "Seeding and setup ended at: $(date '+%Y-%m-%d %H:%M:%S')"
11 echo "Duration: $((END - START)) seconds"
12
13 echo "Now starting the dev server..."
14 npm run dev

```

Listagem 5.14: Script de avaliação de performance da iniciação da aplicação desenvolvida

A Figura 5.18 mostra o resultado obtido com a execução deste *script*, confirmando o cumprimento do requisito avaliado e evidenciando um tempo de compilação inicial de 49,5 segundos.

```

$ ./start-good-soft.sh
Seeding and setup started at: 2025-06-29 19:24:13

> esg-platform@4.0.0 db:soft-reset
> cross-env tsx prisma/clearDatabase.ts

  ✓ Clearing database...
  ✓ Database cleared!

> esg-platform@4.0.0 seed:good
> cross-env SEED_SCENARIO=good tsx prisma/seed.ts

  Seeding good scenario...
  ✓ Seeded excellent ESG data with very high scoring metrics.
  Seeding and setup ended at: 2025-06-29 19:24:50
  Duration: 37 seconds
  Now starting the dev server...

> esg-platform@4.0.0 dev
> next dev --turbopack

  ▲ Next.js 15.1.2 (Turbopack)
    - Local:      http://localhost:3000
    - Network:    http://192.168.165.1:3000
    - Environments: .env

  ✓ Starting...
  ✓ Ready in 2.4s
  ○ Compiling /home ...
  ✓ Compiled /home in 12.5s

```

Figura 5.18: Execução do *script* de avaliação do tempo de compilação inicial  
(autoria própria)

As animações utilizadas são fornecidas pelos componentes nativos do *React* e pelo template *Vuexy*, para uma experiência visual contínua e suave, sem sacrificar o desempenho da aplicação.

O *design* da aplicação é de fácil utilização e foi elogiado por várias partes interessadas, tais como o supervisor de estágio, o *buddy* (colaborador responsável pelo acompanhamento do estagiário na empresa), a equipa de produtos DevScope, que também analisou o protótipo

apresentado no Apêndice C, bem como colegas da universidade e do estágio. A navegação foi clara e intuitiva, facilitando a interação dos utilizadores com o sistema.

Os componentes implementados atendem ao esperado, sem apresentar erros imprevistos ao serem utilizados. O código é organizado em módulos, facilitando alterações e manutenções futuras, fruto não apenas da arquitetura tradicional das aplicações *React*, mas também da aplicação estrita de boas práticas de programação.

No que respeita à acessibilidade, a aplicação tem a língua inglesa e incorpora as cores institucionais da organização. A aplicação destina-se apenas a ser utilizada em computadores, posicionando-se como uma área com espaço para melhorias no futuro.

Foram efetuados testes unitários e testes de integração para garantir a estabilidade e a qualidade do sistema disponível. Além disso, a usabilidade foi testada com a participação de colegas, o que ajudou a revelar possíveis melhorias na experiência do utilizador. Os conjuntos de dados aplicados são imaginários, numa tentativa de manter a confidencialidade da empresa no que respeita às questões ESG abrangidas. A cobertura dos testes é considerada satisfatória, apesar de haver alguma margem para melhorias, nomeadamente em termos de ramos de código.

## 5.6 Implementação Alternativa

Durante o processo de desenvolvimento da aplicação, foi tida em conta a utilização de ferramentas de terceiros para complementar ou substituir, em determinadas áreas, componentes da solução desenvolvida. Especificamente, o *Microsoft Power BI* surgiu como uma alternativa que poderia trazer vários benefícios, nomeadamente na visualização de dados, criação de relatórios interactivos e gestão do esquema de dados de forma dinâmica e centralizada. Esta abordagem permitiria que os utilizadores finais pudessem analisar os dados ESG de uma forma mais detalhada e visualmente apelativa, tirando partido de um ecossistema maduro e popular no contexto empresarial.

Esta hipótese ganhou um significado ainda maior à luz do acordo de parceria entre a DevScope e a *Microsoft*, que facilitaria a integração do *Power BI* na estrutura da solução, tanto estratégica como tecnicamente.

No entanto, na sequência de uma análise interna e de uma consulta à equipa de produto, concluiu-se que, relativamente aos objetivos estabelecidos para o projeto, essa abordagem teria inconvenientes consideráveis. A principal consideração a favor de uma solução personalizada foi o modelo de negócio pretendido: a criação de um produto de venda única, independente de subscrições ou licenciamentos contínuos de terceiros. A utilização do *Power BI* obrigaria a que os utilizadores da solução estivessem associados a uma subscrição ativa do serviço por parte da *Microsoft*, o que poderia ser um inibidor de adoção e comprometer a independência da organização relativamente ao produto final. Desta forma, foi determinada a implementação de uma aplicação autónoma, que fornece visualizações personalizadas e permite o controlo total sobre todos os elementos da experiência do utilizador, alinhando-se com os objetivos de negócio da DevScope, bem como com as melhores práticas no desenvolvimento de soluções de negócio escaláveis a nível empresarial.



# Capítulo 6

## Conclusões

Neste último Capítulo, é apresentada uma síntese do trabalho efetuado, bem como os resultados obtidos, que sustentam as conclusões finais. Os objetivos definidos para o projeto são reafirmados e é discutida a forma como os resultados os cumprem. De seguida, são apontadas as principais limitações encontradas, bem como recomendações para futuras extensões e melhorias da solução desenvolvida, seguindo-se uma análise global do projeto.

### 6.1 Objetivos concretizados

Tal como referido no Capítulo 1, os objetivos especificados e o estado de concretização desses objetivos estão resumidos na Tabela 6.1:

Objetivo	Estado
<b>Consolidar dados de múltiplas fontes</b>	Realizado
<b>Desenvolver uma aplicação configurável</b>	Realizado
<b>Visualização clara e interactiva dos dados ESG</b>	Realizado
<b>Definir e monitorizar objetivos baseados em métricas</b>	Realizado

Tabela 6.1: Objetivos do Projeto e o seu Estado de Concretização (autoria própria)

Todos os objetivos estabelecidos foram plenamente atingidos. A solução criada permite a centralização da informação ESG, a definição de objetivos baseados em metas ligadas a métricas (sejam estas pré-existentes ou personalizadas), tem uma interface adaptável e oferece uma visualização gráfica visível e interactiva dos dados.

A planificação cuidadosa e análise prévia permitiram uma implementação eficaz e livre de complicações, garantindo a realização dos objetivos definidos.

### 6.2 Trabalho Futuro e Limitações

Apesar do facto de a solução proposta ser eficiente e completa, foram descobertas algumas limitações ao longo do processo de desenvolvimento. A funcionalidade do caso de uso UC-07 (Importação de Dados) tem limitações nos formatos de ficheiros suportados. De momento, apenas são suportados ficheiros CSV com uma estrutura predefinida.

Como consequência da limitação acima mencionada, a representação dos diagramas está igualmente limitada, pois a biblioteca usada, o *ApexCharts* disponibiliza 20 diagramas diferentes, enquanto que a plataforma apenas usufrui de 5.

Por último, apesar de dispor de testes unitários e de integração, o projeto beneficiaria de uma *pipeline* de integração contínua (CI/CD) mais robusta, como a execução automática de testes e implementações. Esta melhoria permitiria uma simulação mais realista do ambiente de produção e alinharia a prática de desenvolvimento com as práticas modernas de *DevOps*.

Como trabalho futuro, destaca-se a possibilidade de:

- Aumentar ou melhorar a importação para aceitar vários tipos de ficheiros e estruturas
- Explorar todas as categorias de gráficos disponíveis na biblioteca *ApexCharts*
- Facilitar a intuitividade do processo de edição dos dados que foram importados
- Implementar a criação automática de relatório com fluxos de recolha, tratamento e apresentação de dados
- Implementar um mecanismo de autenticação para limitar o acesso à plataforma apenas a utilizadores autorizados
- Impôr uma *pipeline* de integração contínua (CI/CD), com a execução automática de testes e implementações

Estas futuras extensões acrescentariam um valor significativo à solução e ao posicionamento da empresa no domínio ESG.

## 6.3 Considerações Finais

A presente Secção divide-se em duas partes: uma perspetiva crítica do problema apresentado pela empresa e das implicações no desenvolvimento da solução, e, por fim, considerações pessoais perante a experiência de estágio e o trabalho desenvolvido.

### 6.3.1 Perspetiva Crítica do Problema definido e Solução proposta

O projeto permitiu a utilização das capacidades adquiridas através da formação académica, desenvolvendo simultaneamente novas competências através do desenvolvimento prático e de vários *workshops* promovidos pela DevScope.

A proposta inicialmente apresentada foi a criação de uma plataforma de apoio à avaliação ESG, com ênfase na vertente técnica de implementação e integração das respectivas funcionalidades, incluindo o acompanhamento e gestão das medidas.

Porém, caso se trate de um projeto de produção, os objetivos estratégicos poderiam ter sido formulados de forma mais clara em termos do perfil do utilizador final, dos fluxos de trabalho pretendidos e dos requisitos organizacionais reais em termos de monitorização ESG. Uma melhor articulação inicial com a organização teria permitido um alinhamento mais precoce no foco da aplicação com as suas prioridades operacionais e tecnológicas.

Durante o desenvolvimento, a utilização do *React.js* para o *frontend*, em conjunto com o *template Vuexy*, revelou-se eficaz para permitir a rápida criação de interfaces modernas. No entanto, em ambientes de produção, uma abordagem mais modular teria sido benéfica para agilizar a manutenção futura e permitir a integração de novas funcionalidades.

A base de dados relacional *MySQL* foi suficiente para o desenvolvimento da prova de conceito, mas para uma maior flexibilidade, particularmente com a diversidade das práticas ESG

entre setores e empresas, sistemas como *PostgreSQL* (com suporte a campos dinâmicos) ou soluções *NoSQL* poderiam ser mais adequados.

No caso das infraestruturas, a utilização do *AWS S3* facilitou o desenvolvimento local e prototipagem rápida. Apesar de esta solução ter sido adequada, numa implementação eficiente, a utilização do *Azure Blob Storage* teria permitido uma integração mais natural com o ecossistema tecnológico já existente na empresa, facilitando futuras configurações e manutenções.

Em relação à fundamentação teórica, a comparação das principais *frameworks ESG*, GRI e SASB, permitiu uma decisão acertada na adoção do SASB, que já é utilizado pela empresa. No entanto, a plataforma poderia ter sido concebida desde o início com mais abstração para suportar várias *frameworks*, melhorando assim a escalabilidade e a reutilização em vários contextos empresariais.

Uma comparação de soluções empresariais como *Workiva*, *IBM ESG Suite* e *SAP Sustainability Control Tower* contribuiu para a definição funcional da aplicação. Uma investigação técnica mais aprofundada das suas API, fluxos de dados e práticas de integração teria enriquecido ainda mais a arquitetura, nomeadamente na forma como os objectivos são geridos.

Em suma, a aplicação desenvolvida atingiu todos os objectivos para esta fase de prova de conceito. Melhorias estratégicas e tecnológicas adicionais, alinhadas com o contexto organizacional e as exigências de escalabilidade, poderiam também ajudar na sua manutenção e evolução a longo prazo.

### 6.3.2 Considerações Pessoais

O planeamento e desenvolvimento do projeto, no âmbito profissional da DevScope, foram extremamente proveitosos. O ambiente positivo de troca de conhecimento, apoio e consistência contribuiu significativamente para a efetiva realização e implementação bem sucedida das funcionalidades propostas.

A nível pessoal, a experiência foi especialmente gratificante. O tema do projeto desperta um grande interesse por parte do autor, o que aumentou o envolvimento. Além disso, o desafio foi uma oportunidade de resiliência e crescimento, incluindo interações com profissionais experientes do setor.

A empresa forneceu um estímulo constante e *feedback* pertinente, inclusive na redação do relatório.

Para concluir, o projeto foi uma experiência muito enriquecedora, tanto do ponto de vista técnico como humano, que abriu portas para futuras possibilidades profissionais e reforçou competências básicas para a integração no mercado de trabalho.



# Bibliografia

- Adams, C et al. (2022). «The development and implementation of GRI Standards: practice and policy issues». Em: *Handbook of Accounting and Sustainability* (3), pp. 3–6. url: <http://go.qub.ac.uk/oa-feedback>.
- Angular (2025). *What is Angular? • Angular*. url: <https://angular.dev/overview> (acedido em 04/07/2025).
- Antolín-López, Raquel. e Natalia. Ortiz-de-Mandojana (2023). *Measuring and disclosing Environmental, Social and Governance (ESG) information and performance*. Rel. téc. European Commision. doi: 10.2760/864272. url: <https://publications.jrc.ec.europa.eu/repository/handle/JRC131932> (acedido em 04/04/2025).
- Asif, Muhammad, Cory Searcy e Pavel Castka (out. de 2023). «ESG and Industry 5.0: The role of technologies in enhancing ESG disclosure». Em: *Technological Forecasting and Social Change* 195 (23), pp. 1–12. issn: 0040-1625. doi: 10.1016/J.TECHFORE.2023.122806.
- Aydoğmuş, Mahmut, Güzhan Gülay e Korkmaz Ergun (dez. de 2022). «Impact of ESG performance on firm value and profitability». Em: *Borsa İstanbul Review* 22, pp. 2–8. issn: 2214-8450. doi: 10.1016/J.BIR.2022.11.006.
- Berg, Florian, Julian F Kölbel e Roberto Rigobon (nov. de 2022). «Aggregate Confusion: The Divergence of ESG Ratings». Em: *Review of Finance* 26 (6), pp. 1315–1344. issn: 1572-3097. doi: 10.1093/ROF/RFAC033. url: <https://dx.doi.org/10.1093/rof/rfac033>.
- Brown, Simon (2025). *Home / C4 model*. url: <https://c4model.com/> (acedido em 06/06/2025).
- Brown, Simon e Thomas Betts (2018). *The C4 Model for Software Architecture - InfoQ*. url: <https://www.infoq.com/articles/C4-architecture-model> (acedido em 31/05/2025).
- Burnaev, Evgeny et al. (ago. de 2023). «Practical AI Cases for Solving ESG Challenges». Em: *Sustainability* 2023, Vol. 15, Page 12731 15 (17), pp. 1–15. issn: 2071-1050. doi: 10.3390/SU151712731. url: <https://www.mdpi.com/2071-1050/15/17/12731/htm> <https://www.mdpi.com/2071-1050/15/17/12731>.
- Carlson, Debbie (2021). *Mentions of 'ESG' and Sustainability are Being Made on Thousands of Corporate Earnings Calls*. url: <https://www.marketwatch.com/story/mentions-of-esg-and-sustainability-are-being-made-on-thousands-of-corporate-earnings-calls-11626712848> (acedido em 25/03/2025).
- code-examples.net (2025). *database - How to Efficiently Store and Retrieve Images in Your Web Applications*. url: <https://code-examples.net/en/q/ea4> (acedido em 29/06/2025).
- Cohen, D., M. Lindvall e P. Costa (2004). «An Introduction to Agile Methods». Em: *Advances in Computers* 62, pp. 14–16. doi: 10.1016/S0065-2458(03)62001-2.
- Cohen, Gil (2023). «The impact of ESG risks on corporate value». Em: *Review of Quantitative Finance and Accounting* 60, pp. 1451–1468. doi: 10.1007/s11156-023-01135-6. url: <https://doi.org/10.1007/s11156-023-01135-6>.

- Cruz, Carolina Almeida e Florinda Matos (fev. de 2023). «ESG Maturity: A Software Framework for the Challenges of ESG Data in Investments». Em: *Sustainability 2023, Vol. 15, Page 2610* 15 (3), pp. 2–10. issn: 2071-1050. doi: 10.3390/SU15032610. url: <https://www.mdpi.com/2071-1050/15/3/2610/htm%20https://www.mdpi.com/2071-1050/15/3/2610>.
- DevScope (2025). *Solutions*. url: <https://devscope.net/solutions/> (acedido em 17/03/2025).
- F5 (2025). *What Is a Load Balancer? / F5*. url: <https://www.f5.com/glossary/load-balancer> (acedido em 30/04/2025).
- Fang, Mingyue, Huihua Nie e Xinyi Shen (jan. de 2023). «Can enterprise digitization improve ESG performance?» Em: *Economic Modelling* 118 (22), pp. 1–15. issn: 0264-9993. doi: 10.1016/J.ECONMOD.2022.106101.
- Goswami, Kuntal, Mohammed Kazi Saidul Islam e Winton Evers (2023). «A Case Study on the Blended Reporting Phenomenon: A Comparative Analysis of Voluntary Reporting Frameworks and Standards—GRI, IR, SASB, and CDP». Em: *International Journal of Sustainability Policy and Practice* 19 (2), pp. 35–64. issn: 23251182. doi: 10.18848/2325-1166/CGP/V19I02/35-64. url: [https://www.researchgate.net/publication/373122132\\_A\\_Case\\_Study\\_on\\_the\\_Blended\\_Reportin...](https://www.researchgate.net/publication/373122132_A_Case_Study_on_the_Blended_Reportin...)
- GRI (2025a). *GRI - GRI Standards English Language*. url: <https://www.globalreporting.org/how-to-use-the-gri-standards/gri-standards-english-language/> (acedido em 04/04/2025).
- (2025b). *GRI - Sector Program*. url: <https://www.globalreporting.org/standards/sector-program/> (acedido em 04/04/2025).
- GRI e SASB (2021). *A Practical Guide to Sustainability Reporting Using GRI and SASB Standards PARTIES INVOLVED*. Rel. téc. Global Reporting Initiative e Sustainability Accounting Standards Board, pp. 5–37. (Acedido em 28/03/2025).
- IBM (2025). *Environmental Intelligence / IBM*. url: <https://www.ibm.com/products/environmental-intelligence> (acedido em 04/04/2025).
- Kim, Soohun e Aaron S Yoon (abr. de 2022). «Analyzing Active Fund Managers' Commitment to ESG: Evidence from the United Nations Principles for Responsible Investment». Em: *Management Science* 69 (2), pp. 7–9. issn: 1526-5501. url: <https://doi.org/10.1287/mnsc.2022.4394>.
- Kruchten, Philippe B. (1995). «The 4+1 View Model of Architecture». Em: *IEEE Software* 12 (6), pp. 42–50. issn: 07407459. doi: 10.1109/52.469759. url: [https://www.researchgate.net/publication/220018231\\_The\\_41\\_View\\_Model\\_of\\_Architecture](https://www.researchgate.net/publication/220018231_The_41_View_Model_of_Architecture).
- Krueger, Philipp et al. (dez. de 2024). «The Effects of Mandatory ESG Disclosure Around the World». Em: *Journal of Accounting Research* 62 (5), pp. 1795–1847. issn: 1475-679X. doi: 10.1111/1475-679X.12548. url: <https://onlinelibrary.wiley.com/doi/full/10.1111/1475-679X.12548%20https://onlinelibrary.wiley.com/doi/abs/10.1111/1475-679X.12548%20https://onlinelibrary.wiley.com/doi/10.1111/1475-679X.12548>.
- Lano, Kevin e Sobhan Yassipour Tehrani (out. de 2023). *Introduction to Software Architecture: Innovative Design using Clean Architecture and Model-Driven Engineering*. Also available under ISBN 3031441435. url: [https://books.google.pt/books?hl=en&lr=&id=g03aEAAAQBAJ&oi=fnd&pg=PA2&dq=clean+architecture+software&ots=sVLMwINN2n&sig=N0anb1HBC8hsEHrwWLQp6bgrPvo&redir\\_esc=y#v=onepage&q=clean%20architecture%20software&f=false](https://books.google.pt/books?hl=en&lr=&id=g03aEAAAQBAJ&oi=fnd&pg=PA2&dq=clean+architecture+software&ots=sVLMwINN2n&sig=N0anb1HBC8hsEHrwWLQp6bgrPvo&redir_esc=y#v=onepage&q=clean%20architecture%20software&f=false) (acedido em 09/06/2025).

- Luque-Vilchez, Mercedes et al. (jul. de 2023). «Key aspects of sustainability reporting quality and the future of GRI». Em: *Sustainability Accounting, Management and Policy Journal* 14 (4), pp. 637–659. issn: 2040803X. doi: 10.1108/SAMPJ-03-2023-0127/FULL/PDF.
- Martin, Robert C. (ago. de 2012). *Clean Coder Blog*. url: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (acedido em 09/06/2025).
- Mastery, Vue (2025). *Vue Mastery*. url: <https://www.vuemastery.com/guides/vue-3-essentials/what-is-vue/> (acedido em 04/07/2025).
- ONU (2025). *What is the UN Global Compact / UN Global Compact*. url: <https://unglobalcompact.org/what-is-gc> (acedido em 31/03/2025).
- onValues (out. de 2005). *Investing for Long-Term Value Integrating environmental, social and governance value drivers in asset management and financial research-A state-of-the-art assessment*. Rel. téc. International Finance Corporation, pp. 8–10. (Acedido em 07/04/2025).
- Patel, N. (2025). *Metodologia ágil: O que é, ferramentas e como aplicar*. url: <https://neilpatel.com/br/blog/metodologia-agil/> (acedido em 21/03/2025).
- PixInvent (2025). *Vuexy - Bootstrap HTML Admin Template - PixInvent*. url: <https://pixinvent.com/vuexy-bootstrap-html-admin-template/> (acedido em 30/04/2025).
- Pizzi, Simone, Salvatore Principale e Elbano de Nuccio (nov. de 2023). «Material sustainability information and reporting standards. Exploring the differences between GRI and SASB». Em: *Meditari Accountancy Research* 31 (6), pp. 1654–1674. issn: 20493738. doi: 10.1108/MEDAR-11-2021-1486/FULL/XML.
- Pollman, Elizabeth (2024). «THE MAKING AND MEANING OF ESG». Em: *Harvard Business Law Review* 14 (2), pp. 404–418. url: [https://journals.law.harvard.edu/hblr/wp-content/uploads/sites/87/2024/10/04\\_HLB\\_14\\_2\\_Elizabeth-Pollman.pdf](https://journals.law.harvard.edu/hblr/wp-content/uploads/sites/87/2024/10/04_HLB_14_2_Elizabeth-Pollman.pdf).
- PRI (2017). *A BLUEPRINT FOR RESPONSIBLE INVESTMENT RESPONSIBLE INVESTORS SUSTAINABLE MARKETS A PROSPEROUS WORLD FOR ALL*. url: <https://www.unpri.org/about-us/a-blueprint-for-responsible-investment> (acedido em 31/03/2025).
- Rau, P. Raghavendra e Ting Yu (mar. de 2024). «A survey on ESG: investors, institutions and firms». Em: *China Finance Review International* 14 (1), pp. 3–33. issn: 20441401. doi: 10.1108/CFRI-12-2022-0260.
- React (2025). *React*. url: <https://react.dev/> (acedido em 04/04/2025).
- SAP (2025). *Características do SAP Sustainability Control Tower*. url: <https://www.sap.com/portugal/products/scm/sustainability-control-tower/features.html> (acedido em 04/04/2025).
- SASB (2025a). *About us - SASB*. url: <https://sasb.ifrs.org/about/> (acedido em 04/04/2025).
- (2025b). *Find your industry - SASB*. url: <https://sasb.ifrs.org/find-your-industry/> (acedido em 30/04/2025).
- Schiemann, Frank e Raphael Tietmeyer (nov. de 2022). «ESG Controversies, ESG Disclosure and Analyst Forecast Accuracy». Em: *International Review of Financial Analysis* 84 (37), pp. 102–373. issn: 1057-5219. doi: 10.1016/J.IRFA.2022.102373.
- Schwarzmueller, Maximilian (dez. de 2022). *React Key Concepts: Consolidate your knowledge of React's core features*. Packt Publishing Ltd, pp. 1–49. isbn: 9781803240480.
- softdesing (2025). *Processo de desenvolvimento de software: veja o passo a passo*. url: <https://softdesign.com.br/blog/processo-de-desenvolvimento-de-software/#h-como-escolher-o-tipo-de-arquitetura-certa> (acedido em 29/06/2025).

- Tsui, Frank, Orlando Karam e Barbara Bernal (2022). *Essentials of software engineering*. Jones & Bartlett Learning.
- UBIMINDS (2025). *Como o Ciclo de Vida do Desenvolvimento de Software funciona e quais ferramentas são necessárias em cada uma das suas fases?* - *Ubiminds*. url: <https://ubiminds.com/pt-br/como-o-ciclo-de-vida-de-desenvolvimento-de-software-funciona-e-quais-ferramentas-sao-necessarias-em-cada-uma-das-suas-fases/> (acedido em 25/06/2025).
- Vue (2025). *Introduction / Vue.js*. url: <https://vuejs.org/guide/introduction.html> (acedido em 04/07/2025).
- Vuexy (2025). *Choosing Between Starter Kit and Full Package / Vuexy - MUI Next.js Admin Dashboard Template*. url: <https://demos.pixinvent.com/vuexy-nextjs-admin-template/documentation/docs/guide/overview/starter-kit-vs-full-package> (acedido em 30/04/2025).
- Wakode, R. B., L. P. Raut e P. Talmale (fev. de 2015). «Overview on Kanban Methodology and its Implementation». Em: *International Journal of Engineering Research and Technology* 3.2, pp. 1–4.
- Wang, Ni et al. (out. de 2023). «How do ESG practices create value for businesses? Research review and prospects». Em: *Sustainability Accounting, Management and Policy Journal* 15 (5), pp. 1155–1177. issn: 2040-8021. doi: 10.1108/SAMPJ-12-2021-0515. url: <https://doi.org/10.1108/SAMPJ-12-2021-0515>.
- Whelan, Tensie et al. (2021). *ESG and Financial Performance: Uncovering the Relationship by Aggregating Evidence from 1,000 Plus Studies Published between 2015–2020*. Rel. téc. NYU Stern Center for Sustainable Business e Rockefeller Asset Management, pp. 3–4. url: [https://www.stern.nyu.edu/sites/default/files/assets/documents/NYU-RAM\\_ESG-Paper\\_2021.pdf](https://www.stern.nyu.edu/sites/default/files/assets/documents/NYU-RAM_ESG-Paper_2021.pdf) (acedido em 02/04/2025).
- Workiva (2025). *Workiva Platform / Integrated GRC, ESG & Financial Reporting*. url: <https://www.workiva.com/en-nl/platform> (acedido em 04/04/2025).

## Apêndice A

# Cronograma do Projeto de Estágio

O presente gráfico de Gantt apresenta o cronograma do projeto de estágio, detalhando as fases e prazos das atividades planeadas.

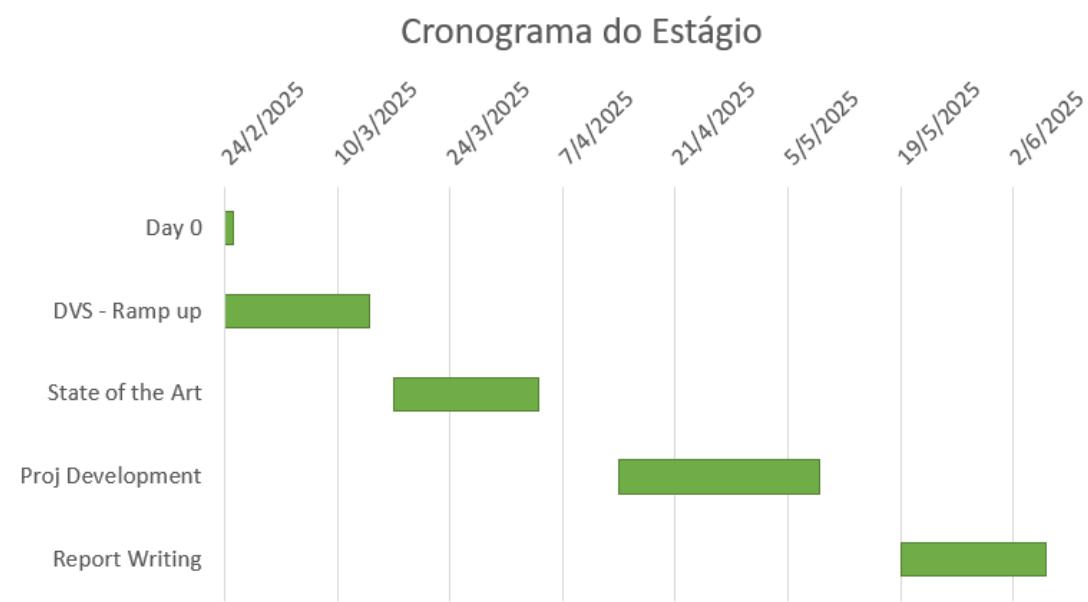


Figura A.1: Cronograma do projeto de estágio (autoria própria)



## Apêndice B

# Mapeamento entre vistas de Implementação e Lógica (Nível 3)

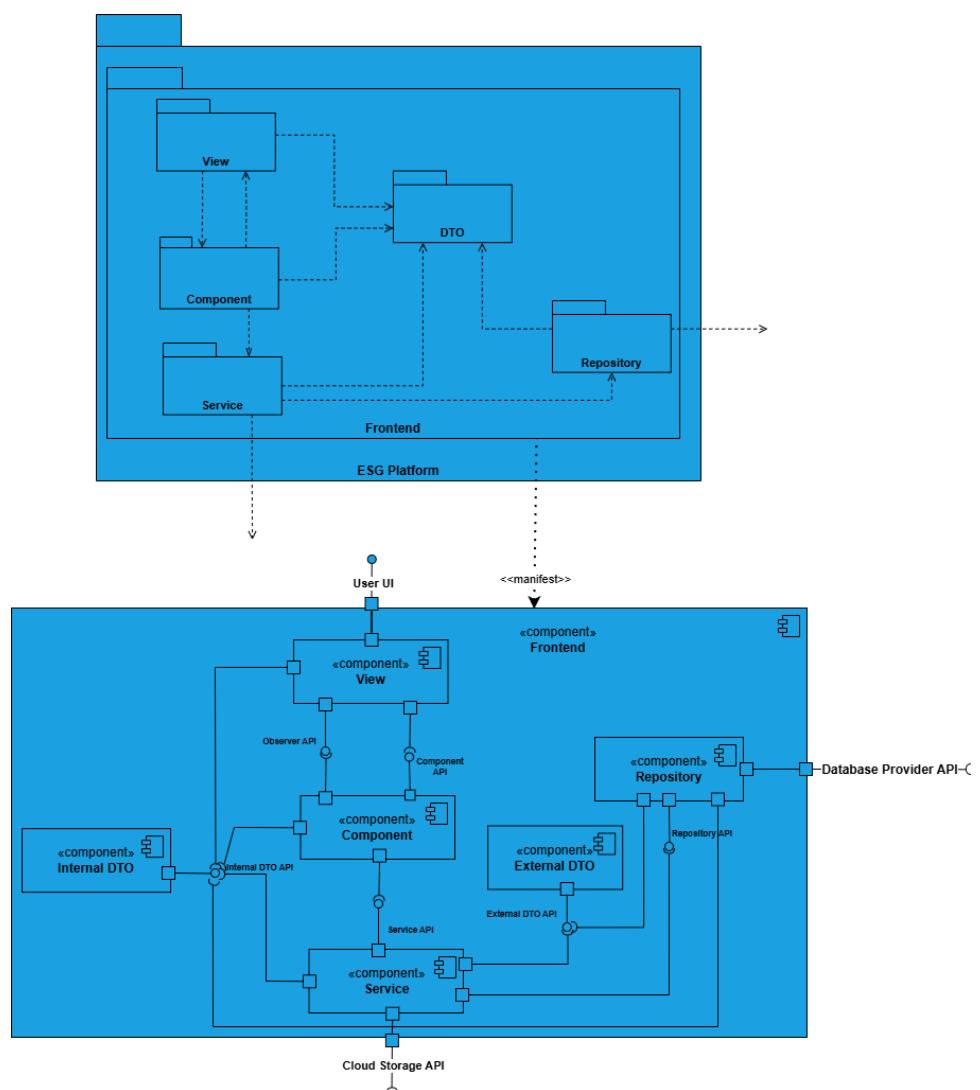


Figura B.1: Mapeamento entre vistas de Implementação e Lógica (Nível 3)  
(autoria própria)



## Apêndice C

# Designs do Figma

O seguinte Apêndice apresenta os *designs* do *mockup* desenvolvido na plataforma Figma para o projeto em questão. A criação deste protótipo surgiu da ênfase dada à área de *Frontend* durante o estágio, sendo a idealização da interface e a organização da informação elementos fundamentais para um desenvolvimento mais ágil, coerente e orientado.

O objetivo principal do *mockup* foi validar a arquitetura de informação, a experiência do utilizador e a adequação visual da plataforma às necessidades da empresa.

O protótipo serviu como base de referência para a implementação da interface, reduzindo ambiguidades e acelerando o processo de desenvolvimento ao fornecer uma visão clara das funcionalidades e do seu comportamento esperado. Este foi submetido à empresa, que o aprovou e forneceu um *feedback* positivo, validando as decisões de *design* tomadas.

As imagens seguintes ilustram algumas das páginas principais do protótipo, incluindo o painel de controlo, a personalização de métricas e a criação de objetivos.

The screenshot displays the main dashboard of the Dev<ESG>scope platform. At the top, there is a navigation bar with the title "Dev<ESG>scope" and a menu icon. Below the title, a sidebar on the left contains links to "Home", "Objectives", "Materiality Map", "Metrics", and "Data". The main content area features a large green banner titled "ESG Rating" with the status "Very High" and a score of "86". Below this, a section titled "Main KPI's" shows three metrics: "Carbon Emissions" (320 tons CO<sub>2</sub>, -20 tons from last month), "Employee Engagement Score" (85/100, +4 pts from last month), and "Board Gender Diversity" (40% women, +5% from last month). Further down, a section titled "Improving Metrics" highlights positive changes: "Carbon Emissions..." (Reduced environmental impact by lowering CO<sub>2</sub> output), "Training Hours per Employee..." (More professional development opportunities provided), and "Board Diversity..." (Higher representation of women in leadership). At the bottom left, there is a user profile for "Matilde Varela" with a small photo and three dots indicating more options. The overall design is clean and modern, using a color palette of green, orange, and white.

Figura C.1: Página Inicial (Dashboard) da Plataforma ESG (autoria própria)

**Dev<ESG>scope**

- Home
- Objectives
- Materiality Map
- Metrics**
- Data

All Environment Social Governance  Create Metric Edit Metrics

## Sustainable Development Goals (SDGs)

Tracking progress on key Sustainable Development Goals through measurable ESG-aligned indicators

**SDG 3**  
Good Health and Well-being  
Ensure healthy lives and promote well-being for all at...

**SDG 5**  
Gender Equality  
Achieve gender equality and empower all women and girls.

**SDG 7**  
Affordable and Clean Energy  
Ensure access to affordable, reliable, sustainable, and mo...

**SDG 13**  
Climate Action...  
Take urgent action to combat climate change and its impac...

...

## Custom Metrics

Custom metrics designed to reflect DevScope's unique impact, culture, and strategic goals

Tech Community Contributions  
Tracks number of open-source commits, talks, or events supp...

Client Solution Efficiency...  
Measures average performance improvement (...)

Innovation Projects Ratio...  
Percentage of R&D or innovation-focused projects...

Employee Skill Growth Index...  
Assesses training hours, certifications, and upskilling...

...

## SASB-Aligned Metrics

Key sustainability metrics based on SASB standards, tailored for transparency and accountability in the Software & IT Services sector

Figura C.2: Página das Métricas Customizáveis e de SASB (autoria própria)

## Metric Customization

[Exit](#)

Customize ESG tracking by adding your own tailored indicators

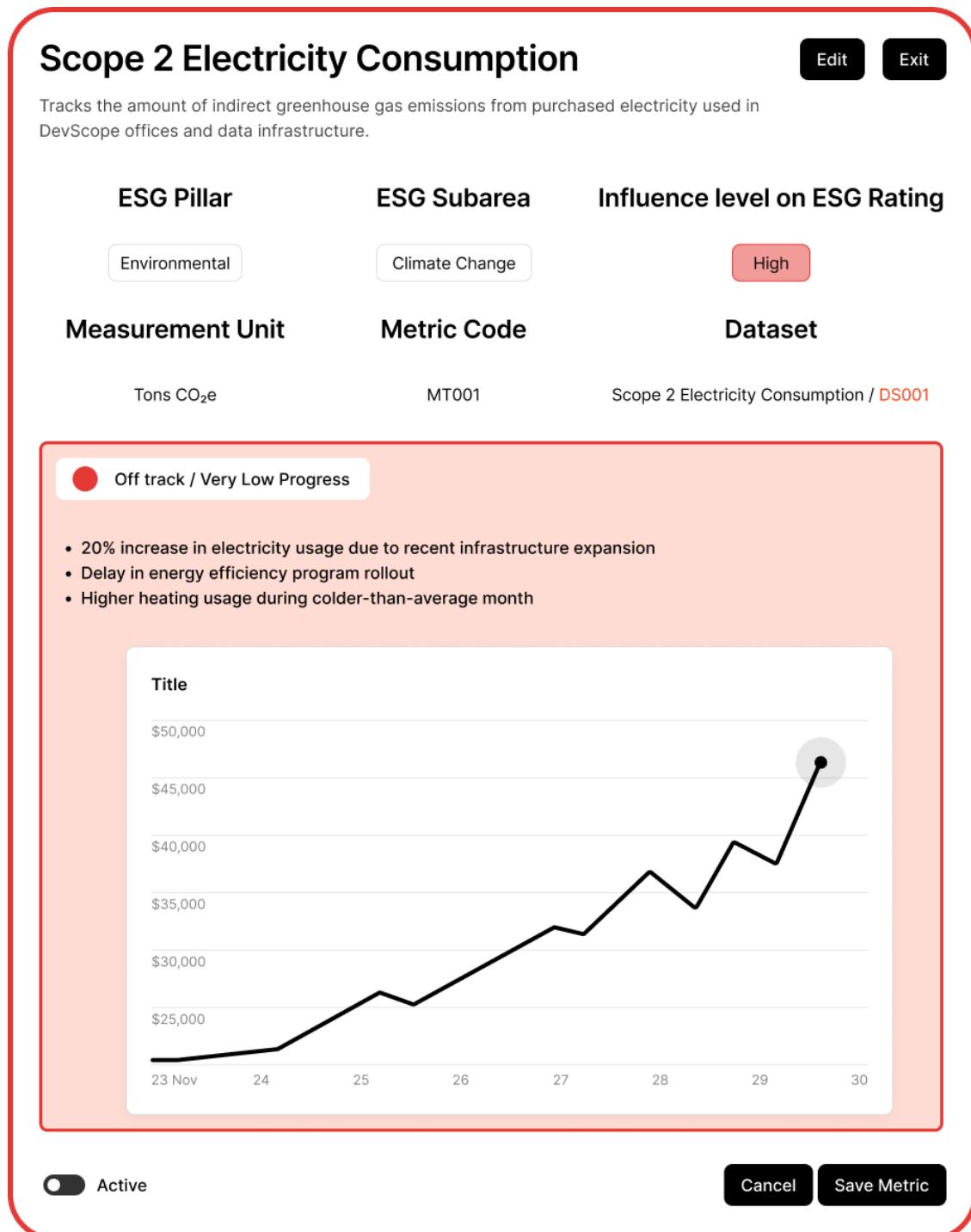
 Name Description ESG Pillar ESG Subarea Measurement Unit Level of Impact Dataset Reference Type of Diagram Metric Code (automatic)

### Metric banner

Add a rectangular image to visually represent your metric. Recommended size: 1200×400px.

[Cancel](#)[Save Metric](#)

Figura C.3: Painel de Criação de Métricas Customizáveis (autoria própria)



**Dataset Collection**

Code	Name	ESG Pillar	Type	Validity	Last Updated
DS001	Scope 2 Electricity Consumption	Environmental	Time Series	✓	Apr 1
DS002	GHG Emissions (Tons CO <sub>2</sub> e)	Environmental	Snapshot	✓	Mar 15
DS003	Employee Satisfaction Survey Results	Social	Survey	✓	Mar 30
DS004	Gender Representation in Tech Roles	Social	Time Series	!	Mar 20
DS005	Ethics & Compliance Training Completion	Governance	Time Series	✓	Apr 5
DS006	Energy Consumption per Server	Environmental	Time Series	✓	Mar 29
DS007	Data Breaches Report	Governance	Incident Log	✓	Mar 18
DS008	Diversity Hiring Funnel Metrics	Social	Time Series	!	Mar 25
DS009	Monthly Water Usage in Office Spaces	Environmental	Time Series	✓	Mar 31
DS010	Governance Policy Revision Tracker	Governance	Log File	✓	Apr 4

Matilde Varela ...

Figura C.5: Página dos Conjuntos de Dados (autoria própria)

**Dataset Import**

Import your datasets and define custom metrics for personalized tracking.

Name

Description

+ Import Dataset

Only CSV files using commas (,) as delimiters are supported for import

Measurement Unit

Dataset Type

ESG Pillar

Dataset Code (automatic)

Cancel Save Dataset

Figura C.6: Painel de Importação de Conjuntos de Dados (autoria própria)

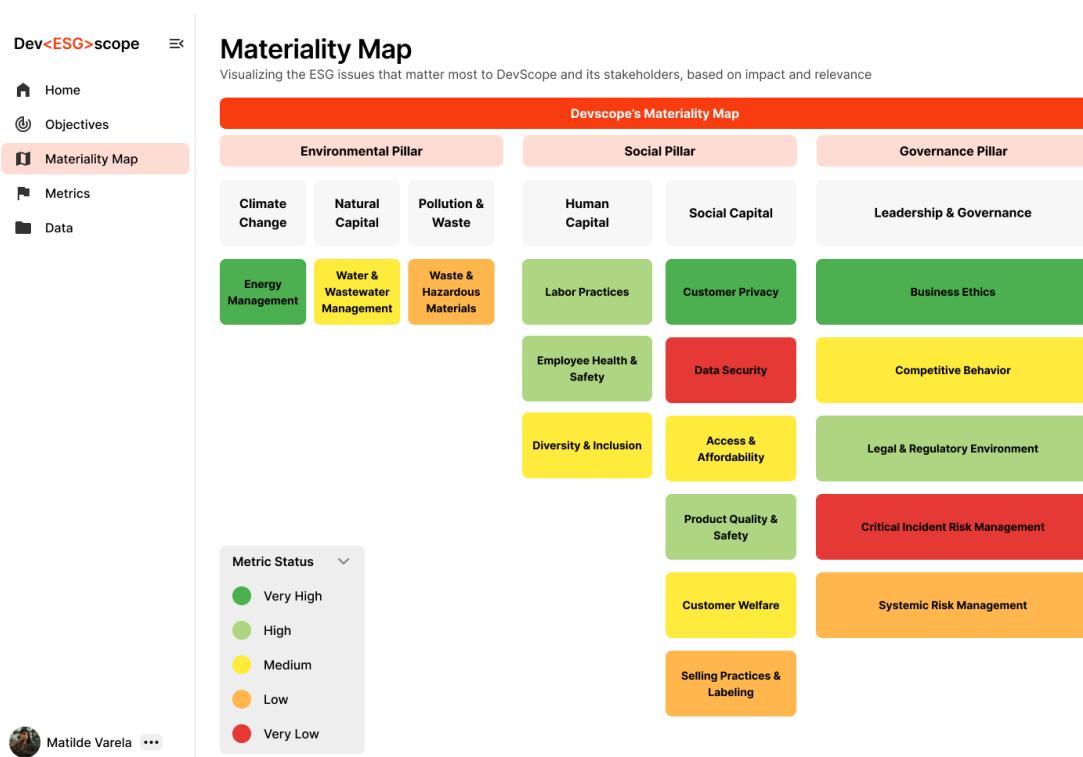


Figura C.7: Matriz de Materialidade segundo o setor de Software e Serviços IT da SASB (autoria própria)

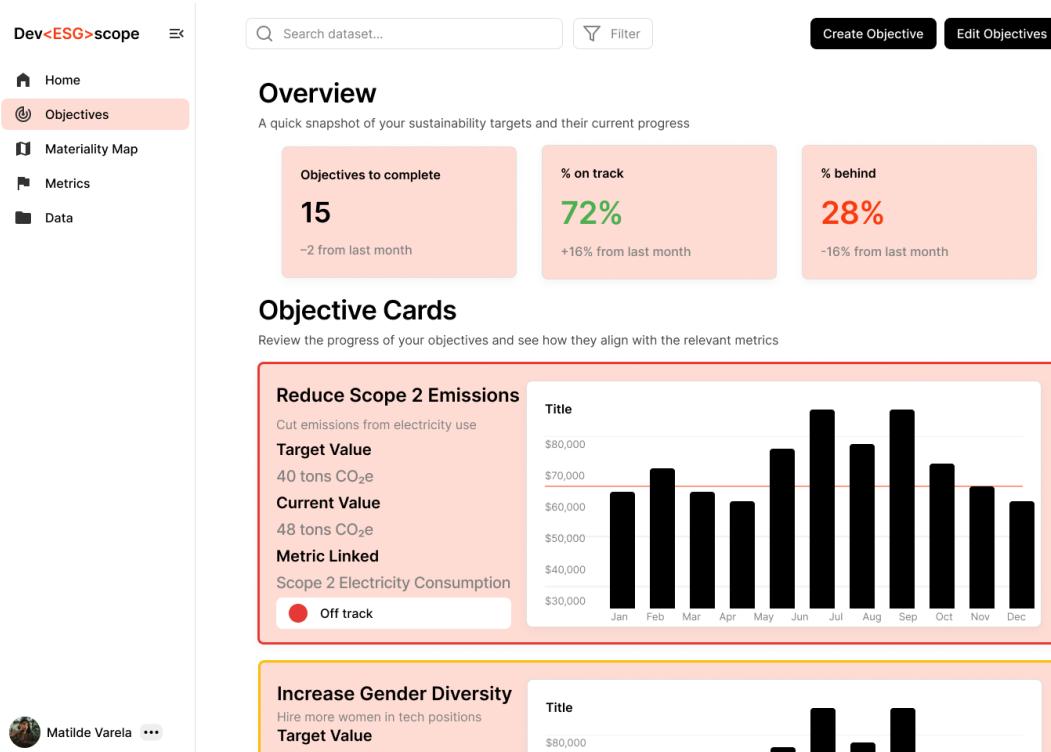


Figura C.8: Página dos Objetivos (autoria própria)

The screenshot shows a user interface for creating an objective. At the top right is a black button labeled "Exit". Below it is a sub-header "Set custom ESG objectives to benchmark against key metrics". The main form consists of several input fields:

- "Name" input field.
- "Description" input field.
- "ESG Pillar" dropdown menu.
- "Priority" dropdown menu.
- "Target value" input field.
- "Comparative Metric" dropdown menu.
- "Time Reference (Deadline or Timeframe)" dropdown menu.
- "Deadline" input field with a calendar icon.
- "Timeframe Beginning Date" input field with a calendar icon.
- "Timeframe End date" input field with a calendar icon.
- "Objective Code (automatic)" input field.

At the bottom right are two buttons: "Cancel" and "Save Objective".

Figura C.9: Painel de Criação de Objetivos (autoria própria)



## Apêndice D

# Quadro Kanban do Desenvolvimento da Solução no Github Projects

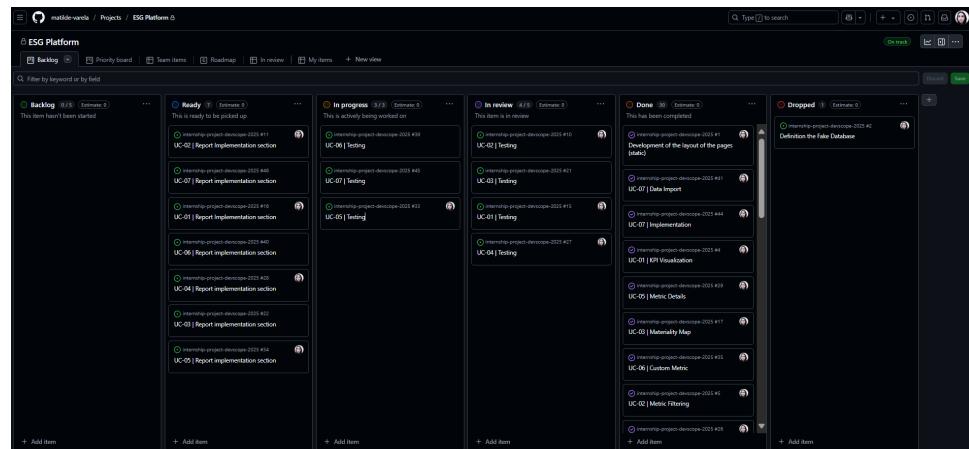


Figura D.1: Painel com quadro *kanban* usado no planeamento do projeto  
(autoria própria)