



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

DARIEL BEZERRA DE SOUSA

ANÁLISE, MEDIÇÃO E REDUÇÃO DO IMPACTO DA EMISSÃO DE CARBONO EM
SOFTWARE DE GRANDE PORTE: APLICAÇÃO PRÁTICA NA EMPRESA
AMADEUS

FORTALEZA

2025

DARIEL BEZERRA DE SOUSA

ANÁLISE, MEDIÇÃO E REDUÇÃO DO IMPACTO DA EMISSÃO DE CARBONO EM
SOFTWARE DE GRANDE PORTE: APLICAÇÃO PRÁTICA NA EMPRESA AMADEUS

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia De
Computação do Centro de Tecnologia da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de bacharel em
Engenharia De Computação.

Orientador: Prof. Dr. José Marques Soares

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S696a Sousa, Dariel Bezerra de.

Análise, medição e redução do Impacto da emissão de carbono em software de grande porte : aplicação prática na empresa Amadeus / Dariel Bezerra de Sousa. – 2025.
60 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia de Computação, Fortaleza, 2025.

Orientação: Prof. Dr. José Marques Soares.

1. Impact framework. 2. Green software. 3. Sustentabilidade. 4. Dados. I. Título.

CDD 621.39

DARIEL BEZERRA DE SOUSA

ANÁLISE, MEDIÇÃO E REDUÇÃO DO IMPACTO DA EMISSÃO DE CARBONO EM
SOFTWARE DE GRANDE PORTE: APLICAÇÃO PRÁTICA NA EMPRESA AMADEUS

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia De
Computação do Centro de Tecnologia da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de bacharel em
Engenharia De Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. José Marques Soares (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. George André Pereira Thé
Universidade Federal do Ceará (UFC)

Prof. Dr. Giovanni Cordeiro Barroso
Universidade Federal do Ceará (UFC)

À Deus e à minha família pelo constante apoio,
presença e investimento dedicados a mim ao
longo desta incrível jornada.

AGRADECIMENTOS

Expresso minha profunda gratidão a Deus pelas inúmeras oportunidades concedidas ao longo desta significativa trajetória.

Agradeço sinceramente à minha mãe, Maria Cleonice Bezerra Ribeiro e ao meus pais, Raimundo Daniel Teles de Sousa e Moacir de Paula Moura, cujo apoio incondicional, amor e força foram fundamentais para a concretização deste percurso.

Ao professor Dr. José Marques Soares, pelo acompanhamento atento e pelo valioso auxílio prestado durante esta fase tão importante.

Sou grato a todos os colegas e amigos, pelos momentos compartilhados de estudo e convivência, que enriqueceram e contribuíram para o meu desenvolvimento pessoal e profissional.

Finalmente, registro minha especial gratidão à Universidade Federal do Ceará e à École Centrale de Nantes pela honrosa oportunidade de participação no programa de duplo diploma, que proporcionou um significativo avanço em minha formação pessoal e profissional.

“O sucesso é a soma de pequenos esforços repetidos dia após dia.”

(Robert Collier)

RESUMO

Este trabalho teve como objetivo o desenvolvimento de uma infraestrutura para analisar, mensurar e possibilitar a tomada de decisões para a redução da pegada de carbono de aplicações computacionais em larga escala, no contexto corporativo. O projeto foi conduzido como parte de um estágio técnico realizado na empresa Amadeus, utilizando como base o projeto Carmen — um motor voltado à medição das emissões de carbono geradas pelas aplicações da organização. Ao longo do trabalho, foram implementadas funcionalidades para coleta automatizada de dados, construção de uma interface web interativa em Angular para exibição das métricas ambientais, bem como a integração com o Impact Framework, ferramenta desenvolvida pela Green Software Foundation. A metodologia adotada se destacou por possibilitar medições com base em dados reais de operação, superando limitações recorrentes identificadas na literatura, como a escassez de dados empíricos e a baixa replicabilidade em ambientes empresariais. Os resultados obtidos demonstram a viabilidade técnica da solução e seu potencial para apoiar decisões estratégicas orientadas à sustentabilidade digital. O trabalho também abre caminho para futuras melhorias, como a realização de simulações e a análise comparativa de diferentes cenários.

Palavras-chave: Impact Framework. Green Software. Sustentabilidade. Dados.

ABSTRACT

This work aimed to develop an infrastructure for analyzing, measuring, and enabling decision-making to reduce the carbon footprint of large-scale software applications in a corporate context. The project was carried out as part of a technical internship at Amadeus, based on the Carmen project — a measurement engine designed to assess the carbon emissions generated by the organization's applications. Throughout the project, functionalities were implemented for automated data collection, the development of an interactive web interface using Angular to display environmental metrics, as well as integration with the Impact Framework, a tool developed by the Green Software Foundation. The adopted methodology stood out by enabling measurements based on real operational data, overcoming recurring limitations identified in the literature, such as the scarcity of empirical data and low replicability in enterprise environments. The results demonstrate the technical feasibility of the solution and its potential to support strategic decisions guided by digital sustainability. This work also paves the way for future improvements, such as scenario simulations and comparative analyses of different contexts.

Keywords: Impact Framework. Green Software. Sustainability. Data.

LISTA DE FIGURAS

Figura 1 – Visão geral do arquivo Manifest	29
Figura 2 – Diagrama de Monitoramento, Análise e Relatórios	30
Figura 3 – Produtos Carmen	33
Figura 4 – Carbon Engine API	36
Figura 5 – Exibição do gráfico de CO ₂ ao longo do tempo na interface do usuário	37
Figura 6 – Campos da Interface do Usuário (UI)	38
Figura 7 – Aprimoramentos técnicos e visuais da UI	39
Figura 8 – Exibição comparativa de consumo de energia e emissões	39
Figura 9 – Referência visual: Dashboard no Grafana	40
Figura 10 – Gráficos de barras e linhas na versão final da interface	41
Figura 11 – Arquivos de entrada fornecidos pela equipe FinOps	43
Figura 12 – Arquivo Manifest de entrada para o Impact Framework (IF)	44
Figura 13 – Versão final da interface: gráficos de emissões, energia, CPU e memória para a aplicação <i>Master Pricer</i>	51
Figura 14 – Versão final da interface: gráficos de emissões, energia, CPU e memória para a aplicação <i>LSL</i>	52
Figura 15 – Carmen Relatando ao Daemon	60
Figura 16 – Carmen como um Serviço - Run Engine	61
Figura 17 – Carmen como um Serviço - Run Engine Hardware	61

LISTA DE TABELAS

Tabela 1 – Comparação entre os artigos – características científicas e analíticas.	22
Tabela 2 – Comparação entre os artigos – características científicas e analíticas.	23
Tabela 3 – Comparação entre os artigos – características científicas e analíticas.	24
Tabela 4 – Comparação entre os artigos – recursos e aplicações práticas.	24
Tabela 5 – Comparação entre os artigos – recursos e aplicações práticas.	25
Tabela 6 – Comparativo entre o dashboard no Grafana e a Interface Web em Angular .	50
Tabela 7 – Principais limitações observadas durante o projeto	53

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
Carmen	Carbon Measurement Engine
GEE	Gases de Efeito Estufa
GSF	Green Software Foundation
IF	Impact Framework
MVP	Minimum Viable Product
SCF	Software Carbon Footprint
SCI	Software Carbon Intensity

SUMÁRIO

1	INTRODUÇÃO	14
1.0.1	<i>Empresa Amadeus</i>	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.0.1	<i>Sustentabilidade na Computação</i>	18
2.0.2	<i>Green Software</i>	18
2.0.3	<i>Green Software Foundation (GSF)</i>	20
2.0.3.1	<i>Métrica Software Carbon Intensity (SCI)</i>	20
2.0.4	<i>Estado da Arte</i>	22
2.0.4.1	<i>Comparação entre a Literatura e a Metodologia Adotada</i>	25
2.0.5	<i>Considerações Finais</i>	26
3	METODOLOGIA	28
3.0.1	<i>Arquitetura e Modelo Proposto</i>	28
3.0.2	<i>Carbon Measurement Engine - Carmen</i>	30
3.0.2.1	<i>Telemetria de aplicações</i>	31
3.0.2.2	<i>Inventário de infraestrutura</i>	31
3.0.3	<i>Casos de Uso</i>	32
3.0.3.1	<i>MVP no Grafana</i>	32
3.0.3.2	<i>Interface do Usuário</i>	32
3.0.3.3	<i>FinOps PowerBI</i>	33
3.0.4	<i>Análise Estatística e Validação</i>	34
4	IMPLEMENTAÇÃO	35
4.1	Desenvolvimento de Interface do Usuário (UI)	35
4.1.1	<i>Conexão com a API e Prova de Conceito</i>	36
4.1.2	<i>Implementação de gráfico CO₂ ao Longo do Tempo para as aplicações escolhidas pelo usuário - Primeia Versão</i>	37
4.1.3	<i>Implementação de funcionalidades adicionais</i>	38
4.1.4	<i>Implementação de novas métricas</i>	39
4.2	Desenvolvimento do Carbon Measurement Engine (Carmen) Engine . .	41
4.2.1	<i>Ampliação da Resposta da API e Validação de Dados</i>	42
4.2.2	<i>Infraestrutura de Recuperação de Dados para o Caso de Uso FinOps . . .</i>	42

4.2.3	<i>Integração com o Impact Framework</i>	44
4.2.4	<i>Recuperação de Dados para Serviços de Armazenamento do FinOps</i> . . .	45
4.3	Estudo e análise	45
4.3.1	<i>Estimativa do Tamanho dos Arquivos no Pipeline Carmen/FinOps</i>	45
4.3.2	<i>Modelo de Serviços de Armazenamento: Estruturação de Entradas e Saídas</i>	47
4.4	Considerações Finais	49
5	RESULTADOS	50
5.1	Entregas Realizadas	50
5.2	Limitações	53
5.3	Eficiência e Destaques da Abordagem	54
6	CONCLUSÕES E TRABALHOS FUTUROS	55
	REFERÊNCIAS	57
	APÊNDICES	60
	ANEXOS	60
	ANEXO A – Diagramas dos Produtos Carmen	60
A.0.0.1	<i>Carmen Relatando ao Daemon</i>	60
A.0.0.2	<i>Carmen como um Serviço - Run Engine</i>	60
A.0.0.3	<i>Carmen como um Serviço - Run Engine Hardware</i>	60

1 INTRODUÇÃO

A intensificação das mudanças climáticas tem impulsionado empresas e instituições a repensarem seus modelos de operação, adotando práticas mais sustentáveis e conscientes em relação ao meio ambiente. No setor de tecnologia, tal movimento ganha ainda mais relevância, dado o crescimento exponencial da digitalização, da computação em nuvem e do consumo energético associado à execução e manutenção de sistemas computacionais.

De acordo com informações divulgadas pela empresa Ernst & Young (2024), uma das maiores do setor de redes de serviços profissionais, constatou-se que o setor de Tecnologias da Informação e Comunicação (TIC) representava cerca de 3,9% das emissões globais de carbono, com projeção de atingir 14% até 2040. Essa tendência ocorre devido ao aumento da demanda por serviços digitais, inteligência artificial e computação em nuvem.

Essa estimativa é reforçada por Freitag *et al.* (2021), que realizaram uma ampla revisão sobre o impacto ambiental das TIC, alertando para a possibilidade de subnotificação nos dados atuais e defendendo a necessidade urgente de ações coordenadas para controlar o crescimento das emissões. Os autores destacam que o setor de TIC não apenas consome energia diretamente, mas também estimula uma série de atividades computacionais indiretas, cujo impacto ambiental precisa ser melhor compreendido e medido.

Em linha com essa preocupação, Gupta *et al.* (2020) abordam a dificuldade em rastrear de forma precisa a pegada de carbono, ou seja, a quantidade total de emissões de gases de efeito estufa (GEE), expressa em CO₂ equivalente, associada direta ou indiretamente a operações computacionais, especialmente em sistemas distribuídos e com múltiplas camadas de abstração. O estudo chama a atenção para o fato de que, embora avanços em hardware e algoritmos venham melhorando a eficiência energética, o volume de operações computacionais cresce em ritmo acelerado, muitas vezes anulando os ganhos obtidos.

Complementando essa visão, Lannelongue *et al.* (2020) propuseram um modelo para estimar o impacto de tarefas computacionais baseado em parâmetros operacionais como tempo de execução, tipo de hardware e características do ambiente de execução. O trabalho, voltado especialmente para pesquisadores e profissionais da ciência de dados, contribui com ferramentas práticas para estimar e refletir sobre o impacto ambiental de suas rotinas computacionais.

Mais recentemente, Kennes (2023) mapeou o estado atual da mensuração da pegada de carbono em TI e identificou lacunas significativas, como a ausência de padronização nos métodos e a dificuldade de acesso a dados sobre o uso real de energia. O autor propõe uma cate-

gorização das fontes de emissão e ressalta a importância da colaboração entre desenvolvedores, gestores e provedores de infraestrutura para garantir a eficácia das ações.

Além disso, Li *et al.* (2023) exploram os impactos ambientais de sistemas de computação de alto desempenho (HPC), propondo métodos para estimar a pegada de carbono desses sistemas e discutir suas implicações ambientais. O estudo enfatiza a importância de integrar métricas de sustentabilidade na avaliação de desempenho de sistemas computacionais modernos, promovendo decisões mais conscientes sobre infraestrutura, uso de energia e eficiência computacional.

Apesar dos avanços técnicos observados nos últimos anos, muitas organizações ainda enfrentam barreiras culturais e estruturais para incorporar práticas sustentáveis de forma efetiva. A priorização de métricas de desempenho tradicionais, como tempo de resposta ou custo operacional, frequentemente sobrepõe considerações ambientais, dificultando a consolidação da sustentabilidade como um valor estratégico. Nesse cenário, a sensibilização dos profissionais e a disponibilização de ferramentas acessíveis e integradas ao fluxo de trabalho são elementos-chave para transformar boas intenções em ações concretas.

Dentro desse contexto, o software torna-se um componente central da discussão, sendo responsável tanto pelo consumo direto de energia quanto pelo uso indireto de recursos computacionais e infraestruturas distribuídas. Assim, medir e reduzir a pegada de carbono associada ao ciclo de vida do software tornou-se uma prática essencial, respaldada por iniciativas internacionais como a Green Software Foundation (GSF) e por diversos estudos acadêmicos que propõem métodos para quantificar e mitigar esse impacto.

Neste documento é apresentado de maneira didática e fundamentada um projeto desenvolvido para a empresa Amadeus, multinacional do setor de tecnologia voltada à indústria de viagens. O objetivo geral é medir, visualizar e reduzir a pegada de carbono das aplicações da empresa, oferecendo aos times de desenvolvimento e operação ferramentas acessíveis para monitorar o impacto ambiental de seus sistemas e tomar decisões alinhadas às metas de sustentabilidade da organização.

Os objetivos específicos correspondem ao desenvolvimento da versão *Minimum Viable Product (MVP)* de uma interface web para a exibição de gráficos da análise das emissões de carbono equivalente, energia consumida, requisições de cpu, requisições de memória, pelo tempo, de acordo com um conjunto de parâmetros definidos pelo usuário. Ademais, visava-se a implantação de novos *endpoints* na *Application Programming Interface (API)*, ou seja,

pontos de acesso a funcionalidades específicas expostas por uma interface, que consiste em um conjunto padronizado de definições e protocolos a qual permite a comunicação entre diferentes componentes de software entre si, de maneira estruturada e eficiente. Além disso, também foi estabelecida a implementação de uma nova infraestrutura conectada à nuvem e estudos para a adoção de novas métricas para a contabilização da pegada de carbono, tal como o tamanho do armazenamento dos dados.

O autor atuou de forma integrada à equipe *Software Carbon Footprint (SCF)*, contribuindo para a construção de dashboards, *APIs* e *pipelines* de dados que fornecem métricas ambientais relevantes em tempo real. A experiência proporcionou não apenas o desenvolvimento de competências técnicas em áreas como processamento de dados, design de interfaces e integração de *APIs*, mas também um envolvimento prático com os desafios atuais da sustentabilidade digital, oferecendo uma vivência concreta sobre como a Engenharia de Software pode contribuir para a atenuação da crise climática global.

Nesse sentido, além de oferecer uma visão prática sobre a incorporação de iniciativas sustentáveis no ciclo de vida do software, busca-se neste documento destacar a importância de soluções tecnológicas voltadas à sustentabilidade como um diferencial competitivo e ético, reforçando o papel da Engenharia de Software no enfrentamento das questões climáticas contemporâneas.

1.0.1 Empresa Amadeus

Considerando o contexto em que este projeto foi desenvolvido, é importante apresentar o cenário e a área de atuação da Amadeus, empresa no qual foi realizado a aplicação prática, onde utilizou-se as informações dos softwares desenvolvidos e comercializados por ela.

O Amadeus IT Group é uma empresa multinacional de tecnologia que fornece soluções para a indústria global de viagens e turismo. A empresa está estruturada em duas principais áreas de atuação: Distribuição e Soluções de TI.

A área de Distribuição tem como objetivo fornecer às agências de viagens um Sistema de Distribuição Global (GDS) que permite a busca, reserva e emissão de bilhetes de passagem aérea. Esse sistema facilita o trabalho das agências de viagens ao centralizar informações de diversas companhias aéreas, permitindo uma comparação e seleção mais eficientes para os clientes.

Por outro lado, a área de Soluções de TI oferece serviços tecnológicos para compa-

nhas aéreas gerenciarem suas operações internas. Essas soluções abrangem uma ampla gama de processos, desde a reserva de bilhetes até o registro de passageiros, incluindo gerenciamento de escalas e programas de fidelidade. Com essas tecnologias, as companhias aéreas buscam otimizar suas operações, melhorar a experiência dos passageiros e fortalecer a lealdade dos clientes, resultando em maior eficiência e satisfação geral.

O grupo emprega mais de 20.000 profissionais em mais de 150 países, atendendo a mais de 190 mercados, consolidando-se como uma das líderes em tecnologia para o setor de viagens.

Considerando o porte da empresa Amadeus e a dimensão de seus mercados, a grande quantidade de dados, de demanda por processamento e o consequente consumo de energia justificam o desenvolvimento de uma infraestrutura capaz de analisar, medir e que possibilite a tomada de decisão para a redução da pegada de carbono equivalente produzidos pelos seus sistemas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos fundamentais relacionados ao *Green Software*, abordando sua origem, definições e importância no contexto da sustentabilidade digital. Também são descritas as principais abordagens e métricas utilizadas para estimar a pegada de carbono de sistemas computacionais, com ênfase na métrica Software Carbon Intensity (SCI), proposta pela GSF, que permite quantificar o impacto ambiental do software ao longo de seu ciclo de vida. Além disso, no desenvolvimento do projeto, utilizou-se como referência a documentação, metodologia e frameworks fornecida pela GSF, o que será discutido posteriormente no capítulo 4.

2.0.1 Sustentabilidade na Computação

A sustentabilidade aplicada à computação diz respeito à adoção de práticas, estratégias e escolhas técnicas que minimizem o impacto ambiental dos sistemas computacionais. Tais impactos incluem o consumo de energia, as emissões de Gases de Efeito Estufa (GEE), o uso de recursos naturais e o descarte de equipamentos eletrônicos. Em um cenário de crescente digitalização e expansão de data centers, cresce também a responsabilidade do setor de tecnologia em adotar soluções conscientes em relação ao meio ambiente.

A sustentabilidade na computação não se limita ao hardware, mas envolve decisões em todas as camadas do sistema: arquitetura, desenvolvimento de software, infraestrutura, operação e até descarte. Surge, assim, o campo de estudo e prática do *Green Software*.

2.0.2 Green Software

O termo *Green Software* representa uma resposta à necessidade crescente de se repensar o desenvolvimento e a operação de sistemas computacionais sob uma perspectiva ambiental. Embora as discussões sobre eficiência energética na computação remontem à década de 1990, principalmente no contexto da engenharia de hardware e do design de data centers, foi a partir dos anos 2000 que o debate sobre a sustentabilidade do software propriamente dito começou a ganhar consistência na comunidade científica.

A partir do livro (CALERO; PIATTINI, 2015), diversas definições de *Green Software* são apresentadas, evidenciando múltiplas abordagens que enriquecem o entendimento do conceito.

Segundo (MURUGESAN; GANGADHARAN, 2012), *Green Software* pode ser compreendido como um software “amigo do ambiente” (*environment-friendly*), que contribui de diferentes formas para a sustentabilidade. Os autores classificam o *Green Software* em quatro categorias:

- Software mais ecológico, o qual consome menos energia para funcionar;
- Software embutido que torna outros dispositivos ou sistemas mais sustentáveis;
- Software voltado para comunicação e gestão de sustentabilidade ou emissões de carbono;
- Software para compreender as alterações climáticas, avaliar as suas implicações e formular resposta políticas adequadas.

Já Taina e Makinen (2011) propõe que um software verde deve cumprir três requisitos de alto nível:

1. Os processos de engenharia de software (desenvolvimento, manutenção e descarte) devem economizar recursos e reduzir resíduos;
2. A execução do software deve ser eficiente no uso de recursos;
3. O software deve contribuir ativamente para o desenvolvimento sustentável.

Por fim, Erdélyi (2013) oferece uma definição mais concisa, caracterizando o *Green Software* como “uma aplicação que produz o mínimo possível de resíduos durante o seu desenvolvimento e funcionamento”.

As definições apresentadas no livro Calero e Piattini (2015) evidenciam que, apesar das variações terminológicas, há convergência quanto à essência do *Green Software*. Em geral, o conceito envolve a minimização do consumo de recursos e da geração de resíduos, tanto no processo de desenvolvimento quanto na execução. Além disso, destaca-se a importância do software como ferramenta de apoio à sustentabilidade, seja por meio de sua função principal ou pelo impacto que gera em outros sistemas. Assim, o *Green Software* é compreendido como uma prática que une eficiência técnica e responsabilidade ambiental ao longo de todo o seu ciclo de vida.

A crescente relevância desse tema resultou na criação de iniciativas como a GSF, fundada em 2021, com o objetivo de padronizar práticas e métricas de sustentabilidade no desenvolvimento de software. Essa fundação tem atuado fortemente na criação de métodos como o SCI, fortalecendo a consolidação do *Green Software* como área estratégica na computação contemporânea.

Dessa forma, o *Green Software* não se resume à economia de energia. Ele representa

um novo paradigma de desenvolvimento, onde a sustentabilidade ambiental é integrada à lógica de projeto, aos requisitos funcionais e à visão de longo prazo dos sistemas. Em um cenário de emergência climática, essa abordagem oferece uma ponte entre tecnologia e responsabilidade socioambiental.

2.0.3 *Green Software Foundation (GSF)*

A GSF é uma fundação sem fins lucrativos, estabelecida sob a égide da Linux Foundation, sendo uma das principais iniciativas globais voltadas à promoção da sustentabilidade na Engenharia de Software por meio de práticas e ferramentas de código aberto. A organização foi criada em 2021 com o apoio de empresas como Microsoft, Accenture, GitHub e ThoughtWorks. Seu propósito central é desenvolver padrões, ferramentas e orientações que auxiliem desenvolvedores e organizações a medirem, entenderem e reduzirem o impacto ambiental dos sistemas computacionais.

A GSF adota uma abordagem holística ao tratar o software como parte integrante de um ecossistema mais amplo, atuando em toda a cadeia do ciclo de vida do software — desde sua concepção até sua desativação. Isso envolve não apenas a redução do consumo energético durante a execução de aplicações, mas também a minimização das emissões associadas às etapas de desenvolvimento, manutenção, testes, provisionamento de infraestrutura e descarte.

Além de fomentar a inovação e a conscientização ambiental na indústria de software, a fundação atua na criação de métricas padronizadas, entre as quais se destaca a SCI, considerada hoje uma das principais ferramentas para quantificar o impacto ambiental de sistemas computacionais.

2.0.3.1 *Métrica Software Carbon Intensity (SCI)*

A SCI é uma métrica proposta pela *Green Software Foundation* para mensurar a intensidade de carbono associada à operação e ao ciclo de vida do software. Ela representa a quantidade de emissões de CO_2 equivalente (CO_2e) geradas por unidade funcional de software, permitindo comparações entre diferentes versões, sistemas, arquiteturas e ambientes de execução.

A fórmula da SCI é dada pela equação 2.1:

$$SCI = \frac{(E \cdot I) + M}{R} \quad (2.1)$$

Em que:

- E (Energia consumida): corresponde à energia elétrica total utilizada pela aplicação durante seu funcionamento. Pode ser medida em diferentes escalas, como por datacenter, máquina virtual, contêiner, função específica ou chamadas de *API*. Deve considerar toda a energia consumida pela infraestrutura reservada, mesmo que parcialmente utilizada;
- I (Intensidade de carbono): representa a quantidade de CO_2e emitido por unidade de energia consumida ($gCO_2e/kW h$). Essa intensidade varia conforme a matriz energética da região em que a aplicação é executada. Em locais com energia renovável predominante, como hidrelétricas, a intensidade tende a ser menor. A GSF recomenda o uso de fatores de emissão marginais de curto ou longo prazo ou intensidades médias da rede local;
- M (Emissões incorporadas): refere-se às emissões associadas à fabricação, transporte e descarte do hardware utilizado. Para estimar esse valor, utiliza-se a fórmula 2.2:

$$M = TE \cdot TS \cdot RS = TE \cdot \frac{TiR}{EL} \cdot \frac{RR}{ToR} \quad (2.2)$$

Onde:

- TE : corresponde ao Total de Emissões incorporadas, ou seja, a soma das emissões da Avaliação do Ciclo de Vida (ACV) para todos os componentes de hardware;
- TiR : refere-se ao tempo reservado de uso do hardware para o software, expresso normalmente em horas;
- EL : trata-se da vida útil estimada do equipamento, expresso em horas, semanas ou anos, de acordo com o tipo de dispositivo;
- RR : consiste nos recursos totais do hardware.
- R (Unidade funcional): métrica de valor entregue pelo software, como número de requisições, chamadas de *API*, sessões de usuário, tarefas executadas ou qualquer outra métrica de output consistente. A escolha dessa unidade deve ser coerente entre os componentes analisados para permitir comparabilidade.

A componente $O = E \cdot I$ é definida como as emissões operacionais, e quando somada a M (emissões incorporadas), representa o total de emissões atribuíveis ao software em um determinado contexto.

2.0.4 Estado da Arte

Com o intuito de analisar e comparar a literatura existente sobre *Green Software* e sustentabilidade digital, realizou-se um levantamento de 30 artigos selecionados a partir de buscas por palavras-chave relacionadas aos temas “*Green Software*” e “*Green IT*”, os quais abordam diretamente esse assunto. Os quadros 1 a 5 apresentam essa análise comparativa, destacando diferentes aspectos tratados nos estudos. Especificamente, os quadros 1 a 3 reúnem informações sobre as características analíticas consideradas, enquanto os quadros 4 a 5 detalham os recursos práticos empregados em cada proposta. Essa estrutura visa facilitar a identificação de padrões, limitações e contribuições relevantes presentes na literatura atual, servindo como base para a discussão crítica apresentada a seguir.

Quadro 1 – Comparação entre os artigos – características científicas e analíticas.

Nº	Autor e Título (Ano)	Objetivo / Metodologia	Resultados / Contribuições	Análise Estatística	Limitações	Observações
1	(LI <i>et al.</i> , 2023) – Toward Sustainable HPC	Estimar a pegada de carbono em sistemas HPC por meio de benchmarks reais	Alto impacto ambiental dos HPC; propõem mitigação	Simulações e análise de logs	Difícil generalização para outras arquiteturas	Abordagem prática para sistemas específicos
2	(KENNES, 2023) – Measuring IT Carbon Footprint	Revisão crítica de abordagens existentes na mensuração de carbono em TI	Identifica falta de padronização e propõe diretrizes	Não aplicável	Ausência de dados empíricos	Ênfase teórica
3	(FREITAG <i>et al.</i> , 2021) – The Climate Impact of ICT	Revisar estimativas e regulamentações sobre TICs	Estima TICs com até 3.9% das emissões globais	Compilação de estudos	Estimativas divergentes; alta incerteza	Enfoque regulatório e de políticas públicas
4	(GUPTA <i>et al.</i> , 2020) – Chasing Carbon	Explorar desafios na quantificação da pegada de carbono	Propõe abordagem bottom-up; IA e datacenters subestimam emissões	Estimativas teóricas	Falta de dados reais dos fornecedores	Relevância para IA e grandes sistemas
5	(LANNELONGUE <i>et al.</i> , 2020) – Green Algorithms	Estimar impacto de modelos de NLP	Modelos grandes têm emissão comparável a veículos	Cálculo energético geolocalizado	Base em consumo estimado	Foco restrito a IA/NLP
6	(GULDNER <i>et al.</i> , 2024) – GSMM	Propor modelo padronizado de medição de software verde	Apresenta modelo GSMM como base comparativa	Testes t; descritiva	Não cobre todos os efeitos do software	Integra métodos de 10 grupos
7	(CASTOR, 2024) – Estimating Energy Footprint	Ensinar como medir consumo energético de software	Explicações práticas de três abordagens	Não aplicável	Trade-offs entre precisão e invasividade	Didático
8	(MEHRA <i>et al.</i> , 2022) – PGQ	Avaliar sustentabilidade via questionário	Avaliação leve aplicada a projetos reais	Não aplicável	Subjetividade; sem métricas técnicas	Útil para conscientização
9	(ABREU <i>et al.</i> , 2021) – Green Software Lab	Criar base técnica da disciplina de green software	Catálogo de métricas e ferramentas	Não aplicável	Foco limitado ao código-fonte	Suporte a desenvolvedores
10	(KAPOOR, 2024) – Green Software Quality Framework	Integrar métricas verdes ao SDLC	Redução de até 18% em uso de recursos	Comparação antes/depois	Exige ferramentas específicas	Integra sustentabilidade à qualidade
11	(VARTZIOTIS <i>et al.</i> , 2024) – Green Code Generation with LLM	Avaliar sustentabilidade de código gerado por IA	Proposta da métrica "capacidade verde"	Quantitativa	Limitado a LLMs e cenários específicos	Comparação IA vs. humanos

Quadro 2 – Comparação entre os artigos – características científicas e analíticas.

Nº	Autor e Título (Ano)	Objetivo / Metodologia	Resultados / Contribuições	Análise Estatística	Limitações	Observações
12	(KIM <i>et al.</i> , 2023) – GreenScale	Framework para edge-cloud com agendamento consciente de carbono	Redução de até 29,1% nas emissões	Avaliação experimental	Foco restrito a edge-cloud	Resultados promissores
13	(BASHIR <i>et al.</i> , 2024) – The Sunk Carbon Fallacy	Avaliar agendamento com reconhecimento de carbono incorporado em datacenters; propõe revisão de métricas tradicionais.	Incluir carbono embutido pode aumentar a pegada total; alerta para decisões equivocadas baseadas em emissões passadas.	Comparação entre métricas de carbono	Aplicável apenas a datacenters com perfil específico de uso	Aponta falhas em métricas convencionais de agendamento
14	(GIL <i>et al.</i> , 2018) – The Effect of Green Software	Prever correção e notas em e-learning com técnicas de mineração de dados (regressão, RNA, SVM).	Acurácia de 74%; desempenho depende de tempo de execução e uso de memória.	Comparação entre algoritmos de predição	Limitado ao contexto educacional	Usa software verde como variável explicativa
15	(PAZIENZA <i>et al.</i> , 2024) – Holistic Sustainable Computing	Propor framework ESC para sustentabilidade computacional ao longo do ciclo de vida de sistemas.	Ajuda a estruturar iniciativas sustentáveis e identificar pontos críticos de emissão.	Estudo de caso e abordagem qualitativa	Falta de exemplos práticos detalhados	Aplicação promissora em grandes organizações
16	(KERN <i>et al.</i> , 2018) – Sustainable Software Products	Propor indicadores hierárquicos para avaliar sustentabilidade de software.	Modelo causal para decisões de sustentabilidade em software.	Aplicação direta de critérios qualitativos	Ausência de experimentação empírica	Fornece base conceitual para futuros frameworks
17	(DODGE <i>et al.</i> , 2022) – Measuring Carbon Intensity of AI	Estimar intensidade de carbono de cargas de IA em diferentes regiões e horários da nuvem.	Variações geográficas e temporais afetam emissões; sugere otimização via agendamento.	Análise de variância entre regiões	Aplicável apenas ao contexto de IA na nuvem	Utiliza dados marginais de emissões regionais
18	(MARINI <i>et al.</i> , 2024) – Green AI: Programming Languages	Analisar consumo de energia entre linguagens de programação aplicadas à IA.	Linguagens compiladas têm menor consumo energético.	Avaliação experimental do consumo energético	Resultados limitados por escopo (linguagens/datasets)	Reforça escolha de linguagem como fator de sustentabilidade
19	(ZHOU <i>et al.</i> , 2024) – Regional Digital Infrastructure	Investigar o impacto da digitalização sobre a neutralidade de carbono em 291 cidades chinesas.	Infraestrutura digital favorece metas de carbono zero.	Propensity Score Matching + DID	Generalização limitada fora da China	Aplica modelagem robusta com dados reais em larga escala
20	(FORTI; BROGI, 2021) – Cloud-IoT Placement	Desenvolver metodologia declarativa para alocação sustentável de serviços no continuum Cloud-IoT.	Demonstra ganhos de eficiência energética e redução de carbono.	Protótipo com simulação controlada	Falta validação em escala real	Iniciativa pioneira para otimização verde no IoT
21	(LUNARDI <i>et al.</i> , 2011) – TI e Sustentabilidade	Identificar práticas de TI Verde com base em anúncios de empresas.	Catalogadas 37 práticas organizadas em 7 categorias.	Não especificado	Fonte limitada e não científica	Contribuição exploratória para mapeamento de práticas
22	(THEIS; SCHREIBER, 2021) – TI Verde e Sustentabilidade	Revisão sistemática do papel da TI Verde em estratégias corporativas de sustentabilidade.	Discute relação entre TI e objetivos ESG.	Análise qualitativa de literatura	Abordagem teórica, sem validação prática	Sugere diretrizes para integrar TI e responsabilidade ambiental

Quadro 3 – Comparação entre os artigos – características científicas e analíticas.

Nº	Autor e Título (Ano)	Objetivo / Metodologia	Resultados / Contribuições	Análise Estatística	Limitações	Observações
23	(SARAIVA <i>et al.</i> , 2021) – Green Software em Currículos de Computação	Survey com educadores e pesquisadores para analisar a presença do tema green software na educação.	Identifica carência de cursos específicos; propõe integração curricular.	Não especificado	Foco exclusivo em ensino	Reforça a necessidade de formação em sustentabilidade digital desde a graduação.
24	(SOUZA <i>et al.</i> , 2024) – CASPER: Carbon-Aware Scheduling	Propor sistema de agendamento com reconhecimento de carbono para serviços web.	Redução de até 70% nas emissões sem perda de desempenho.	Análise quantitativa com workloads reais.	Depende de dados precisos de intensidade de carbono.	Aplicável em cloud e edge; destaca eficiência ambiental no provisionamento.
25	(SUDARSHAN <i>et al.</i> , 2024) – GreenFPGA	Apresentar ferramenta para estimar pegada de carbono (CFP) de FPGAs; comparação com ASICs.	FPGAs mais eficientes em alguns contextos; ferramenta específica para CFP.	Comparativa (FPGAs x ASICs)	Aplicável apenas a FPGAs	Contribuição importante para hardware especializado, mas com escopo técnico restrito.
26	(DANUSHI <i>et al.</i> , 2025) – Carbon-Efficient Software Design	Revisão sistemática (RSL 5W1H) sobre técnicas para software eficiente em carbono.	Estrutura modelos, práticas e diretrizes em uma taxonomia.	Não aplicável	Limitações em escopo e bases usadas	Boa referência teórica para organizar o campo, mas sem validação prática.
27	(WASIF, 2024) – GSE e Desenvolvimento Sustentável	Estudo teórico integrando GSE com energias renováveis e ODS.	Sugere que GSE pode ampliar eficiência energética e sustentabilidade digital.	Não aplicável	Ausência de validação empírica	Abordagem conceitual promissora, mas ainda sem aplicações diretas.
28	(SILVA <i>et al.</i> ,) – Avaliação energética de software	Estudo de caso com monitoramento e análise de consumo de energia em aplicações.	Demonstra influência da implementação no consumo energético.	Não aplicável	Escopo limitado a poucos sistemas	Reforça a importância da implementação no impacto energético; falta generalização.
29	(LUNARDI <i>et al.</i> , 2014) – TI VERDE em Organizações	Estudo exploratório com 202 publicações sobre práticas e benefícios da TI Verde.	Lista práticas como reciclagem, virtualização e DCs verdes.	Análise de conteúdo	Amostra e recorte temporal limitados	Boa visão panorâmica; contribuições práticas, mas desatualizadas.
30	(MANUQUIAN; BOLIN,) – Green IT: práticas e tecnologias sustentáveis	Revisão conceitual sobre práticas e tecnologias verdes em TI.	Discorre sobre a importância da TI Verde para redução do impacto ambiental.	Não aplicável	Ausência de detalhamento técnico	Texto introdutório útil para contextualização, mas carece de profundidade técnica.

Quadro 4 – Comparação entre os artigos – recursos e aplicações práticas.

Nº	Área de Aplicação	Tipo de Dados	Dataset Utilizado	Frameworks / Ferramentas
1	HPC / Sustentabilidade	Logs de energia, métricas de uso de CPU/GPU	Benchmarks internos (não públicos)	Scripts próprios para coleta de dados, ferramentas de HPC
2	TI Corporativa / Sustentabilidade	Revisão documental	Sem dataset	Nenhum específico; foco em políticas e processos
3	TICs e Regulações Climáticas	Estudos publicados, estimativas governamentais	Compilação de estudos	Nenhum; abordagem baseada em literatura
4	Sistemas de Computação (IA, servidores)	Estimativas energéticas de cargas de trabalho	Próprios cenários de carga	Modelos energéticos customizados, simulações
5	Processamento de Linguagem Natural	Consumo de energia durante treino de modelos NLP	Modelos como BERT, GPT	PyTorch/TensorFlow (implícito), ferramenta de monitoramento de energia
6	Avaliação do consumo de energia e recursos de software	Dados de medição, metadados	Dataset SPECPower	Modelo de Estimativa de Uso de Energia em Nuvem, Software Footprint, Framework de Estimativa de Emissão, Metodologia de Medição de Overhead de Contêiner

Quadro 5 – Comparação entre os artigos – recursos e aplicações práticas.

Nº	Área de Aplicação	Tipo de Dados	Dataset Utilizado	Frameworks / Ferramentas
7	Desenvolvimento de Software Verde	Consumo de energia, potência	N/A	Contadores de desempenho de hardware, hardware de medição especializado, modelos analíticos
8	Projetos de Software	Escolhas feitas ao longo das diferentes dimensões do SDLC	N/A	N/A
9	Desenvolvimento de Software Verde	Código fonte do software	N/A	Analisador de perfil de energia, detector de "red smell"
10	Sistemas de Software	Consumo de energia, uso de CPU e memória, manutenibilidade, longevidade do código	N/A	N/A
11	Desenvolvimento de Software Sustentável	Código auto-gerado, gerado por humanos	Código gerado por IA (GitHub Copilot, ChatGPT-3, Code Whisperer) e código humano.	Modelos de Linguagem de IA
12	Computação Edge/Cloud	Características da carga de trabalho, intensidade de carbono	Dados de infraestrutura Edge-Cloud.	Framework GreenScale
13	Gerenciamento de Datacenter	Dados do servidor, emissões de carbono	Dados de datacenter do mundo real.	Métricas de contabilidade de carbono
14	E-learning	Exercícios de programação, submissões de alunos	Dados da ferramenta de avaliação online.	Regressão Logística, Árvores de Decisão, RNA, SVM
15	Computação Sustentável	Dados do ciclo de vida de sistemas de computação	Casos de uso da indústria bancária e multi-utilitária.	Framework ESC, Kubernetes, Kube-Green
16	Desenvolvimento de Software, Avaliação Ambiental	Propriedades do software, consumo de recursos	Não aplicável (Conceitual)	Não aplicável
17	Computação em Nuvem, IA	Consumo de energia, emissões de carbono	Dados de instâncias de nuvem Microsoft Azure	Microsoft Azure
18	Desenvolvimento de IA, Engenharia de Software	Consumo de energia, características da linguagem	3 datasets (não especificados)	C++, Java, Python, MATLAB, R
19	Desenvolvimento Urbano, Política Ambiental	Infraestrutura digital, emissões de carbono	Dados de painel de 291 cidades chinesas (2008-2021)	Não aplicável
20	Computação Cloud-IoT	Requisitos de serviço de aplicação, características da infraestrutura	Exemplo motivador	Prolog
21	TI Verde	Anúncios online	Não especificado	N/A
22	Sustentabilidade Corporativa	Literatura acadêmica	SciELO, SPELL	N/A
23	Educação em Computação	Respostas de surveys	Não especificado	N/A
24	Cloud-IoT	Requisitos de aplicações	Exemplo motivacional	Prolog
25	Hardware (FPGAs)	Ciclo de vida de FPGAs	Não especificado	GreenFPGA
26	Design e desenvolvimento de software com eficiência de carbono	Literatura sobre software eficiente em carbono	N/A (Revisão da Literatura)	N/A
27	Sistemas de energia renovável (smart grids, IoT, nuvem)	Estratégias de software para eficiência energética e redução de carbono	Não se aplica (abordagem teórica)	Práticas de GSE, algoritmos otimizados, low-power computing, cloud-aware scheduling
28	Avaliação energética de software	Gasto energético de aplicações	N/A (Estudo de Caso)	Não especificado
29	Práticas de TI Verde em organizações	Práticas e benefícios de TI Verde	Dados de 202 publicações	Não especificado
30	Green IT	Práticas e tecnologias sustentáveis de TI	N/A	Virtualização, equipamentos eficientes, gestão de resíduos, teletrabalho

2.0.4.1 Comparação entre a Literatura e a Metodologia Adotada

A partir da análise dos 30 artigos apresentados nos quadros 1 a 5, observa-se um cenário de avanço conceitual no campo do *Green Software*, refletido na diversidade de propostas, métricas e frameworks para mensuração da sustentabilidade digital. No entanto, apesar do crescente interesse acadêmico, muitas dessas abordagens ainda apresentam limitações relevantes quando confrontadas com os desafios de aplicabilidade em contextos corporativos reais.

Entre os principais pontos fracos identificados na literatura, destacam-se: a ausência de dados empíricos, com predominância de estudos baseados em estimativas teóricas, dados simulados ou análises conceituais (GUPTA *et al.*, 2020; KENNES, 2023); o foco restrito a protótipos, simulações ou ambientes acadêmicos, muitas vezes sem validação em sistemas reais (FORTI; BROGI, 2021; KAPOOR, 2024); a cobertura parcial do ciclo de vida do software, com ênfase apenas em fases específicas como execução ou código-fonte (ABREU *et al.*, 2021), negligenciando aspectos como desenvolvimento, testes, manutenção e descarte; e a falta de padronização metodológica, com grande variedade de métricas e ausência de consenso sobre unidades funcionais, dificultando a comparação entre os trabalhos.

Adicionalmente, muitos estudos concentram-se em contextos ou arquiteturas específicas, como sistemas HPC, edge-cloud ou aplicações de IA, o que limita a generalização de seus resultados para ambientes corporativos amplos, complexos e heterogêneos, como o da empresa Amadeus.

2.0.5 Considerações Finais

Em contraponto, a metodologia adotada neste trabalho propõe uma abordagem mais robusta e aderente à realidade empresarial, baseada na utilização do IF, uma ferramenta desenvolvida com base nos princípios da GSF. Ao ser aplicada diretamente em sistemas reais da Amadeus, essa abordagem superou diversas limitações recorrentes nos estudos revisados. Dentre os principais diferenciais, destacam-se:

- **A coleta automatizada de dados reais de execução** das aplicações, possibilitando medições com alto grau de precisão e aplicabilidade prática;
- O uso de uma **métrica padronizada e adaptada**, derivada da SCI, que considera tanto as emissões operacionais quanto as incorporadas, proporcionando uma visão abrangente do ciclo de vida do software;
- **A integração prática com a infraestrutura da empresa**, incluindo a implementação de *dashboards*, *pipelines* e *APIs* que viabilizam o acompanhamento contínuo das emissões, em tempo real;
- **A aderência a padrões internacionais reconhecidos**, o que favorece a replicabilidade da abordagem e sua comparação com outras práticas do setor.

Diferentemente de propostas limitadas a análises pontuais ou à validação teórica, neste trabalho demonstra-se a viabilidade de operacionalizar a mensuração da pegada de carbono

no fluxo de desenvolvimento de software de uma grande empresa, integrando sustentabilidade ambiental aos processos técnicos e de gestão.

É importante destacar que grande parte da literatura revisada ainda se concentra em abordagens conceituais ou protótipos acadêmicos, com escassez de validação empírica em ambientes reais. Nesse sentido, a presente proposta se diferencia por oferecer uma aplicação concreta baseada em dados operacionais, superando parte das limitações observadas.

Assim, esta pesquisa não apenas responde diretamente às fragilidades metodológicas recorrentes na literatura, como também oferece uma contribuição concreta, prática e replicável, ampliando o repertório de soluções disponíveis para organizações que desejam adotar práticas sustentáveis na Engenharia de Software. Ao alinhar teoria e prática, o trabalho reforça a importância de soluções que sejam efetivamente implementáveis em escala corporativa.

3 METODOLOGIA

Neste capítulo é apresentada a metodologia empregada no desenvolvimento do projeto voltado para mensuração e visualização da pegada de carbono de aplicações computacionais no contexto corporativo da empresa Amadeus.

Para isso, optou-se pela utilização do IF, uma ferramenta desenvolvida com base nos princípios da GSF, que operacionaliza a coleta e o cálculo automatizado das emissões de carbono a partir de dados reais de execução.

Com base nesta metodologia, foi possível implementar um processo automatizado de medição da intensidade de carbono, integrado a sistemas reais em operação, o que reforça a relevância e aplicabilidade da abordagem adotada no contexto corporativo. Além disso, essa escolha metodológica mitiga, de forma significativa, diversas limitações identificadas na literatura especializada, como a escassez de dados empíricos, a baixa replicabilidade em ambientes reais e o foco restrito a protótipos acadêmicos, oferecendo uma alternativa prática, confiável e alinhada às necessidades do setor.

3.0.1 Arquitetura e Modelo Proposto

O IF trata-se de uma das iniciativas significativas da GSF, um projeto de código aberto composto por diretrizes e ferramentas projetadas para auxiliar os desenvolvedores na avaliação, monitoramento e redução do impacto ambiental da produção de software.

A escolha pelo IF decorreu de uma análise criteriosa das ferramentas e abordagens atualmente disponíveis e mais alinhadas com os padrões emergentes de sustentabilidade digital. Além disso, a própria empresa Amadeus apoia institucionalmente a comunidade da Green Software Foundation, o que contribui para a integração prática do framework em seus processos internos de monitoramento ambiental.

O IF abrange metodologias para calcular a pegada de carbono do software, oferecendo uma abordagem sistemática para quantificar as emissões geradas ao longo do ciclo de vida do desenvolvimento de software, incluindo as fases de codificação, testes, implantação e manutenção. Ao utilizar essas metodologias, os desenvolvedores podem obter uma compreensão abrangente do impacto ambiental de seu software, o que é o primeiro passo em direção a uma redução significativa.

Além disso, o IF sugere práticas ecológicas que podem ser incorporadas ao ciclo de

vida do desenvolvimento de software. O framework propõe estratégias para integrar a sustentabilidade em cada etapa do desenvolvimento, incentivando a adoção de padrões de codificação sustentáveis, a implementação de sistemas de monitoramento contínuo para acompanhar o uso de energia e o estabelecimento de políticas que promovam o uso de tecnologias ecológicas. Ao incorporar essas práticas verdes no processo de desenvolvimento, é possível garantir que a sustentabilidade se torne um componente essencial dos projetos de software.

O IF também inclui ferramentas que facilitam a aplicação prática dessas diretrizes. Essas ferramentas ajudam os desenvolvedores a identificar e mitigar os impactos ambientais em tempo real, como estimar as emissões de carbono de operações específicas de software, monitorar o consumo de energia de ambientes de desenvolvimento e otimizar o uso de recursos em nuvem para reduzir o desperdício de energia.

O cálculo da pegada de carbono do software é realizado por meio da geração de um pipeline chamado arquivo *Manifest*. O IF também inclui um conjunto de plugins que podem ser utilizados de acordo com os modelos desejados para o pipeline. Na Figura 1 é mostrada a estrutura do arquivo *Manifest*. O conjunto completo de informações sobre o IF está disponível em (Green Software Foundation, 2025).

```
name:
description:
tags:
initialize:
  plugins:
    <PLUGIN-NAME-HERE>:
      method:
      path:
tree:
  children:
    child:
      pipeline:
        observe:
        regroup:
        compute:
      config:
      defaults:
      inputs:
        - timestamp: 2023-08-06T00:00
          duration: 3600
```

Figura 1 – Visão geral do arquivo Manifest

3.0.2 Carbon Measurement Engine - Carmen

O Carmen trata-se do projeto que visa o desenvolvimento do sistema de análise, monitoramento, simulação e relatórios sobre a pegada de carbono das aplicações da Amadeus. Nesse contexto, o projeto Carmen utiliza diretrizes e dados fornecidos pela *Green Software Foundation* (GSF), com a qual mantém uma parceria contínua, integrando o *Impact Framework* (IF) em sua infraestrutura, além da adoção de princípios definidos pela própria GSF, como a métrica *Software Carbon Intensity* (SCI). O Carmen segue o modelo proposto pela GSF, que considera fatores como consumo de energia, intensidade de carbono e um conjunto de suposições bem documentadas e transparentes. O objetivo também é contribuir para o desenvolvimento do IF, que foi estabelecido como um dos objetivos do Carmen.

Na Figura 2 é ilustrada o diagrama da infraestrutura do Carmen, destacando as informações de entrada, como o inventário de infraestrutura e as aplicações telemétricas, a interação constante com o IF para o cálculo da pegada de carbono, e a saída, que é dividida em três interfaces distintas, construídas separadamente: monitoramento, análise e relatórios.

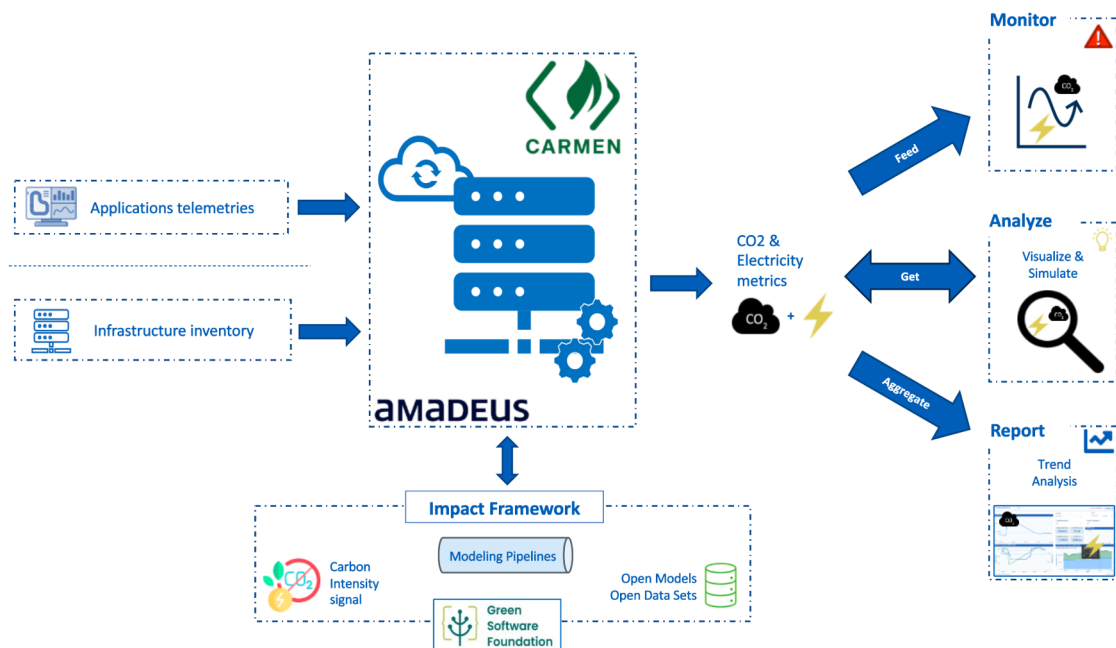


Figura 2 – Diagrama de Monitoramento, Análise e Relatórios

Ao analisar a Figura 2, identificamos os dois tipos de entradas: telemetria de aplicações e inventário de infraestrutura, os quais são discutidos a seguir.

3.0.2.1 Telemetria de aplicações

As telemetrias de aplicações consistem em um conjunto de informações sobre as aplicações necessárias para calcular as emissões de carbono. Para organizar e gerenciar essas aplicações em ambientes distribuídos, utilizamos o Kubernetes, uma plataforma de código aberto de orquestração de contêineres, os quais consistem em ambientes isolados que encapsulam uma aplicação e suas dependências para garantir que ela execute de forma consistente em diferentes sistemas, que automatiza a implantação, o dimensionamento e a gestão dos recursos. As telemetrias utilizadas são definidas a seguir.

- **Aplicações** refere-se ao nome da aplicação existente.
- **Plataforma como Serviço (PaaS)** refere-se a um modelo de computação em nuvem que oferece uma plataforma permitindo aos clientes desenvolver, executar e gerenciar aplicações sem a complexidade de construir e manter a infraestrutura subjacente.
- **Namespaces** são uma forma de organizar objetos e recursos em um ambiente de computação, frequentemente usados para evitar conflitos de nomes e gerenciar o escopo de vários componentes dentro de um sistema maior.
- **Entrega Orgânica Persistente (Persistent Organic Deliverable - POD)** no Kubernetes, é a menor unidade implantável, composta por um ou mais contêineres que compartilham a mesma rede, armazenamento e ciclo de vida. Ele agrupa contêineres que precisam trabalhar juntos e é gerenciado como uma única entidade para execução e escalabilidade.
- **Cluster** refere-se a um conjunto de máquinas (nós), sejam físicas ou virtuais, que trabalham juntas para executar aplicações, distribuindo a carga de trabalho. No Kubernetes, um cluster é composto por nós que executam os Pods (onde as aplicações residem) e gerenciam os recursos do sistema.

Essa entrada é usada no caso de uso da interface do usuário (UI), que será apresentado posteriormente.

3.0.2.2 Inventário de infraestrutura

O inventário de infraestrutura refere-se à lista de nós e máquinas virtuais utilizadas como entrada no caso de uso de FinOps, que será discutido ulteriormente. Abaixo estão as definições para nós e máquinas virtuais.

- **Máquinas Virtuais (MVs)** são instâncias de sistemas operacionais virtuais que executam

dentro de um ambiente físico ou virtualizado. Elas permitem que múltiplos sistemas operacionais e aplicações operem em um único servidor físico, compartilhando recursos como CPU, memória e armazenamento. As MVs são isoladas umas das outras e possuem suas próprias configurações e sistemas operacionais.

- **A lista de nós**, no contexto do Kubernetes, refere-se à coleção de todos os nós (máquinas físicas ou virtuais) que fazem parte de um cluster. Cada nó na lista executa serviços e aplicações, gerencia recursos e é responsável por hospedar os PODs. A lista de nós fornece uma visão geral dos recursos disponíveis no cluster e auxilia na gestão e distribuição das cargas de trabalho.

3.0.3 Casos de Uso

Ao analisar a Figura 2, identificamos três tipos de saídas: monitoramento, análise e relatório. Para cada tipo de saída, desenvolveu-se no projeto ou está em processo de desenvolvimento de um caso de uso distinto, que será detalhado nas seções a seguir.

3.0.3.1 MVP no Grafana

Um MVP é uma versão simplificada de um produto que inclui apenas as funcionalidades essenciais necessárias para atender às necessidades básicas dos usuários e validar a ideia do produto.

Nesse sentido, utiliza-se um MVP no Grafana para monitoramento, empregando as informações de telemetria das aplicações como metodologia. O Grafana é uma plataforma de código aberto para visualização e análise de dados. Ele permite que os usuários criem *dashboards* e gráficos interativos a partir de dados coletados de diversas fontes, como bancos de dados, sistemas de monitoramento e serviços em nuvem. O Grafana é amplamente utilizado para monitorar métricas e *logs*, criar *dashboards* personalizados e realizar análises de desempenho em tempo real.

3.0.3.2 Interface do Usuário

A interface do usuário (UI), é o caso de uso para análise baseada em solicitações sob demanda dos usuários, utilizando dados de telemetria de aplicativos.

3.0.3.3 FinOps PowerBI

FinOps, abreviação de *Financial Operations*, é uma abordagem que busca integrar equipes de finanças, operações e engenharia para promover uma gestão mais eficiente dos custos e recursos em ambientes de computação em nuvem. Essa prática tem como objetivo fornecer visibilidade financeira contínua, permitindo que as organizações tomem decisões baseadas em dados para otimizar seus investimentos em infraestrutura de TI, equilibrando desempenho, custo e sustentabilidade.

O *Power BI*, desenvolvido pela Microsoft, é uma ferramenta de visualização e análise de dados que permite aos usuários criar *dashboards* e relatórios interativos a partir de diversas fontes de dados. Ele transforma dados brutos em *insights* visuais, suporta integração com diferentes serviços e bancos de dados e facilita a tomada de decisões empresariais por meio de visualizações dinâmicas e relatórios compartilháveis.

O *FinOps PowerBI* é usado para relatórios, empregando a metodologia de inventário de infraestrutura. Nesse contexto, todo o inventário é processado diariamente, para ser exibido no *PowerBI* para relatórios de longo prazo.

Na Figura 3 é fornecido um resumo dos três casos de uso e das tecnologias usadas em cada um.

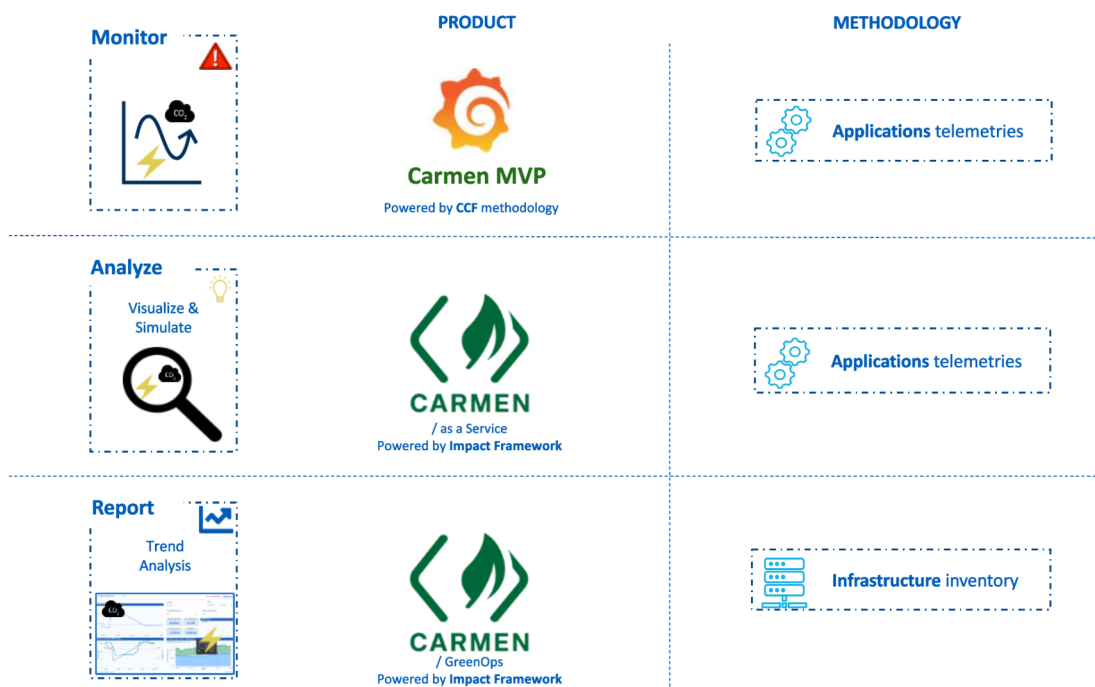


Figura 3 – Produtos Carmen

Além disso, o anexo A contém diagramas que delineiam a estrutura de cada um dos Produtos Carmen.

Desse modo, é importante destacar que a principal contribuição do autor com este projeto foi o desenvolvimento da interface de Análise, denominada Carmen como um Serviço (*Carmen as a Service*).

3.0.4 Análise Estatística e Validação

Embora o foco principal deste trabalho tenha sido o desenvolvimento da infraestrutura e integração do IF no contexto corporativo da Amadeus, foi conduzido um processo de verificação e validação contínua para assegurar a consistência dos dados coletados e o correto funcionamento dos sistemas implementados.

Do ponto de vista estatístico, as análises se concentraram na consistência dos valores gerados automaticamente a partir das entradas fornecidas pelo inventário de infraestrutura e pela telemetria das aplicações. Foram avaliadas métricas como consumo de energia, intensidade de carbono e a emissão estimada por unidade funcional. A análise envolveu a comparação entre os resultados gerados em diferentes ambientes (*clusters*, *namespaces* e *PaaS*) e a identificação de possíveis anomalias, como variações inesperadas ou ausência de dados.

A validação foi realizada de forma prática, com foco na integração correta dos dados com o IF e no funcionamento dos *pipelines* de medição. A construção da interface de Análise (Carmen como um Serviço) permitiu visualizar os dados de emissão em tempo real, contribuindo para a verificação da eficiência da arquitetura proposta.

Cabe destacar que o projeto ainda estava em andamento no momento da finalização da participação do autor, o que justifica a existência de ajustes pendentes e inconsistências em algumas partes do sistema. Ainda assim, a estrutura metodológica adotada demonstrou-se funcional e viável, estabelecendo uma base sólida para futuras melhorias, expansão dos módulos e refinamento das medições.

4 IMPLEMENTAÇÃO

Neste capítulo é descrito com detalhes técnicos as contribuições específicas para o projeto, organizadas em três frentes principais: o desenvolvimento da interface do usuário (*front-end*), o aprimoramento do Carmen Engine (*back-end*) e a realização de análises e estudos voltados a futuras implementações. Cada uma dessas áreas demandou a aplicação de habilidades técnicas específicas, contribuindo para uma experiência abrangente e alinhada aos objetivos do projeto.

A contribuição do autor concentrou-se majoritariamente no desenvolvimento da interface do usuário (UI), uma vez que não havia uma versão funcional disponível. As tarefas envolveram desde a criação de elementos visuais intuitivos até a implementação de funcionalidades essenciais para a interação eficiente com os dados de pegada de carbono. Também foi realizada a integração entre a API e a interface, com foco na exibição clara dos dados por meio de gráficos interativos, assegurando usabilidade e responsividade.

No âmbito do Carmen Engine, foram implementadas funcionalidades responsáveis pela captação e transmissão de dados ambientais. Isso incluiu a escrita e adaptação de *scripts*, testes de funcionamento e ajustes para garantir a confiabilidade do sistema. Além disso, foram realizadas contribuições para a refatoração de trechos de código, buscando maior legibilidade, aderência aos padrões definidos e facilidade de manutenção futura.

Por fim, foram desenvolvidas atividades de análise e estudo para orientar possíveis evoluções do sistema. Essas atividades envolveram investigações técnicas, reuniões com a equipe de FinOps e levantamento de novas abordagens que pudessem complementar ou otimizar a infraestrutura já existente. A meta foi identificar oportunidades de inovação e aprimoramento contínuo da solução, mantendo-a alinhada às demandas internas da empresa e às diretrizes de sustentabilidade digital.

A seguir, são apresentados em detalhes os conjuntos de atividades executadas em cada uma dessas frentes, evidenciando as contribuições específicas e os principais resultados alcançados ao longo da implementação.

4.1 Desenvolvimento de Interface do Usuário (UI)

Esta seção apresenta o trabalho desenvolvido relacionado ao caso de uso da interface do usuário, conforme apresentado na seção 3.0.3.2 Interface do Usuário.

4.1.1 Conexão com a API e Prova de Conceito

Uma das tarefas do projeto era desenvolver uma interface de usuário capaz de exibir os dados de pegada de carbono de forma clara e interativa, em sintonia com as visualizações previamente disponíveis no Grafana. Para isso, foi necessário construir um modelo de prova de conceito (*proof of concept*), com o intuito de validar a viabilidade da criação de um *dashboard* funcional e alinhado com os dados processados pela API.

Desse modo, efetuou-se a integração com a API desenvolvida pela equipe, construída com o framework FastAPI em Python, apresentada na Figura 4.

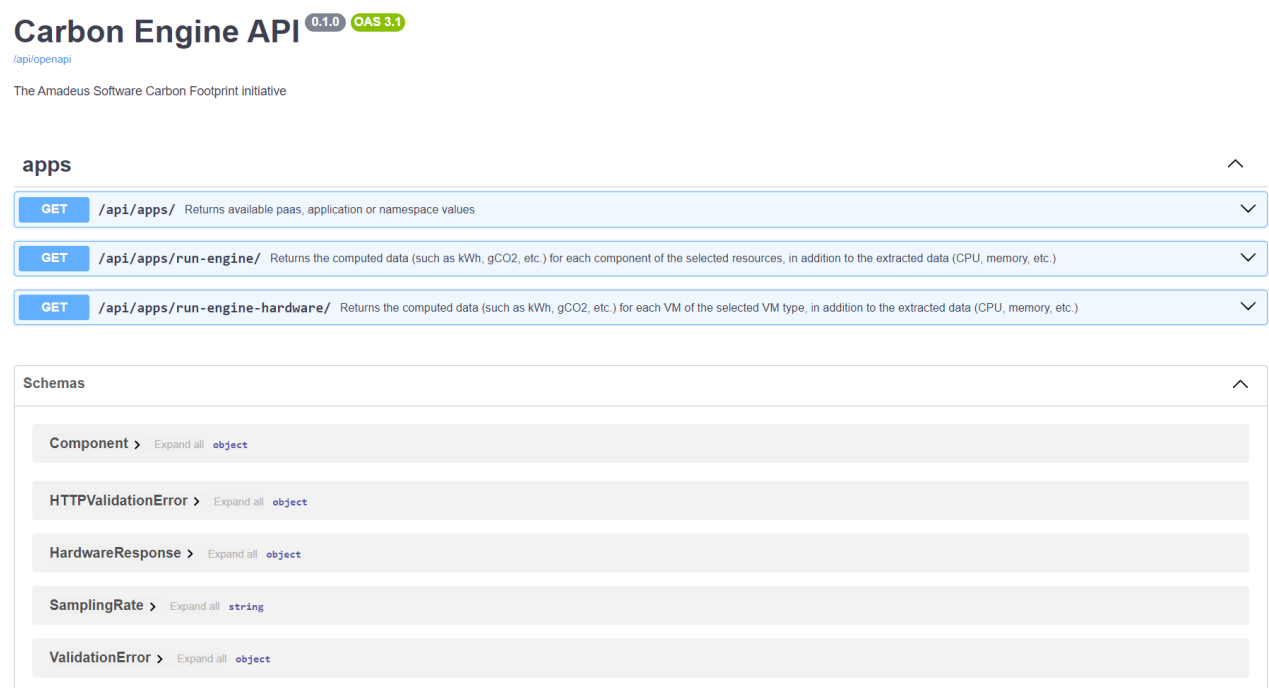


Figura 4 – Carbon Engine API

Essa API fornece os dados para a interface, por meio de três principais *endpoints*:

- `/api/apps/` : retorna a lista de *Platform as a Service (PaaS)*, nomes de aplicações e seus *namespaces*.
- `/api/apps/run-engine/`: fornece dados calculados (como consumo de energia em kWh e emissões em gCO2) e dados extraídos (uso de CPU, memória, entre outros) por aplicação.
- `/api/apps/run-engine-hardware/` retorna informações semelhantes ao endpoint anterior, mas com foco em máquinas virtuais específicas, no nível de hardware.

Esses dados são acessados por meio de requisições *HTTP* e processados na interface para exibição. As requisições são feitas a partir das *URLs* geradas pelos endpoints da API,

recuperando os dados em formato *JSON* e armazenando-os em variáveis da aplicação.

A biblioteca **D3.js** foi utilizada para a renderização dos gráficos, por sua flexibilidade e robustez em visualizações baseadas em dados. Já o *design* visual da interface foi desenvolvido utilizando o **Design Factory**, um *framework* interno da Amadeus semelhante ao *Bootstrap*, garantindo aderência aos padrões visuais da empresa.

Após a integração bem-sucedida com a *API* e a visualização inicial dos dados de uma aplicação, o modelo de prova de conceito foi concluído com sucesso. Essa primeira etapa confirmou a viabilidade técnica do *dashboard* e serviu de base para futuras evoluções da interface, incluindo a extensão da visualização para múltiplas aplicações da Amadeus.

4.1.2 Implementação de gráfico CO₂ ao Longo do Tempo para as aplicações escolhidas pelo usuário - Primeira Versão

A funcionalidade de visualização gráfica das emissões de CO₂ ao longo do tempo foi implementada, permitindo a seleção dinâmica de diferentes aplicações e plataformas (*PaaS*). Desse modo, buscou-se ampliar a flexibilidade da interface para atender a diferentes necessidades de análise e comparação de dados ambientais.

Na Figura 5 é apresentada o resultado dessa implementação, demonstrando a exibição do gráfico interativo com base nos parâmetros definidos pelo usuário.

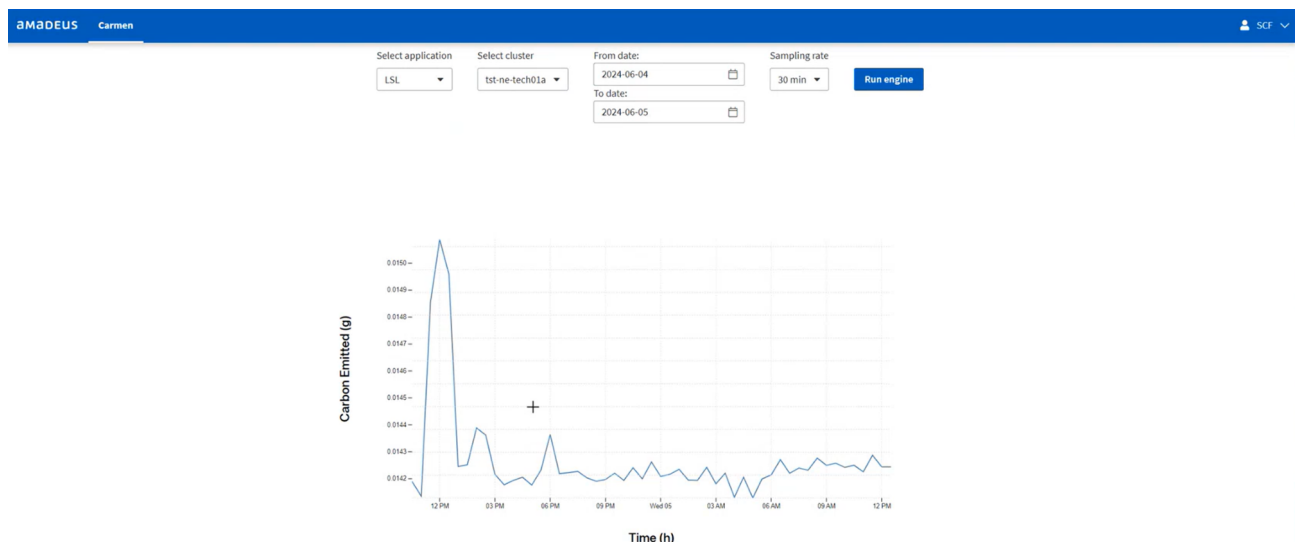


Figura 5 – Exibição do gráfico de CO₂ ao longo do tempo na interface do usuário

Os componentes da interface permitem a definição de diversos parâmetros de entrada, como nome da aplicação, plataforma (*cluster*), intervalo de datas e taxa de amostragem. Inicial-

mente, a seleção era limitada a uma única aplicação e um único *cluster*, mas a funcionalidade foi posteriormente expandida para suportar múltiplas seleções simultâneas, incluindo o filtro por *namespaces*. Essa evolução proporcionou uma interface mais robusta e alinhada com a estrutura de requisição da *API*.

Além disso, foi implementada a funcionalidade de zoom interativo no gráfico, com a opção de redefinição, visando facilitar a análise de séries temporais com grande volume de dados. Essa funcionalidade contribui para uma melhor experiência de navegação e leitura dos dados apresentados, especialmente em contextos com variações significativas na granularidade das amostras.

4.1.3 Implementação de funcionalidades adicionais

Após a criação da primeira versão funcional da interface, novas funcionalidades foram gradualmente incorporadas com o objetivo de torná-la mais flexível, informativa e alinhada com a experiência esperada pelos usuários da Amadeus.

Foram incluídos novos filtros que permitem ao usuário selecionar múltiplas aplicações, *clusters* (*PaaS*) e *namespaces*, com atualização dinâmica dos campos conforme as seleções realizadas. Essa abordagem proporcionou maior granularidade e controle sobre os dados exibidos. Na Figura 6 é exemplificado os campos disponíveis para entrada de dados.

A imagem mostra a interface de usuário (UI) com os seguintes campos:

- Select application:** Um menu suspenso com a opção "MASTER-PRICER" selecionada.
- Select cluster:** Um menu suspenso com "ERD4" e "ERD6" selecionados. Abaixo dele, há uma lista de opções com caixas de seleção: "All", "erd4" (selecionada), "erd6" (selecionada), "nld11", "prd-fc-shop09a", "prd-fc-shop09b", e "prd-qwc-shop08a".
- Select Namespace:** Um menu suspenso com a opção "ALL" selecionada.
- From date:** Um campo de texto com o valor "2024-07-29" e um ícone de calendário.
- To date:** Um campo de texto com o valor "2024-07-30" e um ícone de calendário.
- Sampling rate:** Um menu suspenso com a opção "30 min" selecionada.
- Run engine:** Um botão azul com o texto "Run engine".

Figura 6 – Campos da Interface do Usuário (UI)

Do ponto de vista técnico, diversas melhorias foram aplicadas para otimizar a usabilidade da aplicação. Isso incluiu:

- Ordenação alfabética das opções nos campos de seleção;
- Definição de seleções padrão mais apropriadas;
- Inclusão de logs de desempenho e resposta da *API*;
- Exibição de indicadores de carregamento ao lado de cada campo;
- Atualização das legendas dos gráficos;

- Reformulação do design da barra de navegação, agora incluindo os logotipos do projeto e cores padronizadas da Amadeus.

Na Figura 7 é ilustrada parte dessas melhorias implementadas.

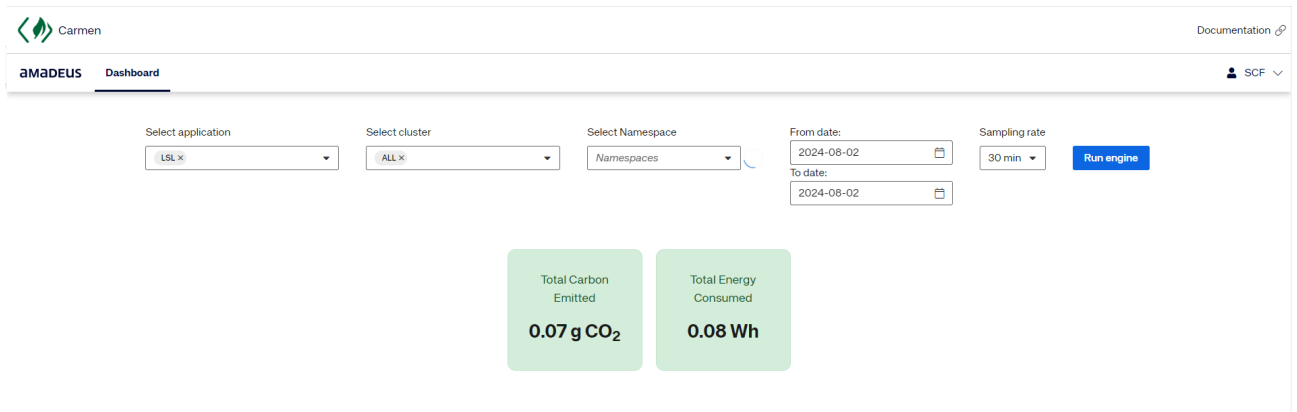


Figura 7 – Aprimoramentos técnicos e visuais da UI

4.1.4 Implementação de novas métricas

Com a estrutura funcional consolidada, a interface passou a oferecer um conjunto completo de recursos para visualização de métricas ambientais das aplicações monitoradas. Inicialmente, os gráficos disponíveis exibiam emissões de carbono ao longo do tempo. Em seguida, foi adicionada a visualização do consumo de energia, utilizando a mesma estrutura gráfica e permitindo a comparação entre as duas métricas, conforme ilustrado na Figura 8.



Figura 8 – Exibição comparativa de consumo de energia e emissões

Considerando a evolução do produto, também foram adicionadas novas métricas

relacionadas ao uso de recursos computacionais, como memória e *CPU* solicitadas. Com base na análise do MVP no Grafana (Figura 9), adotou-se o uso de gráficos de barras para emissões e energia, e gráficos de linha para *CPU* e memória, promovendo uma visualização mais clara e alinhada às práticas já conhecidas pelos usuários da plataforma.



Figura 9 – Referência visual: Dashboard no Grafana

Na Figura 10 é apresentada a versão final da visualização implementada, incluindo os gráficos combinados, informações numéricas consolidadas e interações como *tooltips*, que

facilitam a leitura dos dados pelo usuário.

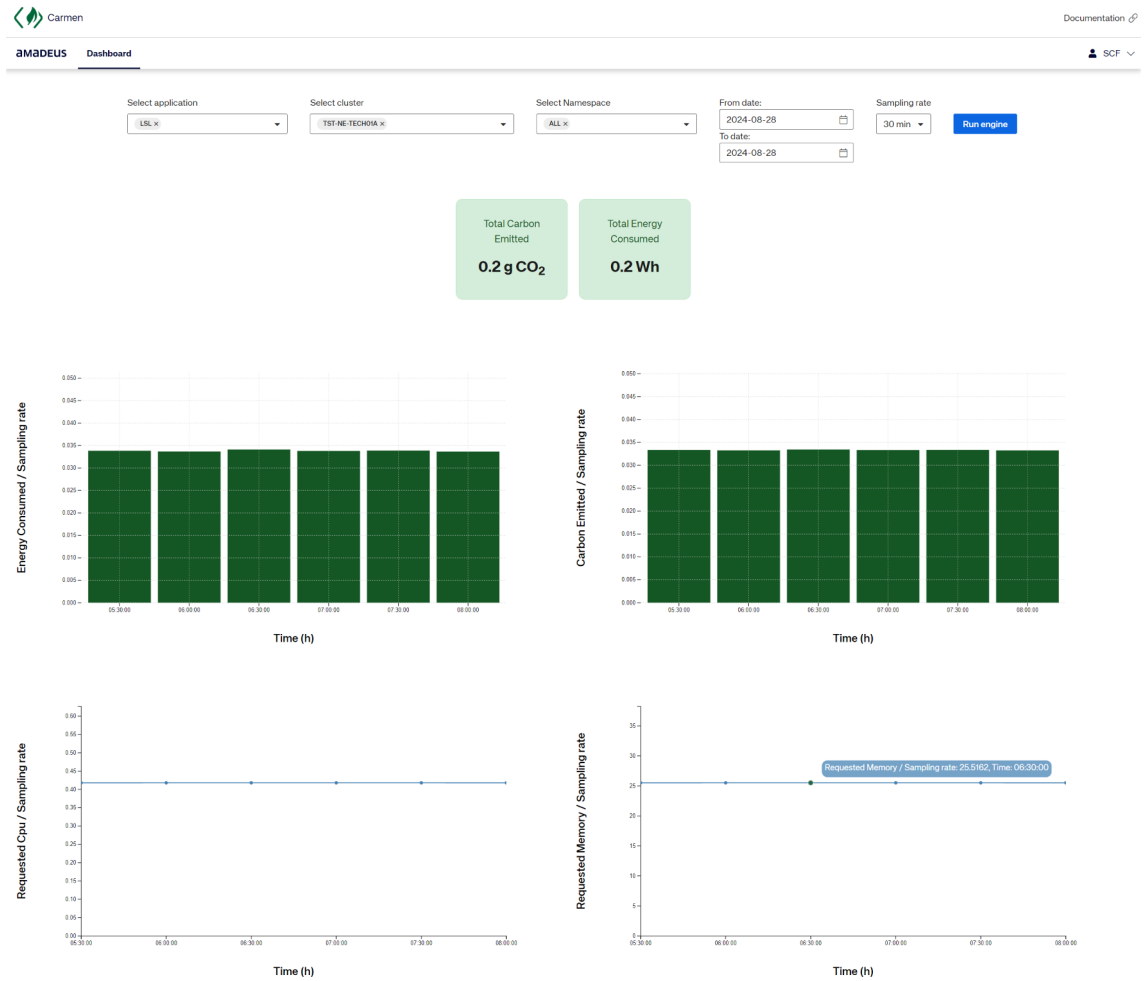


Figura 10 – Gráficos de barras e linhas na versão final da interface

Com essas melhorias, a interface atingiu um grau de maturidade, permitindo que os usuários realizem até certo ponto, análises sobre a eficiência energética e o impacto ambiental das aplicações corporativas monitoradas.

4.2 Desenvolvimento do Carmen Engine

Esta seção apresenta o trabalho de aprimoramento do Carmen Engine, *back-end* do projeto, responsável pela construção da *pipeline*, envio, recebimento dos resultados do *IF* e exibição na *API*.

4.2.1 *Ampliação da Resposta da API e Validação de Dados*

Com o objetivo de viabilizar a exibição de novas métricas na interface do usuário, foi necessário expandir a resposta da *API* responsável pelo fornecimento dos dados. Para isso, foram incluídas novas telemetrias na resposta do *endpoint run-engine* como memória solicitada, *CPU* solicitada, *CPU* utilizada, *PaaS*, *namespace* e *pod*. Essas informações passaram a ser fundamentais para a geração de gráficos mais detalhados na *UI*, permitindo análises mais ricas sobre o desempenho e o impacto ambiental das aplicações.

A atualização foi realizada a partir de uma compreensão aprofundada da arquitetura da *API* e da forma como os dados eram tratados internamente. A atividade contou com suporte técnico da equipe responsável pelo desenvolvimento inicial da *API*, possibilitando uma integração adequada dos novos campos na estrutura existente.

Para assegurar a estabilidade e o correto funcionamento dos *endpoints* atualizados, foram desenvolvidos casos de teste automatizados utilizando o *framework Robot*. Esses testes validaram a integridade das respostas da *API*, verificando a presença dos novos campos e o formato dos dados retornados.

Além de garantir maior robustez à *API*, essa etapa também contribuiu para o aprendizado prático de novas ferramentas, como o *framework Robot*, ampliando o domínio técnico sobre estratégias de validação automatizada em *APIs RESTful*.

4.2.2 *Infraestrutura de Recuperação de Dados para o Caso de Uso FinOps*

Nesta seção, descreve-se o desenvolvimento da infraestrutura necessária para a recuperação de dados utilizados no cálculo da pegada de carbono de máquinas virtuais, no contexto do caso de uso *FinOps*, conforme apresentado na Seção 3.0.3.3. Essa infraestrutura visa permitir a integração entre os dados operacionais da plataforma Azure e o mecanismo de cálculo do sistema Carmen, possibilitando a visualização dos resultados em ferramentas corporativas como o *PowerBI*.

Com a migração dos serviços da Amadeus para a nuvem *Azure*, tornou-se necessário acessar periodicamente arquivos com informações sobre o uso de recursos computacionais. Esses dados são disponibilizados pela equipe interna de *FinOps*, a partir da coleta automática dos registros de uso da *Azure*, e são essenciais para o cálculo das emissões de carbono associadas ao consumo de energia das máquinas virtuais.

O processo de implementação envolveu a criação de mecanismos de autenticação e autorização, incluindo a configuração de uma entidade de serviço (*service principal*), definição de permissões em um grupo de acesso (*AMACP*), e o provisionamento de um contêiner para armazenar os dados. Com isso, foi possível estabelecer um fluxo seguro e contínuo para a recuperação dos arquivos gerados pelo *FinOps*.

Na Figura 11 é apresentada um exemplo dos arquivos utilizados como fonte de dados:

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type
ppt_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:10	Block Blob	application/octet-stream
dts_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:17	Block Blob	application/octet-stream
lab_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:17	Block Blob	application/octet-stream
rnd_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:22	Block Blob	application/octet-stream
hos_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:29	Block Blob	application/octet-stream
amacp_vm_usage-2024-04-01-00.csv	Hot (inferred)		01/04/2024 03:35	Block Blob	application/octet-stream
ppt_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:10	Block Blob	application/octet-stream
dts_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:17	Block Blob	application/octet-stream
lab_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:17	Block Blob	application/octet-stream
rnd_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:23	Block Blob	application/octet-stream
hos_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:30	Block Blob	application/octet-stream
amacp_vm_usage-2024-04-01-01.csv	Hot (inferred)		01/04/2024 04:35	Block Blob	application/octet-stream
ppt_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:10	Block Blob	application/octet-stream
dts_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:17	Block Blob	application/octet-stream
lab_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:17	Block Blob	application/octet-stream
rnd_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:23	Block Blob	application/octet-stream
hos_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:30	Block Blob	application/octet-stream
amacp_vm_usage-2024-04-01-02.csv	Hot (inferred)		01/04/2024 05:36	Block Blob	application/octet-stream
ppt_vm_usage-2024-04-01-03.csv	Hot (inferred)		01/04/2024 06:10	Block Blob	application/octet-stream
dts_vm_usage-2024-04-01-03.csv	Hot (inferred)		01/04/2024 06:17	Block Blob	application/octet-stream

Figura 11 – Arquivos de entrada fornecidos pela equipe FinOps

Cada arquivo representa um *tenant*, ou seja, uma instância separada da infraestrutura na *Azure*, associada a um ambiente ou conjunto de aplicações. Atualmente, são processados seis arquivos por hora (*ppt*, *dts*, *lab*, *rnd*, *hos* e *amacp*), totalizando até 144 arquivos diários.

Para a leitura desses arquivos, foi desenvolvida a função `read_file_from_datalake` responsável por processar os dados armazenados no contêiner da Azure. A função recebe como entrada um dicionário com os arquivos e pastas de destino, e retorna um dicionário estruturado com as informações extraídas, agrupadas por *ID* de máquina virtual. Esses dados alimentam o mecanismo de cálculo do Carmen, que utiliza o IF para determinar as emissões de carbono associadas ao uso computacional.

Essa implementação representa um passo importante para consolidar o caso de uso *FinOps*, permitindo a geração de relatórios periódicos e confiáveis sobre o impacto ambiental da infraestrutura em nuvem da Amadeus.

4.2.3 Integração com o Impact Framework

A integração com o IF foi uma etapa essencial para a operacionalização do cálculo da pegada de carbono a partir dos dados fornecidos pela equipe de *FinOps*. A partir dos arquivos fornecidos por *tenant*, em formato *Excel*, foram identificadas e selecionadas as colunas relevantes que serviriam como dados de entrada para a construção do *pipeline* de análise no IF.

Com base nesse conjunto de dados, foram extraídas as colunas necessárias para alimentar o *pipeline* do IF, incluindo informações como *id*, uso médio de CPU, memória disponível (GB), a memória total (GB), tamanho do serviço, tipo de instância, ambiente de partição, número de CPUs virtuais, componentes e região. Essas colunas formaram a base para a construção do arquivo de entrada (*manifest file*), que é a principal estrutura de *input* exigida pelo *framework*.

Na Figura 12 é ilustrado parte do *manifest file* gerado com esses dados.

```
tree:
  children:
    child:
      pipeline:
        - cloud_metadata
        - teads-curve
        - e-mem
        - sci-e
        - sci-m
        - sci-o
        - sci
      config:
        sci:
          functional-unit-time: 3 hours
          functional-unit: requests
      defaults:
        cloud/vendor: azure
        cloud/instance-type: Standard_D13_v2
        grid/carbon-intensity: 251.0
        memory/capacity: 56
        memory/utilization: 100
        device/emissions-embodied: 1887000
        device/expected-lifespan: 126144000
      inputs:
        - timestamp: 2024-06-26 08:52:00+00:00
          duration: 1
          cpu/utilization: 13
        - timestamp: 2024-06-26 08:53:00+00:00
          duration: 1
          cpu/utilization: 13
        - timestamp: 2024-06-26 08:54:00+00:00
          duration: 1
          cpu/utilization: 11
        - timestamp: 2024-06-26 08:55:00+00:00
          duration: 1
          cpu/utilization: 12
        - timestamp: 2024-06-26T09:59:00.000Z
          duration: 1
          cpu/utilization: 0
          cloud/vendor: azure
          cloud/instance-type: Standard_D13_v2
          grid/carbon-intensity: 251
          memory/capacity: 56
          memory/utilization: 100
          device/emissions-embodied: 1887000
          device/expected-lifespan: 126144000
          vcpus-allocated: 8
          vcpus-total: 64
          memory-available: 56
          physical-processor: >-
            Intel® Xeon® Platinum 8370C, Intel® Xeon® Platinum 8272CL, Intel®
            Xeon® 8171M 2.1 GHz, Intel® Xeon® E5-2673 v4 2.3 GHz, Intel® Xeon®
            E5-2673 v3 2.4 GHz
          cpu/thermal-design-power: 270
          cpu/energy: 0.0000011249999999999999
          memory/energy: 0.021952
          energy: 0.021953125
          carbon-embodied: 0.001869886796042618
          carbon-operational: 5.5102343750000005
          carbon: 5.512104261796043
          sci: 59530.726027397264
          aggregated:
            carbon: 374.8291317325559
```

Figura 12 – Arquivo Manifest de entrada para o IF

Após o processamento pelo IF, o *manifest file* é enriquecido com os dados calculados, retornando informações sobre o consumo de energia e as emissões de carbono associadas à operação da infraestrutura analisada.

O resultado final desse processo é um arquivo no formato *Excel*, consolidando os dados de entrada e as métricas geradas, o qual é entregue novamente à equipe de *FinOps*. Esse

arquivo contém, além dos dados brutos, as estimativas de:

- Energy (kWh)
- Operational Carbon (gCO₂e)
- Embodied Carbon (gCO₂e)

Esse conjunto de informações é posteriormente utilizado na ferramenta *PowerBI*, correspondente ao caso de Uso do *FinOps*, para visualização e análise contínua do impacto ambiental das operações computacionais, permitindo à organização adotar medidas mais sustentáveis com base em dados reais e atualizados.

4.2.4 Recuperação de Dados para Serviços de Armazenamento do *FinOps*

Com base nos resultados obtidos na tarefa 4.3.2, descrita na Seção 4.3, foi implementada a infraestrutura de recuperação de dados provenientes do *FinOps* especificamente voltada aos serviços de armazenamento. Essa etapa visou estender a metodologia de cálculo da pegada de carbono, inicialmente baseada apenas no consumo de CPU e memória, para também considerar o impacto associado ao uso de armazenamento.

Para viabilizar essa funcionalidade, foi desenvolvida uma função dedicada à leitura de arquivos no formato *Parquet*, denominados por seu alto desempenho e menor tamanho em comparação a arquivos *CSV* tradicionais. A função `read_parquet_from_datalake` é responsável por acessar os arquivos diretamente do *Data Lake* da *Azure*, uma arquitetura de armazenamento que permite guardar grandes volumes de dados brutos para análise posterior. Os dados processados por essa função foram então integrados ao IF.

Essa função recebe como parâmetros o caminho dos arquivos no ambiente da *Azure* e seus respectivos nomes, retornando uma lista estruturada de dicionários contendo os dados extraídos. A partir dessa estrutura, tornou-se possível estender os cálculos de emissões para considerar o volume de armazenamento consumido pelas aplicações, contribuindo para uma visão mais abrangente do impacto ambiental da infraestrutura computacional.

4.3 Estudo e análise

4.3.1 Estimativa do Tamanho dos Arquivos no Pipeline *Carmen/FinOps*

Durante a integração entre os sistemas *Carmen* e *FinOps*, foi necessário estimar o volume de dados processados ao longo do *pipeline*, visando garantir que os arquivos gerados

estivessem dentro das limitações operacionais do *FinOps*, especialmente no que diz respeito à capacidade de carregamento diário no *PowerBI*. Essa análise tornou-se fundamental uma vez que os dados reais completos ainda não estavam totalmente disponíveis no momento da implementação.

Para isso, foi realizada uma avaliação dos arquivos de entrada recebidos do *FinOps*, bem como uma simulação do tamanho dos arquivos de saída gerados pelo Carmen após o processamento das emissões de carbono. A estimativa envolveu a criação de arquivos de teste com dados sintéticos, utilizando scripts em *Python*.

Além disso, foi considerada a diferença entre os dois formatos utilizados: *CSV* e *Parquet*. Os arquivos *Parquet*, por serem comprimidos, apresentaram tamanho significativamente menor, aproximadamente cinco vezes menores que seus equivalentes em *CSV*, embora requeiram ferramentas específicas para leitura, diferentemente do formato *CSV*, que é mais acessível, porém mais pesado.

Com base nessa análise, os dados foram organizados e documentados em uma página interna da equipe, a fim de auxiliar na tomada de decisão sobre o formato ideal a ser adotado. A seguir, apresenta-se um resumo das estimativas realizadas:

Estimativa de Tamanho dos Arquivos no Pipeline Carmen/FinOps

Arquivos de entrada recebidos do *FinOps* (por *tenant*, por dia):

Tenant	CSV	Parquet
AMACP	≈ 264 MiB	≈ 27.38 MiB
DTS	≈ 32.44 MiB	≈ 2.65 MiB
HOS	≈ 8.41 MiB	≈ 1.22 MiB
LAB	≈ 9.24 KiB	≈ 154.19 KiB
PPT	≈ 9.24 KiB	≈ 154.19 KiB
RND	≈ 15.59 MiB	≈ 2.06 MiB
Total diário	≈ 320.46 MiB	≈ 33.61 MiB

Arquivo de saída gerado pelo Carmen:

- CSV: ≈ 10 MiB
- Parquet: ≈ 1 MiB

Estimativa dos arquivos de CO₂ alocados criados pelo FinOps:

- CSV: ≈ 1.4 GB
- Parquet: ≈ 210 MiB

Com essas estimativas, foi possível orientar a escolha do formato de arquivo mais adequado ao fluxo de trabalho, conciliando desempenho, acessibilidade e escalabilidade da solução proposta.

4.3.2 Modelo de Serviços de Armazenamento: Estruturação de Entradas e Saídas

Para complementar o cálculo da pegada de carbono das aplicações, além das informações sobre máquinas virtuais, tornou-se necessário considerar os serviços de armazenamento utilizados na infraestrutura da Amadeus. Esta etapa visou ampliar a precisão da análise ao incluir o consumo de energia e as emissões relacionadas ao uso de discos e volumes persistentes em nuvem.

O processo começou com a investigação das fontes de dados disponíveis na plataforma Azure, em colaboração com a equipe de FinOps. Após obter acesso aos arquivos de cobrança e uso dos serviços de armazenamento, foi possível identificar as colunas essenciais para alimentar o Impact Framework. Esses arquivos, disponibilizados nos formatos CSV e Parquet,

foram analisados para extrair as informações relevantes, com ênfase no uso do formato Parquet, devido à sua eficiência no armazenamento e leitura de grandes volumes de dados.

Com base nessa análise, foi construída uma especificação contendo as colunas selecionadas como entradas para o pipeline. Entre os principais campos identificados, destacam-se: identificador do recurso, tipo e categoria do armazenamento, região do data center, capacidade alocada (em terabytes) e tipo de disco (HDD ou SSD). Essa estrutura foi documentada de forma colaborativa em uma página técnica da equipe, permitindo alinhamento com os demais desenvolvedores envolvidos na construção do modelo.

Principais Entradas para o Impact Framework

- Identificador do recurso (Resource ID e Subscription ID)
- Categoria do medidor (Meter e Meter Category)
- Tipo de produto (Storage Product)
- Região do recurso (Resource Region)
- Capacidade do disco (Storage Disk Size) – em TB

Com essas informações, o framework pode calcular o consumo de energia com base em coeficientes específicos por tipo de hardware e região. Para o tipo de disco, os valores seguem a metodologia do Cloud Carbon Footprint:

- HDD: 0.65 W/TB
- SSD: 1.20 W/TB

Além das entradas, também foi estruturado o formato de saída a ser enviado ao FinOps, com os dados calculados de consumo energético, pegada de carbono e custo estimado por tipo de armazenamento. Esses dados alimentam os painéis de visualização e permitem análises comparativas por tecnologia.

Principais Saídas do Impact Framework

- Coeficiente de consumo por tipo de disco (W/TB)
- Emissões por tipo de armazenamento (gCO_{2eq})
- Custo por tipo de armazenamento (EUR)
- Consumo total de energia por tipo (kWh)

Essa estruturação foi um passo essencial para tornar o modelo mais completo e compatível com os padrões de sustentabilidade digital. Além disso, a modularidade adotada

permite que novas metodologias ou fontes de coeficientes possam ser integradas futuramente, oferecendo flexibilidade para adaptações e melhorias contínuas.

4.4 Considerações Finais

Neste capítulo, foram apresentadas as principais etapas de implementação do projeto, destacando as atividades realizadas no desenvolvimento da interface de usuário, na construção do motor de cálculo (Carmen Engine) e na integração com o Impact Framework. As tarefas envolveram desde a criação de componentes visuais e interativos até o aprimoramento técnico de funcionalidades, garantindo maior usabilidade, precisão e robustez do sistema. Além disso, foi detalhada a infraestrutura necessária para a recuperação, processamento e envio de dados relacionados ao consumo de recursos computacionais, com ênfase nas máquinas virtuais e nos serviços de armazenamento.

As seções também mostraram a preocupação com a escalabilidade da solução, a adequação aos requisitos corporativos e a integração com ferramentas já utilizadas na empresa, como PowerBI e a plataforma Azure. A modularidade e flexibilidade da infraestrutura desenvolvida permitem sua continuidade e evolução mesmo após o encerramento desta etapa do projeto.

A seguir, no capítulo de Resultados, serão apresentados os principais dados obtidos a partir da infraestrutura implementada, bem como análises sobre sua eficácia, eventuais inconsistências encontradas durante a execução e os ganhos práticos obtidos com a aplicação da metodologia no ambiente corporativo da Amadeus.

5 RESULTADOS

Neste capítulo são apresentados os principais resultados alcançados ao longo do desenvolvimento do projeto. As entregas compreendem a implementação de funcionalidades para mensuração e visualização da pegada de carbono de aplicações computacionais da Amadeus, por meio da integração do *Impact Framework* com dados reais extraídos da infraestrutura corporativa. Além disso, é apresentada a versão funcional da interface web desenvolvida, bem como uma análise crítica das limitações observadas na metodologia.

5.1 Entregas Realizadas

O resultado mais significativo deste trabalho foi a entrega de uma interface *web* funcional, capaz de consumir os dados fornecidos pela *API* do Carmen Engine e apresentar visualizações interativas sobre as emissões de carbono, consumo de energia e uso de recursos computacionais. Essa interface oferece filtros dinâmicos por aplicação, *PaaS (clusters)* e *namespaces*, além de permitir a seleção de intervalos de tempo e amostragem.

Nas Figuras 13 e 14 são ilustradas duas execuções da versão final proposta pelo autor. Elas apresentam a capacidade da aplicação de representar visualmente diferentes métricas por meio de gráficos de linhas e de barras, oferecendo ao usuário uma visão clara e detalhada do impacto ambiental das aplicações monitoradas.

Na Tabela 6 é exibida uma comparação entre o *dashboard* da interface *web* feito em *Angular* e o *dashboard* utilizando Grafana.

Tabela 6 – Comparativo entre o dashboard no Grafana e a Interface Web em Angular

Aspecto	Grafana (MVP)	Interface Web (Angular)
Objetivo Principal	Visualização de séries temporais de dados de carbono e energia	Plataforma interativa e personalizável para análise ambiental detalhada
Flexibilidade de Filtros	Filtros limitados (aplicação e tempo)	Filtros dinâmicos por aplicação, cluster, namespace, data e amostragem
Capacidade de Expansão	Limitado à visualização dos dados existentes	Potencial para suportar simulações, análises comparativas e novas métricas
Integração com APIs Externas	Integração mais complexa e limitada	Integração direta com o Carmen Engine e o Impact Framework
Tecnologias Utilizadas	Grafana + Prometheus (visualização padrão)	Angular + D3.js + Design Factory + FastAPI (backend)
Customização de Gráficos	Baixa (baseada em templates do Grafana)	Alta (gráficos customizados com linhas, barras e tooltips)
Interatividade com o Usuário	Visualização passiva	Interface responsiva e interativa com logs, carregamento e feedback visual
Capacidade de Evolução	Limitado às funcionalidades nativas do Grafana	Estrutura modular que permite evolução contínua do produto
Alinhamento com Sustentabilidade	Exibe métricas, mas não interage com frameworks especializados	Integrado ao Impact Framework da Green Software Foundation



Figura 13 – Versão final da interface: gráficos de emissões, energia, CPU e memória para a aplicação *Master Pricer*

Ao analisar a Tabela 6, observa-se que a interface web desenvolvida em Angular apresenta vantagens significativas em relação ao dashboard construído no Grafana. A solução em Angular oferece maior nível de personalização dos gráficos, permitindo o uso de diferentes tipos de visualizações (como linhas, barras e tooltips), além de uma interface responsiva e interativa que proporciona uma experiência mais fluida para o usuário.

Outro ponto de destaque é a flexibilidade no uso de filtros dinâmicos, que possibilitam a seleção de dados por aplicação, cluster, namespace, intervalo de tempo e amostragem, superando as limitações do Grafana, que possui opções de filtragem mais restritas.

Além disso, a interface em Angular demonstra um potencial superior de evolução e escalabilidade, com estrutura modular que permite a incorporação de novas funcionalidades, como simulações, análises comparativas e integração com frameworks de sustentabilidade, como o Impact Framework. A integração direta com o Carmen Engine e a API FastAPI também garante

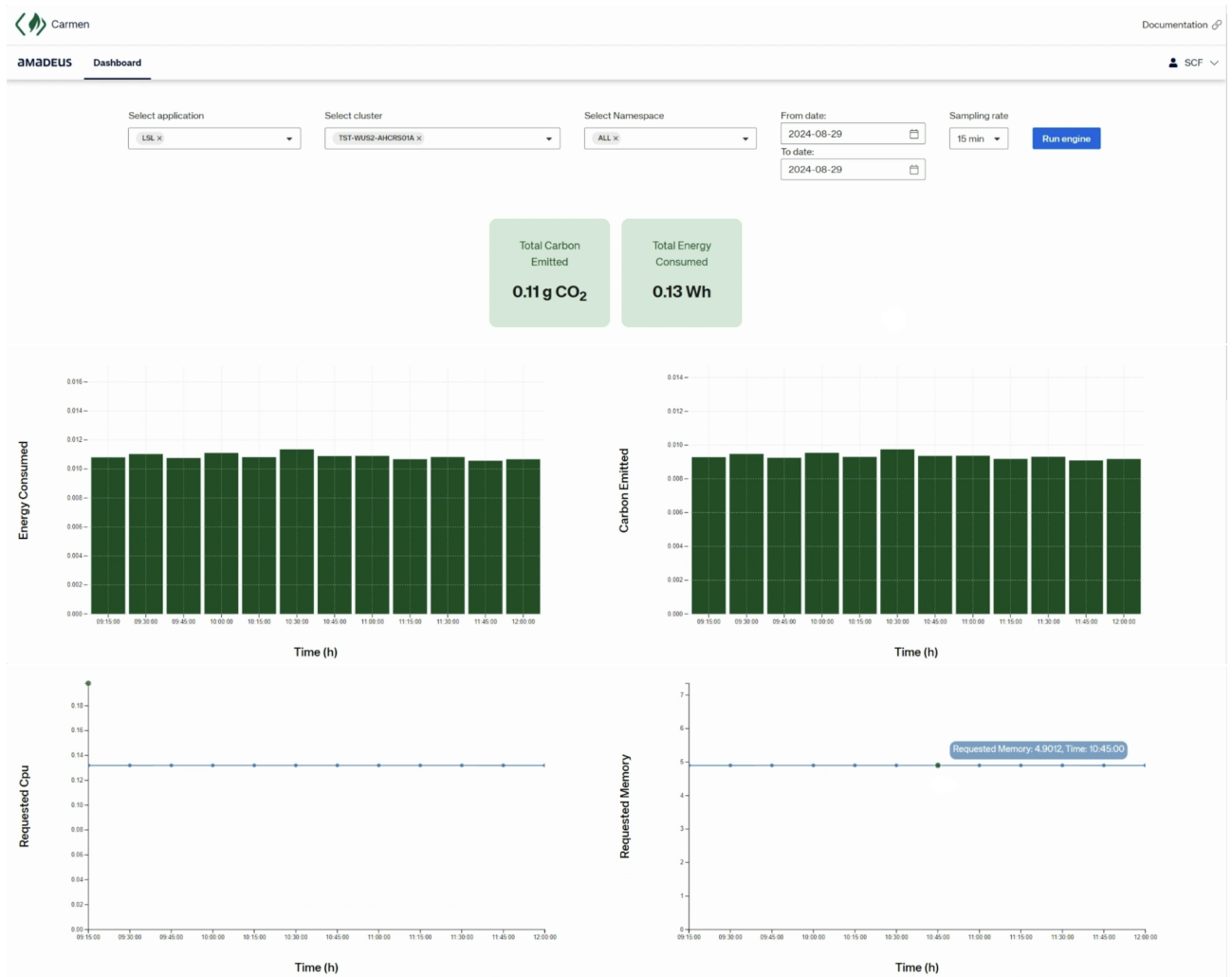


Figura 14 – Versão final da interface: gráficos de emissões, energia, CPU e memória para a aplicação *LSL*

maior fluidez na comunicação entre sistemas, favorecendo o desenvolvimento de soluções mais completas e alinhadas com os objetivos ambientais da empresa.

Além da UI, destacam-se como entregas relevantes:

- Integração da interface com a API construída em *FastAPI*, incluindo novos endpoints e adição de telemetrias;
- Desenvolvimento de testes automatizados para validação dos endpoints principais;
- Construção de uma infraestrutura de coleta de dados do FinOps para alimentar o Impact Framework com dados reais da Azure;
- Implementação de um pipeline completo que permite a análise de emissões em serviços de armazenamento, além das máquinas virtuais;
- Geração de *manifest files* e envio de resultados processados de volta ao FinOps, possibilitando visualização no PowerBI.

Esses resultados reforçam a aplicabilidade prática da abordagem proposta, ao integrar sustentabilidade digital em um ambiente corporativo real, com potencial de escalabilidade e replicabilidade.

5.2 Limitações

Embora a metodologia adotada tenha se mostrado eficiente e tenha superado várias limitações presentes na literatura, algumas restrições foram identificadas ao longo da execução do projeto. Essas limitações estão relacionadas tanto ao estágio de desenvolvimento da solução quanto à dependência de fatores externos, como a disponibilidade de dados e o escopo das métricas atualmente integradas.

Na Tabela 7 é resumida as principais limitações observadas, organizando-as por categoria e fornecendo uma breve descrição de cada uma.

Tabela 7 – Principais limitações observadas durante o projeto

Categoria	Descrição da Limitação
Projeto em andamento	No momento da finalização deste trabalho, o projeto ainda estava em desenvolvimento, o que impediu a validação completa de todos os módulos implementados.
Divergência de resultados na UI	Inconsistências foram identificadas entre os valores exibidos na interface web e os dados de referência, indicando necessidade de ajustes na lógica de visualização ou nos filtros aplicados.
Dependência dos dados do FinOps	A execução do pipeline depende da disponibilidade, integridade e regularidade dos arquivos fornecidos pelo FinOps. Falhas no envio ou na estruturação dos dados podem comprometer os resultados.
Cobertura parcial de métricas ambientais	A solução contempla telemetrias de CPU, memória e armazenamento, mas ainda não inclui métricas como tráfego de rede, descarte de hardware e emissões indiretas.

Essas limitações não comprometem a relevância dos resultados obtidos até o momento, mas evidenciam oportunidades claras para aprimoramentos futuros, especialmente no que diz respeito à robustez da visualização, à ampliação da cobertura de métricas ambientais e à estabilidade do fluxo de dados com o *FinOps*.

5.3 Eficiência e Destaques da Abordagem

A metodologia baseada no uso do IF, associada ao desenvolvimento de uma infraestrutura robusta de coleta, processamento e visualização de dados, apresentou forte potencial para promover práticas sustentáveis no ciclo de vida de software corporativo.

Dentre os principais diferenciais da abordagem, destacam-se:

- A coleta automatizada de dados reais de execução, com alta fidelidade e replicabilidade;
- A integração direta com sistemas de produção da Amadeus, garantindo aplicabilidade concreta;
- A capacidade de oferecer insights visuais em tempo real sobre consumo de energia e emissões;
- A aderência aos princípios estabelecidos pela GSF, promovendo alinhamento com padrões reconhecidos internacionalmente.

Esses aspectos evidenciam que, mesmo com limitações pontuais, o modelo proposto representa um avanço relevante no campo da sustentabilidade digital, tanto em termos técnicos quanto estratégicos.

O próximo capítulo apresenta uma reflexão final sobre o trabalho desenvolvido, bem como as perspectivas futuras para a evolução do projeto e novas oportunidades de pesquisa e melhoria.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi apresentado um estudo sobre as práticas e metodologias voltadas à sustentabilidade na Engenharia de Software, bem como o desenvolvimento de uma solução prática capaz de mensurar, analisar e possibilitar a redução da pegada de carbono de aplicações computacionais em ambientes corporativos, com foco na empresa Amadeus. Para isso, foi adotado o IF, uma ferramenta aberta e em constante evolução proposta pela GSF, que permite calcular as emissões de carbono com base em dados reais de execução.

A abordagem adotada se destaca por utilizar dados reais extraídos de ambientes produtivos, promovendo maior fidelidade nas medições e relevância prática na aplicação. Além disso, a integração com o Impact Framework garante alinhamento com metodologias reconhecidas e aplicáveis em larga escala, superando as principais limitações observadas na literatura da área, como a ausência de replicabilidade, foco restrito a protótipos acadêmicos e uso de dados simulados.

Como contribuição para a empresa, destaca-se a entrega de uma interface *web* inicial e funcional, que oferece maior flexibilidade do que o *dashboard* atualmente implementado no Grafana. Por ser baseada em *Angular* e conectada a *APIs* externas, a nova interface possibilita integrações mais amplas e adaptáveis, além de abrir caminhos para futuras expansões da solução, como simulações e análises preditivas.

Este trabalho contribuiu não apenas para o avanço técnico da iniciativa Carmen da Amadeus, mas também demonstrou a viabilidade da aplicação de métricas ambientais em larga escala, com potencial para evoluir em direção a soluções cada vez mais completas, transparentes e sustentáveis.

Por se tratar de um tipo de aplicação complexa, muitas vezes abordada exclusivamente em pesquisas acadêmicas e com o uso de simulações, registrou-se algumas limitações ao longo do desenvolvimento prático apresentado. Entre elas, destacam-se divergências ocasionais nos resultados exibidos pela interface, devido a inconsistências de dados ou variações no tempo de resposta da *API*. Além disso, o projeto ainda se encontrava em andamento no momento da finalização deste trabalho, o que impediu a conclusão de etapas como o envio automatizado dos arquivos para o *FinOps* e a validação integral dos dados de armazenamento. Também se observou a dependência da periodicidade e completude dos dados fornecidos pelo *FinOps*, fator que pode impactar a confiabilidade das análises.

Diante dessas limitações, algumas atividades adicionais para este complexo desen-

volvimento são prospectadas:

- **Validação automatizada dos dados recebidos da API**, com tratamento de exceções e alertas em caso de inconsistência;
- **Implementação de testes integrados e unitários mais robustos**, especialmente para os dados de entrada do IF;
- **Desenvolvimento de uma camada de simulação na interface web**, permitindo que usuários explorem diferentes cenários de consumo e configurem variáveis como região, tipo de hardware e quantidade de recursos utilizados;
- **Adoção de estratégias de fallback** caso o envio de dados do *FinOps* falhe, garantindo continuidade mínima das análises.

REFERÊNCIAS

- ABREU, R.; COUTO, M.; CRUZ, L.; CUNHA, J.; FERNANDES, J. P.; PEREIRA, R.; PEREZ, A.; SARAIVA, J. Green software lab: Towards an engineering discipline for green software project's final report. **arXiv**, 2021.
- BASHIR, N.; SHAHRAD, M.; GOHIL, V.; IRWIN, D.; DELIMITROU, C.; BELAVADI, A.; OLIVETTI, E. The sunk carbon fallacy: Rethinking carbon footprint metrics for effective carbon-aware scheduling. **arXiv**, 2024.
- CALERO, C.; PIATTINI, M. (Ed.). **Green in Software Engineering**. Cham: Springer, 2015. ISBN 978-3-319-08580-7. Disponível em: <<https://doi.org/10.1007/978-3-319-08581-4>>.
- CASTOR, F. Estimating the energy footprint of software systems: a primer*. **arXiv**, 2024.
- DANUSHI, O.; FORTI, S.; SOLDANI, J. Carbon-efficient software design and development: A systematic literature review. **ACM Comput. Surv.**, 2025.
- DODGE, J.; PREWITT, T.; COMBES, R. T. des; ODMARK, E.; SCHWARTZ, R.; STRUBELL, E.; LUCCIONI, A. S.; SMITH, N. A.; DECARIO, N.; BUCHANAN, W. Measuring the carbon intensity of ai in cloud instances. **arXiv**, 2022.
- ERDÉLYI, K. Special factors of development of green software supporting eco sustainability. In: **Proceedings of the 11th IEEE International Symposium on Intelligent Systems and Informatics (SISY)**. [S.l.]: IEEE, 2013. p. 337–340.
- Ernst & Young. **Sustainable coding is the need for a greener tomorrow**. 2024. Acesso em: 20 abr. 2025. Disponível em: <https://www.ey.com/en_in/insights/technology/sustainable-coding-is-the-need-for-a-greener-tomorrow>.
- FORTI, S.; BROGI, A. Green application placement in the cloud-iot continuum. **arXiv**, 2021.
- FREITAG, C.; BERNERS-LEE, M.; WIDDICKS, K.; KNOWLES, B.; BLAIR, G.; FRIDAY, A. **The climate impact of ICT: A review of estimates, trends and regulations**. 2021. ArXiv preprint arXiv:2102.02622. Acesso em: 20 abr. 2025. Disponível em: <<https://arxiv.org/abs/2102.02622>>.
- GIL, D.; FERNÁNDEZ-ALEMÁN, J. L.; TRUJILLO, J.; GARCÍA-MATEOS, G.; LUJÁN-MORA, S.; TOVAL, A. The effect of green software: A study of impact factors on the correctness of software. **Sustainability**, v. 10, n. 3471, 2018.
- Green Software Foundation. **Impact Framework - Introduction**. 2025. Acesso em: 20 abr. 2025. Disponível em: <<https://if.greensoftware.foundation/intro>>.
- GULDNER, A.; BENDER, R.; CALERO, C.; FERNANDO, G. S.; FUNKE, M.; GRÖGER, J.; HILTY, L. M.; HÖRNSCHEMEYER, J.; HOFFMANN, G.-D.; JUNGER, D.; KENNES, T.; KRETEN, S.; LAGO, P.; MAI, F.; MALAVOLTA, I.; MURACH, J.; OBERGÖCKER, K.; SCHMIDT, B.; TARARA, A.; VEAUGH-GEISS, J. P. D.; WEBER, S.; WESTING, M.; WOHLGEMUTH, V.; NAUMANN, S. Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components-green software measurement model (gsmm). **Future Generation Computer Systems**, v. 155, p. 402–418, 2024.

GUPTA, U.; KIM, Y. G.; LEE, S.; TSE, J.; LEE, H.-H. S.; WEI, G.-Y.; BROOKS, D.; WU, C.-J. **Chasing Carbon: The Elusive Environmental Footprint of Computing**. 2020. ArXiv preprint arXiv:2011.02839. Acesso em: 20 abr. 2025. Disponível em: <<https://arxiv.org/abs/2011.02839>>.

KAPOOR, S. Green software quality: A comprehensive framework for sustainable metrics in software development. **International Journal of Computer Trends and Technology**, v. 72, n. 10, p. 113–120, 2024.

KENNES, T. **Measuring IT Carbon Footprint: What is the Current Status Actually?** 2023. ArXiv preprint arXiv:2306.10049. Acesso em: 20 abr. 2025. Disponível em: <<https://arxiv.org/abs/2306.10049>>.

KERN, E.; HILTY, L. M.; GULDNER, A.; MAKSIMOV, Y. V.; FILLER, A.; GRÖGER, J.; NAUMANN, S. Sustainable software products-towards assessment criteria for resource and energy efficiency. **Future Generation Computer Systems**, v. 86, p. 199–210, 2018.

KIM, Y. G.; GUPTA, U.; MCCRABB, A.; SON, Y.; BERTACCO, V.; BROOKS, D.; WU, C.-J. Greenscale: Carbon-aware systems for edge computing. **arXiv**, 2023.

LANNELONGUE, L.; GREALEY, J.; INOUE, M. **Green Algorithms: Quantifying the Carbon Footprint of Computation**. 2020. ArXiv preprint arXiv:2007.07610. Acesso em: 20 abr. 2025. Disponível em: <<https://arxiv.org/abs/2007.07610>>.

LI, B.; ROY, R. B.; WANG, D.; SAMSI, S.; GADEPALLY, V.; TIWARI, D. **Toward Sustainable HPC: Carbon Footprint Estimation and Environmental Implications of HPC Systems**. 2023. ArXiv preprint arXiv:2306.13177. Acesso em: 20 abr. 2025. Disponível em: <<https://arxiv.org/abs/2306.13177>>.

LUNARDI, G. L.; FRIO, R. S.; BRUM, M. de M. Tecnologia da informação e sustentabilidade: Levantamento das principais práticas verdes aplicadas à Área de tecnologia. **Gerais: Revista Interinstitucional de Psicologia**, v. 4, n. 2, p. 159–172, 2011.

LUNARDI, G. L.; SIMÕES, R.; FRIO, R. S. Ti verde: Uma análise dos principais benefícios e práticas utilizadas pelas organizações. **Revista Eletrônica de Administração**, v. 77, n. 1, p. 1–30, 2014.

MANUQUIAN, R. B.; BOLIN, R. A. P. Green it: práticas e tecnologias sustentáveis para minimizar o impacto ambiental da tecnologia da informação. Informações de publicação (revista, volume, número, páginas, ano) não encontradas no documento.

MARINI, N.; PAMPALONI, L.; MARTINO, F. D.; VERDECCHIA, R.; VICARIO, E. Green ai: Which programming language consumes the most? **arXiv**, 2024.

MEHRA, R.; SHARMA, V. S.; KAULGUD, V.; PODDER, S.; BURDEN, A. P. Towards a green quotient for software projects. **arXiv**, 2022.

MURUGESAN, S.; GANGADHARAN, G. R. (Ed.). **Harnessing Green IT: Principles and Practices**. United Kingdom: Wiley, 2012. ISBN 978-1-119-97005-7.

PAZIENZA, A.; BASELLI, G.; VINCI, D. C.; TRUSSONI, M. V. A holistic approach to environmentally sustainable computing. **Innovations in Systems and Software Engineering**, v. 20, p. 347–371, 2024.

SARAIVA, J.; ZONG, Z.; PEREIRA, R. Bringing green software to computer science curriculum: Perspectives from researchers and educators. In: **Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21)**. Virtual Event, Germany: ACM, 2021. p. 498–504.

SILVA, J. C. da; BRITO, A. V.; FILHO, G. F. de S. Um estudo de caso sobre a avaliação energética de software como uma contribuição à computação verde. Informações de publicação (revista, volume, número, páginas, ano) não encontradas no documento.

SOUZA, A.; JASORIA, S.; CHAKRABARTY, B.; BRIDGWATER, A.; LUNDBERG, A.; SKOGH, F.; ALI-ELDIN, A.; IRWIN, D.; SHENOY, P. Casper: Carbon-aware scheduling and provisioning for distributed web services. **arXiv preprint arXiv:2403.14792**, 2024.

SUDARSHAN, C. C.; ARORA, A.; CHHABRIA, V. A. Greenfpga: Evaluating fpgas as environmentally sustainable computing solutions. In: **Proceedings of the Design Automation Conference (DAC '24)**. San Francisco, CA, USA: ACM, 2024. p. 1–6.

TAINA, J.; MAKINEN, S. Good, bad, and beautiful software: In search of green software quality factors. **CEPIS Upgrade**, XII, n. 4, p. 22–27, 2011.

THEIS, V.; SCHREIBER, D. Análise reflexiva acerca das contribuições da tecnologia da informação verde para a sustentabilidade corporativa. **Desenvolvimento em Questão**, Editora Unijuí, v. 19, n. 56, p. 264–281, 2021. ISSN 2237-6453.

VARTZIOTIS, T.; DELLATOLAS, I.; DASOULAS, G.; SCHMIDT, M.; SCHNEIDER, F.; HOFFMANN, T.; KOTSOPOULOS, S.; KECKEISEN, M. Learn to code sustainably: An empirical study on llm-based green code generation. **arXiv**, 2024.

WASIF. Green software engineering as a catalyst for renewable energy efficiency and sustainable development. **International Journal of Computer Applications**, v. 177, n. 24, p. 1–6, 2024.

ZHOU, F.; LI, L.; WEN, H. Regional digital infrastructure and carbon neutrality: A technology-structure-efficiency perspective. **Energy Strategy Reviews**, v. 56, p. 101583, 2024.

ANEXO A – DIAGRAMAS DOS PRODUTOS CARMEN

Os diagramas nas figuras 15, 16, 17 fornecem uma visão detalhada da funcionalidade do Carmen, incluindo sua automação no Jenkins e os endpoints do Run Engine e Run Engine Hardware.

A.0.0.1 Carmen Relatando ao Daemon

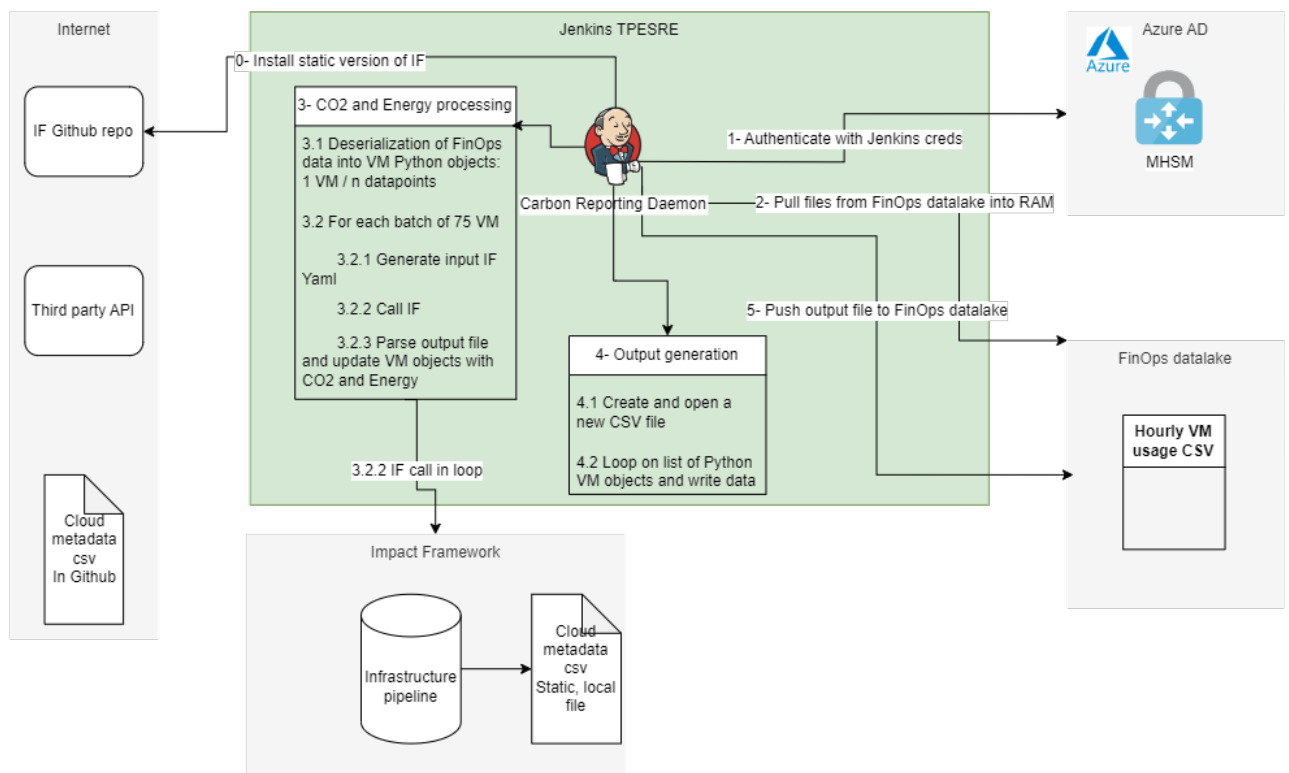


Figura 15 – Carmen Relatando ao Daemon

A.0.0.2 Carmen como um Serviço - Run Engine

A.0.0.3 Carmen como um Serviço - Run Engine Hardware

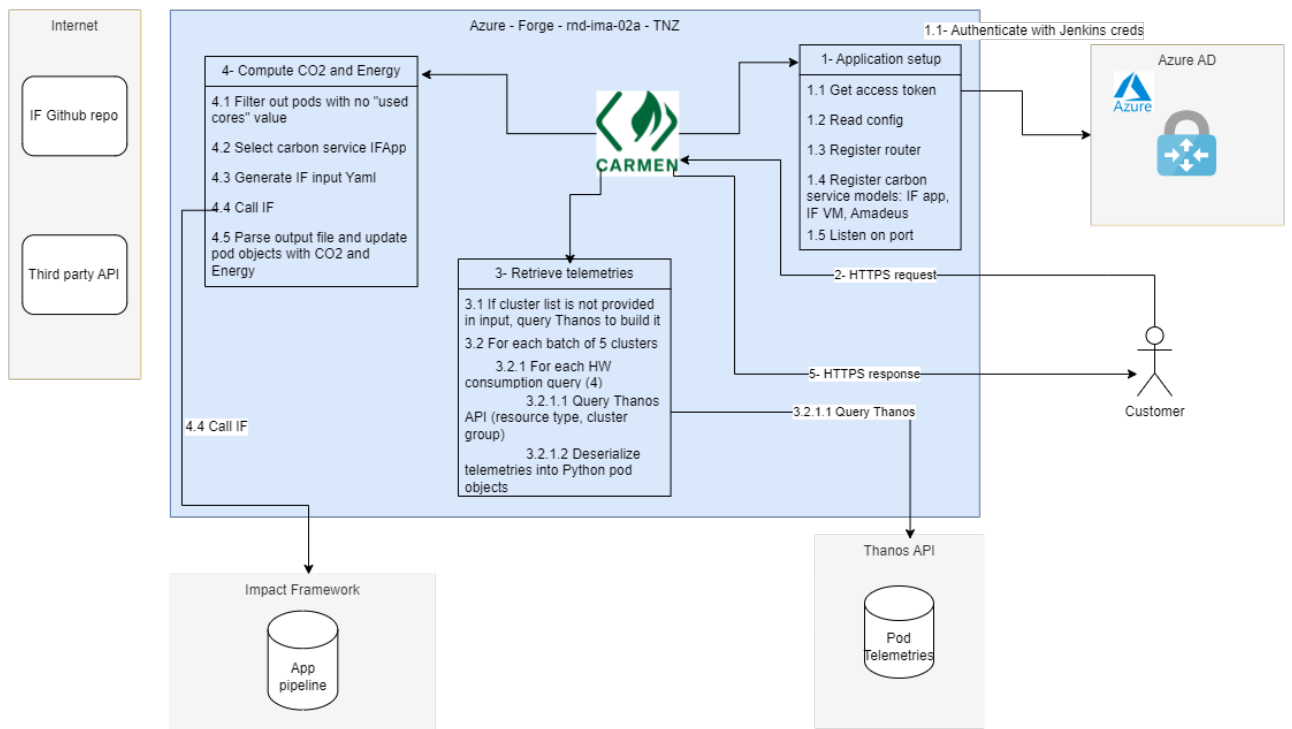


Figura 16 – Carmen como um Serviço - Run Engine

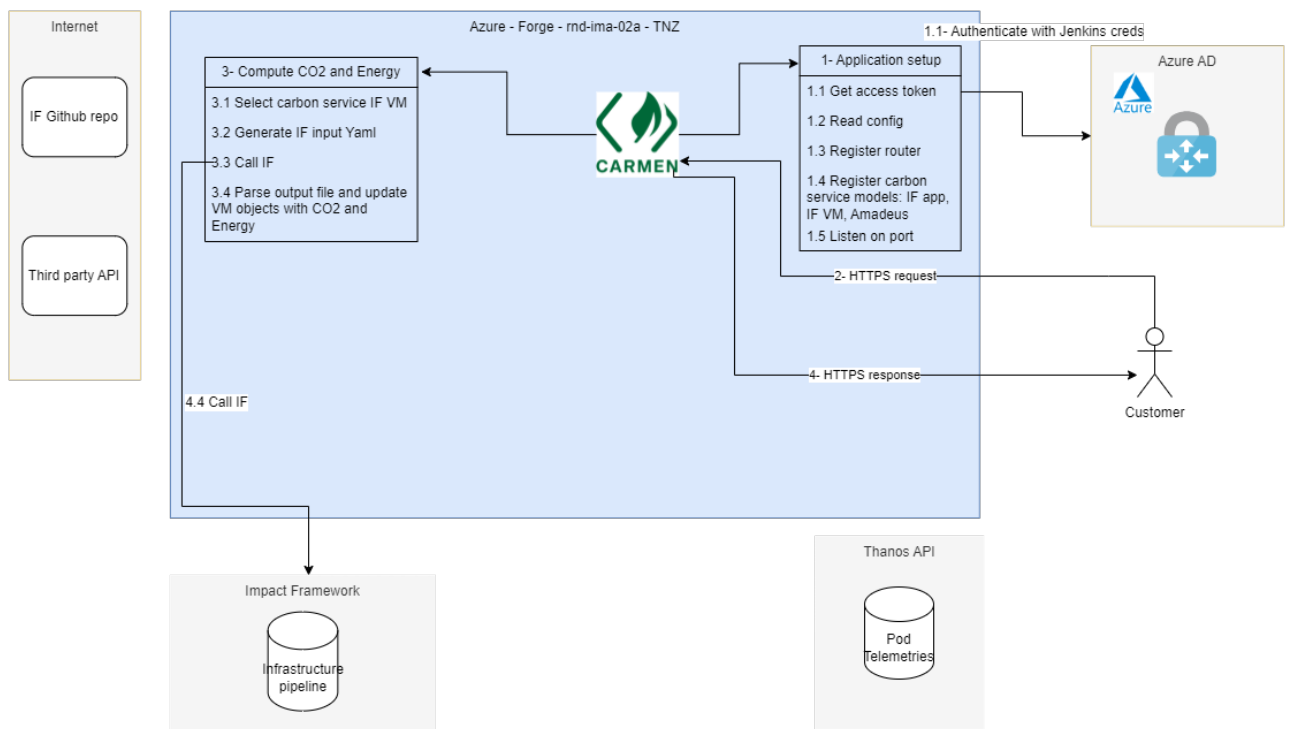


Figura 17 – Carmen como um Serviço - Run Engine Hardware