



# Desenvolvimento de Interface Gráfica para um Simulador de Comunidades de Energia

**JOSÉ CARLOS LEAL OLIVEIRA**

Junho de 2025

# **Desenvolvimento de Interface Gráfica para um Simulador de Comunidades de Energia**

**José Carlos Leal Oliveira**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Luís Lino Ferreira**

**Co-orientador: Tiago Carlos Caló Fonseca**

Porto, junho 2025



# Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

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

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

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

ISEP, Porto, 28 de junho de 2025



# Resumo

As interfaces de utilizador são essenciais para que qualquer aplicação seja intuitiva e acessível aos seus utilizadores. Elas têm como papel principal funcionar como intermediárias entre a lógica interna de um programa e o utilizador final, lidando com obstáculos e garantido que as funcionalidades compreensíveis sejam apresentadas de maneira clara, intuitiva e robusta. A sua inclusão é ainda mais crucial em casos de aplicações como o CityLearn, que permite testar e comparar algoritmos de aprendizagem por reforço para otimização do consumo de energia em ambientes urbanos, que carece de uma UI adequadamente organizada e acessível ao público. Assim será possível proporcionar uma melhor experiência ao utilizador, aumentando a eficiência, a facilidade de uso e a inclusão de audiências mais amplas. A adoção de uma aplicação pode ser comprometida sem uma interface amigável, restringindo o seu potencial e eficácia. Qualquer aplicação que pretenda comunicar ou apresentar algo aos seus utilizadores, seja de qual forma for, deve aproveitar dos benefícios que uma UI providencia.

**Palavras-chave:** Interfaces de utilizador, Acessibilidade, Facilidade de uso



# Abstract

User interfaces are essential for any application to be intuitive and accessible to its users. Their main role is to function as intermediaries between the internal logic of a program and the end user, dealing with obstacles and ensuring that understandable functionalities are displayed in a clear, intuitive and robust manner. Its inclusion is even more crucial in situations of applications such as CityLearn, which allows testing and comparing reinforcement learning algorithms for optimizing energy consumption in urban environments, which lacks a properly organized and accessible UI. This will allow for a better user experience, increasing efficiency, ease of use and the inclusion of a wider audience. The adoption of an application can be compromised without a user-friendly interface, restricting its potential and effectiveness. Any application that wishes to communicate or present something to its users, in any way, must take advantage of the benefits that a UI provides.

**Keywords:** User interfaces, Accessibility, Ease of use



# Agradecimentos

Primeiramente, gostaria de manifestar minha profunda e sincera gratidão ao meu orientador de dissertação, Professor Doutor Luís Miguel Moreira Lino Ferreira, e também ao meu coorientador, Professor Doutor Tiago Carlos Caló Fonseca, por estarem constantemente disponíveis para me oferecer suporte, orientações e conhecimentos valiosos durante todo o processo de elaboração da minha dissertação. Foi uma experiência de aprendizagem fantástica que contribuiu, não só para o meu crescimento pessoal, como também para a minha evolução como engenheiro permitindo-me evoluir em vários aspetos profissionais.

Em segundo lugar, gostaria de agradecer à minha mãe e avó pelo seu amor, cuidado e sacrifícios para me educar, apoiar nesta fase da minha vida e me preparar devidamente para o que o futuro me trará.

Finalmente, quero expressar o meu agradecimento aos meus amigos e colegas do ISEP pela companhia e apoio durante todo o meu percurso letivo. Um especial agradecimento para os meus grandes amigos José Dias, Petra Pisco e Rodrigo Tavares por todos os bons momentos que passámos durante o nosso período académico.



# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização	1
1.1.1	Introdução ao Tema	1
1.1.2	Problema	2
1.2	Solução Proposta	3
1.3	Objetivos	4
1.4	Questões de Investigação	4
1.5	Considerações Éticas	5
1.6	Contribuições	6
1.6.1	Contribuições Sociais	6
1.6.2	Contribuições Científicas	7
1.7	Estrutura da Dissertação	7
<b>2</b>	<b>Planeamento</b>	<b>9</b>
2.1	Competências a desenvolver	9
2.2	Metodologia de Trabalho	10
2.3	Tarefas e Marcos	10
2.4	Riscos Associados	12
<b>3</b>	<b>Estado da Arte</b>	<b>15</b>
3.1	Metodologia de Pesquisa	15
3.2	Conceitos Base	16
3.2.1	Tecnologias de Desenvolvimento Frontend	16
3.2.2	Biblioteca ReCharts	21
3.2.3	Bootstrap	22
3.3	Estado da Arte	24
3.3.1	CityLearn	24
3.3.2	EnergyPlus	27
3.3.3	SustainGym	28
3.3.4	EV2Gym	30
3.3.5	Importância da utilização de UIs em aplicações de simulações	30
<b>4</b>	<b>Análise de Requisitos e Design</b>	<b>33</b>
4.1	Requisitos Funcionais e Não Funcionais	33
4.1.1	Requisitos Funcionais	34
4.1.2	Requisitos Não Funcionais	35
4.2	Modelo C4 + 1	38
4.2.1	Nível 1	39
4.2.2	Nível 2	40

4.2.3	Nível 3 .....	48
4.2.4	Nível 4 .....	54
<b>5</b>	<b>Implementação da Solução .....</b>	<b>55</b>
5.1	Estrutura do projeto .....	55
5.2	Estrutura da interface .....	57
5.2.1	Página “REC Dashboards” .....	58
5.2.2	Página “Use Cases” .....	63
5.2.3	Página “KPIs” .....	64
5.2.4	Página “Datasets” .....	66
5.2.5	Página “Schema” .....	69
5.2.6	Página “Experiment Configs” .....	70
5.2.7	Página “Run Simulations” .....	70
5.2.8	Card Components .....	71
5.3	Implantação da interface .....	82
5.4	Avaliação de Métricas .....	84
5.4.1	WCAG .....	84
5.4.2	WAVE.....	84
5.5	Testes.....	87
5.5.1	Testes E2E (Cypress).....	87
5.5.2	Descrição dos Testes.....	88
<b>6</b>	<b>Considerações Finais e Trabalho Futuro .....</b>	<b>91</b>
6.1	Objetivos Concretizados .....	91
6.2	Trabalho Futuro .....	92



# Lista de Figuras

Figura 1 - Resultados de um edifício em vista de tabela .....	3
Figura 2 - Dados de um edifício em vista de gráficos .....	3
Figura 3 - Gráfico de Gant com as tarefas definidas .....	11
Figura 4 - Fluxo do processo de trabalho nos componentes da UI .....	11
Figura 5 - Análise comparativa do tamanho de pacotes de compilação (Diniz et al., 2022) .....	19
Figura 6 - Análise comparativa do Tpl (Diniz et al., 2022) .....	20
Figura 7 - Análise comparativa do tempo de manipulação do DOM (Diniz et al., 2022) .....	20
Figura 8 - Gráficos da biblioteca Recharts, imagem obtida em (Ably, 2023) .....	22
Figura 9 - Sistema de grelha do Bootstrap (Jake Spurlock, 2013) .....	23
Figura 10 - Resultados do CityLearn Challenge 2020, (Vázquez-Canteli et al., 2020) .....	25
Figura 11 - Resultados do CityLearn Challenge 2021 (Nagy et al., 2021) .....	26
Figura 12 - Resultados do CityLearn Challenge 2023 (Garmendia et al., n.d.) .....	26
Figura 13 - Interface do EnergyPlus (Crawley et al., n.d.) .....	28
Figura 14 - Resultados experimentais dos 5 ambientes RL (Yeh et al., n.d.) .....	29
Figura 15 - Processo de simulação no EV2Gym (Orfanoudakis et al., 2024) .....	30
Figura 16 - Casos de uso .....	34
Figura 17 - Vista Lógica - Nível 1 .....	39
Figura 18 - Vista Lógica - Nível 2 .....	40
Figura 19 - Vista Física - Nível 2 .....	41
Figura 20 - SSD: Visualização de dados das simulações das comunidades de energia .....	42
Figura 21 - SSD: Visualização dos KPIs das simulações e comparação dos mesmos .....	43
Figura 22 - SSD: Criação de um ficheiro de configuração do algoritmo .....	44
Figura 23 - SSD: Criação de <i>dataset</i> e ficheiro de configuração de treino .....	45
Figura 24 - SSD: Execução das simulações com base em <i>schemas</i> , configurações e <i>datasets</i> ..	45
Figura 25 - SSD: Configuração da comunidade de energia .....	46
Figura 26 - SSD: Visualização de dados reais em cenários diferentes .....	47
Figura 27 - SSD: Personalização da vista dos dados .....	47
Figura 28 - Vista Lógica - Nível 3 .....	48
Figura 29 - Estrutura inicial para a representação de gráficos .....	49
Figura 30 - Estrutura decidida para a representação de gráficos .....	50
Figura 31 - Estrutura para seleção de observações e ações .....	51
Figura 32 - SD: Execução das simulações com base em <i>schemas</i> , configurações e <i>datasets</i> ....	51
Figura 33 - Componente para array de pares .....	52
Figura 34 - SD: Visualização de dados reais em cenários diferentes .....	52
Figura 35 - Gráfico com <i>slider</i> .....	53
Figura 36 - Gráfico com o componente Brush .....	54
Figura 37 - Filtro de intervalos .....	54
Figura 38 - Estrutura do projeto .....	56
Figura 39 - Estrutura base da interface .....	58
Figura 40 - Botão "Upload Simulations" .....	58

Figura 41 - Estrutura de pastas das simulações .....	59
Figura 42 - Lista de ficheiros com os dados da simulação .....	59
Figura 43 - Aviso do browser no upload de ficheiros.....	60
Figura 44 - Janela com o botão "Select Simulations".....	60
Figura 45 - Janela de seleção de simulações visíveis .....	60
Figura 46 - Detalhes da simulação .....	61
Figura 47 - Gráficos adicionais das simulações.....	62
Figura 48 - Divisão dos dados em episódios .....	63
Figura 49 - Estrutura dos dados na base de dados MongoDB .....	63
Figura 50 - Visualização dos dados dos cenários .....	64
Figura 51 - Visualização dos KPIs de uma simulação .....	65
Figura 52 - Janela de seleção da simulação a comparar .....	65
Figura 53 - Separador com os resultados da comparação .....	66
Figura 54 - Formulário de preenchimento dos dados do <i>dataset</i> .....	67
Figura 55 - Secção dos dados do agente e função de recompensa .....	67
Figura 56 - Caixa de texto de pré-visualização do JSON .....	68
Figura 57 - Campos do período e data e hora inicial e final.....	68
Figura 58 - <i>Canvas</i> com componentes e as suas ligações.....	69
Figura 59 - Formulário do ficheiro de configuração.....	70
Figura 60 - Página "Run Simulations".....	71
Figura 61 - Mensagem de sucesso .....	71
Figura 62 - Representação dos dados com intervalo de 10 minutos.....	73
Figura 63 - Representação dos dados com intervalo de 40 minutos.....	73
Figura 64 - Exemplo do filtro de visualização de campos .....	75
Figura 65 - Exemplo do filtro da data inicial e final.....	77
Figura 66 – Separação de valores positivos e negativos.....	78
Figura 67 - Gráfico de produção.....	79
Figura 68 - Gráfico de consumo .....	79
Figura 69 - Gráfico dos carregadores .....	80
Figura 70 - Gráfico das baterias .....	81
Figura 71 - Gráfico dos veículos elétricos .....	82
Figura 72 - Gráfico dos preços .....	82
Figura 73 - Interface do puTTY .....	83
Figura 74 - Fluxo do processo de atualização da imagem no Docker .....	83
Figura 75 - Extensão do Chrome para uso do WAVE .....	85
Figura 76 - Resultados da análise da página .....	86
Figura 77 - Lista de testes por página .....	88
Figura 78 - Estrutura da execução dos testes .....	89
Figura 79 - Página da vista em mapa .....	92



# Lista de Tabelas

Tabela 1 – Questões de Investigação.....	5
Tabela 2 - Tarefas Principais.....	10
Tabela 3 - Principais Marcos .....	12
Tabela 4 - Riscos associados ao projeto.....	12
Tabela 5 - Lista de <i>queries</i> utilizadas.....	16
Tabela 6 - Requisitos Funcionais .....	35
Tabela 7 - Requisitos de funcionalidade .....	36
Tabela 8 - Requisitos de usabilidade.....	36
Tabela 9 - Requisitos de confiabilidade .....	36
Tabela 10 - Requisitos de desempenho .....	37
Tabela 11 - Requisitos de suportabilidade.....	37
Tabela 12 - Requisitos de design.....	38
Tabela 13 - Requisitos de implementação.....	38



# Lista de Excertos de Código

Excerto de Código 1 - Esqueleto geral de um gráfico .....	72
Excerto de Código 2 - Método para determinar intervalo base .....	73
Excerto de Código 3 - Estrutura do filtro de intervalos.....	74
Excerto de Código 4 - Lógica de soma/média de valores por intervalo.....	74
Excerto de Código 5 - Inicialização dos campos visíveis .....	75
Excerto de Código 6 - <i>Checkboxes</i> com a visibilidade dos campos.....	76
Excerto de Código 7 - Controlo da visibilidade dos elementos do gráfico .....	76
Excerto de Código 8 - Componente <i>DateRangeSlider</i> .....	77
Excerto de Código 9 - Método para filtrar dados com base no valor do slider .....	78
Excerto de Código 10 - Mensagem de partida/chegada de um veículo elétrico .....	80
Excerto de Código 11 - Múltiplos eixos Y .....	81



# Acrónimos e Símbolos

## Lista de Acrónimos

<b>CC</b>	Contribuição Científica
<b>CE</b>	Comunidade de Energia
<b>CEI</b>	Comunidades de Energia Inteligentes
<b>CORS</b>	Cross-Origin Resource Sharing
<b>CS</b>	Contribuição Social
<b>CSS</b>	Cascading Style Sheets
<b>DOM</b>	Document Object Model
<b>E2E</b>	End-To-End
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IA</b>	Inteligência Artificial
<b>IoT</b>	Internet of Things
<b>ISEP</b>	Instituto Superior de Engenharia do Porto
<b>JSX</b>	JavaScript XML
<b>KPI</b>	Key Performance Indicator
<b>MVC</b>	Model-View-Controller
<b>ODS</b>	Objetivos de Desenvolvimento Sustentável
<b>OPEVA</b>	Optimization of Electric Vehicle Autonomy
<b>PREPD</b>	Preparação para a dissertação
<b>REC</b>	Renewable Energy Communities
<b>RL</b>	Reinforced Learning
<b>RQ</b>	Research Question
<b>SFC</b>	Single File Components

<b>SOC</b>	State Of Charge
<b>SoftCPS</b>	Software Technologies for Cyber-Physical Systems
<b>SPA</b>	Single Page Application
<b>SSD</b>	System Sequence Diagram
<b>UI</b>	User Interface
<b>URL</b>	Uniform Resource Locator
<b>VE</b>	Veículos Eléctricos
<b>WCAG</b>	Web Content Accessibility Diretriz
<b>XML</b>	Extensible Markup Language

# 1 Introdução

Este capítulo inicia-se com uma visão geral da dissertação, oferecendo uma contextualização do tema da dissertação e depois a descrição do problema que impulsionou o desenvolvimento deste trabalho. Em seguida, amplia progressivamente o foco do mesmo, oferecendo uma visão da solução proposta, além de expor as questões de investigação desenvolvidas, objetivos e contribuições. Esta metodologia, que transita de uma perspectiva mais ampla para um aprofundamento minucioso, é reiterada ao longo do documento, com o propósito de orientar os leitores de forma clara e fluida através das decisões de design tomadas.

## 1.1 Contextualização

### 1.1.1 Introdução ao Tema

Com o progresso da tecnologia e a ampla digitalização de processos, as aplicações de software tornaram-se recursos essenciais em várias áreas, desde o setor corporativo até o uso individual. Entretanto, conforme as funcionalidades desses sistemas se tornam mais complexas, aparece a necessidade de desenvolver modos de interação que sejam acessíveis e fáceis de usar para os utilizadores. Isso torna-se ainda mais evidente em situações de aplicações cujo foco principal é a apresentação de dados estatísticos, seja na forma de gráficos, tabelas, mapas, etc.

É nesse contexto que as UIs têm uma função essencial pois estas são uma forma de ligação entre o objeto e o produto de software. Estas permitem fazer a ligação entre a interação humana e do computador, onde os utilizadores interagem com computadores ou máquinas para completar tarefas (Pratama & Cahyadi, 2020). Em diversas situações, programas criados inicialmente sem uma interface acessível ou sem qualquer interface podem encontrar obstáculos na aceitação e na usabilidade, o que enfatiza a necessidade de implementar uma UI bem elaborada para aprimorar a experiência do utilizador, aumentar o valor do software e, num

sentido mais orientado para a vertente de marketing do produto, para facilitar a explicação e venda de resultados a diferentes tipos de *stakeholders* do projeto.

Um problema que existe nos simuladores e cada vez mais tem impacto nos mesmos e nas comunidades de energia, é a falta de uma UI bem definida, prejudicando assim a utilidade destes no contexto do mundo real no qual, cada vez mais, é imperativa a existência de uma forma de interagir com os utilizadores de forma visual que permita ilustrar e elucidar quem utiliza este tipo de aplicações. O simulador no qual esta dissertação se enquadra, o CityLearn, é um destes casos.

### **1.1.2 Problema**

No contexto da transição para fontes de energia renováveis e da criação de CEIs, a visualização clara e intuitiva das simulações desempenha um papel central. As redes elétricas modernas cada vez mais tornam-se incrementalmente complexas, integrando uma gama diversificada de fontes de energia, como solar, eólica e armazenamento em baterias. Além disso, estas redes precisam de lidar com variáveis dinâmicas, como o consumo em tempo real, as condições climáticas e as decisões de controlo baseadas em algoritmos avançados de otimização. Normalmente são utilizados simuladores, para preparar e estudar estas otimizações antes de irem para o mercado, no entanto, tal como indicado na secção 1.1.1, estas muitas vezes carecem de uma UI que permita ter uma visualização clara dos resultados. Para que os operadores de rede, desenvolvedores e membros das comunidades possam compreender essas dinâmicas complexas, a criação de uma UI eficaz é fundamental.

Essa interface deve ser capaz de representar visualmente os ativos elétricos (por exemplo, painéis solares, baterias, inversores e consumidores de energia), permitindo que os utilizadores consigam ter uma visão geral dos componentes que formam a rede. A representação gráfica do fluxo de energia é igualmente importante, uma vez que torna mais fácil visualizar como a energia é gerada, distribuída e consumida em diferentes partes da rede. A capacidade de apresentar ações aplicadas em tempo real, como o consumo de energia de uma bateria ou equipamento elétrico, permite que os operadores de rede ajustem as suas decisões com base no feedback visual e permite a quem desenvolve, avaliar se está a funcionar de acordo com o esperado e assim manter/ganhar a confiança dos *stakeholders*.

Uma interface intuitiva pode ser um fator decisivo para tornar acessível a análise dos dados obtidos para otimizar o funcionamento das redes elétricas inteligentes. Algoritmos de otimização, como aqueles que determinam a melhor alocação de energia em horários de pico ou que minimizam as perdas, geram grandes quantidades de dados. A UI deve ser capaz de simplificar essas informações complexas num formato visual que permita uma rápida avaliação do desempenho e dos resultados.

Outro ponto importante a ter em conta é que o CityLearn é uma aplicação que mostra diversos tipos de informações distintas, desde dados sobre o consumo de equipamentos eletrónicos num determinado edifício, até comparações de gastos ao longo de variados períodos de tempo,

entre horas, dias ou até mesmo meses de diferença. É por isso importante a existência de uma forma gráfica de ilustrar os diversos tipos de informações a apresentar ao utilizador e formas de efetuar uma pesquisa de forma mais avançada, entre os intervalos de tempo desejados.

Em suma, é essencial a criação de uma UI robusta, que apresente claramente os ativos elétricos, o fluxo de energia, as ações dos algoritmos em tempo real e que permita analisar ao pormenor em diferentes instâncias temporais, para garantir que as dinâmicas complexas das redes inteligentes sejam compreensíveis para operadores e utilizadores. Isso não só facilita o monitoramento e a gestão eficiente da rede, como também fortalece a confiança nas tecnologias de energia renovável e promove uma transição mais suave para essas novas formas de energia.

## 1.2 Solução Proposta

Para o desenvolvimento de uma solução para o problema acima definido, o estudante escolheu trabalhar sobre o CityLearn, pois é um dos casos em que não possui uma UI desenvolvida e por isso pode servir como uma boa base para outros simuladores adotarem, componentes semelhantes nas suas UIs. Atualmente, a UI do CityLearn é inexistente apresentando principalmente os dados fornecidos através de uma vista em consola, tendo alguns casos em que são apresentados alguns gráficos básicos como mostram as Figuras 1 e 2. Os dados para estas imagem foram obtidos através de um tutorial disponíveis na documentação do CityLearn. (Kingsley Nweye, s.d.)

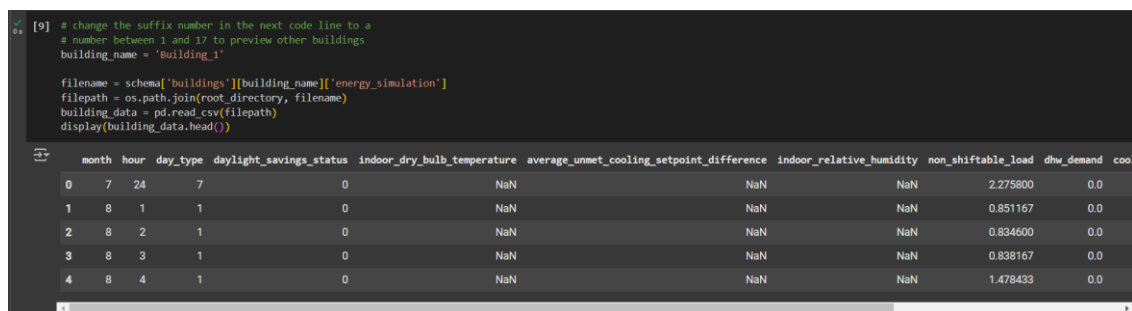


Figura 1 - Resultados de um edifício em vista de tabela

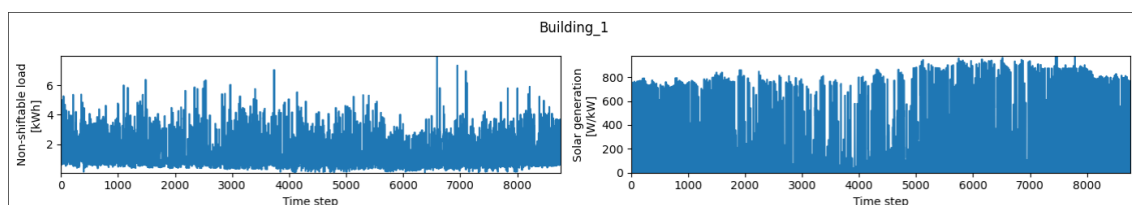


Figura 2 - Dados de um edifício em vista de gráficos

De forma a conseguir elaborar uma solução que permita garantir a visualização clara e intuitiva das simulações fornecidas pelo CityLearn, esta dissertação pretende apresentar uma UI

desenvolvida utilizando React com a adição de Bootstrap nos seus elementos visuais, fornecendo assim um formato simples de entender e facilmente navegável.

Para aprimorar a interação dos utilizadores também é pretendido conseguir fornecer um grau de complexidade de pesquisa adaptável aos desejos do utilizador, ou seja, a aplicação oferece a possibilidade ao utilizador de ser capaz de ver o tipo de informações que deseja sejam estas sob a forma de vários tipos de gráficos generalizados, sob a forma de KPIs, detalhando todos os consumos dos diferentes equipamentos presentes nas casas ou, numa vista mais real e dinâmica do sistema, utilizando a ajuda de um mapa onde é possível ver as casas e os equipamentos instalados em cada uma delas, com detalhes acerca dos consumos energéticos de cada um.

### 1.3 Objetivos

Esta dissertação tem como objetivo o desenvolvimento de uma interface gráfica apelativa e funcional para a visualização de simulações em comunidades de energia renovável tendo como especial foco o CityLearn. Os objetivos específicos incluem:

- **O1** - Uma análise de requisitos onde serão identificadas as necessidades para a interface em termos de funcionalidade, usabilidade e design, através da análise das especificações das simulações.
- **O2** - A criação do esqueleto inicial da UI englobando a definição de rotas, a hierarquia de componentes e a configuração do ambiente. Nesta fase, são reconhecidos os módulos principais e os fluxos de dados, garantindo que a arquitetura permaneça modular e escalável. Essa organização inicial atua como fundamento para o posterior desenvolvimento das páginas e funções específicas.
- **O3** - Definição do design e finalmente a implementação da UI de cada página da aplicação de forma a permitir uma visualização clara e intuitiva dos ativos elétricos, das ações aplicadas, das observações e do fluxo de energia.
- **O4** - Testes gerais do fluxo analisando a integração e a operação de todas as páginas e elementos dentro da aplicação. Serão feitos cenários reais de uso para validar o comportamento esperado, incluindo a navegação entre páginas, a resposta às interações do utilizador e a correta apresentação de dados.

Devido ao cariz da aplicação do CityLearn, baseado em simulações e obtenção de dados estatísticos, o design deve incorporar elementos gráficos e interativos que facilitem a interpretação dos dados, melhorar as formas de pesquisa e filtragem dos mesmos e por fim, uma melhoria no processo de interação dos utilizadores com a aplicação.

### 1.4 Questões de Investigação

Conforme os objetivos do projeto, propõe-se a exploração de técnicas de forma a poder criar uma UI para o CityLearn com o intuito de avaliar o efeito que a adoção dessa interface pode ter

não só, na forma como os utilizadores podem interagir com a ferramenta, como também no potencial de evolução da mesma. Para avaliar a aplicabilidade das linguagens e ferramentas na criação da UI e os seus efeitos na interação com os utilizadores, foram formuladas uma série de questões de investigação que servem como diretrizes para a aplicação da metodologia de pesquisa escolhida pelo estudante.

Na Tabela 1 é possível ver a lista de questões formuladas:

Tabela 1 – Questões de Investigação

Identificador	Questão Investigação
QI1	Quais são as ferramentas para desenvolvimento em <i>frontend</i> mais adequadas para usar como base para criar uma UI para um aplicação baseada em simulações?
QI2	Quais são os componentes de UI mais adequados para mostrar grupos complexos de dados de forma dinâmica e intuitiva?
QI3	Quais são os benefícios que uma UI pode trazer para uma aplicação baseada em simulações?

## 1.5 Considerações Éticas

No âmbito ético deste trabalho, como estudante, foi utilizado o código de conduta do IPP. (Diário Da República, 2.a Série PARTE E Artigo 2.o, n.d.)

As seguintes condutas foram respeitadas durante o desenvolvimento deste projeto: o artigo 6º, ponto 2.8, menciona o uso de ideias, frases, parágrafos ou textos sem citar ou referenciar corretamente as fontes (Diário Da República, 2.a Série PARTE E Artigo 2.o, n.d.). O ponto 2.9 refere-se à apresentação de um trabalho como original que já foi exibido ou publicado em outra ocasião sem o devido reconhecimento. O ponto 2.10 refere-se à apresentação de obras de terceiros, levando a um descumprimento autorizado, enquanto o item 2.11 discute a utilização de trabalhos com resultados fraudulentos ou interpretados de maneira tendenciosa.

O artigo 8º aparece no começo do documento, que consiste na declaração de compromisso e o artigo 10º foi seguido em pontos como: 1 a) analisar e documentar uma investigação de forma cuidadosa e ponderada; e) citar de maneira apropriada e abrangente os trabalhos relacionados; e g) apresentar os resultados de forma consistente, assegurando que as conclusões sejam verificáveis e reproduzíveis.

Como aluno de mestrado em Engenharia Informática, na área de Engenharia de Software, igualmente importante ter em atenção o código de conduta do IEEE (Gotterbarn et al., n.d.). Certos princípios do IEEE, como a honestidade, integridade e a preservação da transparência e veracidade durante este trabalho, junto à competência profissional, asseguram a máxima qualidade do trabalho realizado. A evitação de conflitos de interesse, como possíveis vieses ou interesses pessoais que possam prejudicar a integridade das decisões, também foi levada em conta. Outros princípios, como "assegurar que os objetivos e metas sejam apropriados e

alcançáveis" e "identificar, definir e tratar questões éticas, econômicas, culturais, legais e ambientais ligadas ao trabalho", foram discutidos com o orientador para estabelecer metas viáveis e empregar projetos de código aberto ou de uso livre.

No contexto do projeto do CityLearn, os dados utilizados para fornecer informações ao simulador e assim conseguir obter com a melhor precisão possível os dados obtidos das simulações, são fornecidos por alguns dos *stakeholders*. Há por isso, o cuidado de garantir que estes dados são anonimizados, dando assim total privacidade a quem forneceu os dados das suas casas e veículos para alimentar o simulador com dados.

## **1.6 Contribuições**

Para o âmbito das contribuições que esta dissertação fornece, foram reconhecidas e descritas duas vertentes principais: as contribuições sociais, que visam aprimorar a experiência dos diferentes utilizadores e alinhar-se com metas globais de sustentabilidade; e as contribuições científicas, que se inserem num esforço constante de inovação tecnológica no contexto dos sistemas ciber-físicos, com ênfase na otimização e acessibilidade de ferramentas de simulação.

### **1.6.1 Contribuições Sociais**

Este projeto tem como grande objetivo beneficiar todos os utilizadores que utilizem a UI como por exemplo, os desenvolvedores do CityLearn, os *stakeholders* ou até mesmo os gestores das REC, melhorando a sua experiência na aplicação. Isto está alinhado com vários acordos e iniciativas políticas, como os números 7, 11 e 13 dos Objetivos de Desenvolvimento Sustentável (ODS) das Nações Unidas (United Nations, Department of Economic and Social Affairs, s.d.).

Ao abordar estes objetivos, o trabalho descrito nesta dissertação pode contribuir para um impacto positivo na forma como os utilizadores podem interagir com o sistema e trabalhar com os dados existentes na mesma.

As contribuições sociais desta dissertação podem ser resumidas em:

**CS1** – Melhorar a experiência dos utilizadores do CityLearn

**CS2** – Aumentar a adoção e aceitação de Energias Renováveis e VEs

**CS3** – Facilitar o desenvolvimento e o teste de novas soluções, tanto no CityLearn como em outros simuladores

**CS4** – Melhorar a visualização e a forma de analisar os dados das simulações

**CS5** – Facilita o processo para efetuar, iniciar e gerir simulações

### 1.6.2 Contribuições Científicas

Este projeto foi desenvolvido no laboratório de investigação Software Technologies for Cyber Physical Systems (SoftCPS), vinculado ao Instituto Superior de Engenharia do Politécnico do Porto (ISEP) e integrante do grupo Human-Centered Computing and Information Science (HumanISE) do INESC TEC. (SoftCPS, s.d.)

O SoftCPS visa criar soluções de software novas e inovadoras para sistemas ciber-físicos, com ênfase especial em *middleware*, IoT, *Edge Computing* e computações paralelas/distribuídas. O trabalho realizado nesta dissertação está integrado no âmbito do projeto OPEVA (OPEVA, s.d.), que é um projeto de investigação e desenvolvimento financiado pela União Europeia, com o objetivo de otimizar a autonomia dos veículos elétricos. O SoftCPS e esta dissertação têm como meta principal aprimorar a forma como os utilizadores do CityLearn interagem com o sistema, trazendo mais valor para a ferramenta, possibilitando e facilitando o trabalho de outros investigadores e desenvolvedores na área.

As contribuições científicas desta dissertação podem ser resumidas em:

**CC1** - Uma análise nos conceitos base e benefícios do uso de diferentes linguagens de desenvolvimento frontend para a criação da UI para o CityLearn

**CC2** - Um estudo comparativo sobre as linguagens de desenvolvimento frontend e impacto em aplicações baseadas em simulações

**CC3** - Uma análise das aplicações com um propósito similar ao CityLearn e os benefícios da adoção de uma UI para as mesmas

**CC4** - Design e implementação de uma UI para o CityLearn, para visualização dos resultados das simulações, facilitando o ciclo de desenvolvimento

**CC5** - Entregáveis, implementação e resultados para projetos de investigação OPEVA

**CC6** – Um conjunto de ações que permite a configuração e execução de novas simulações

## 1.7 Estrutura da Dissertação

Este documento está estruturado em três partes principais: a Introdução, o Planeamento e Estado da Arte. Inicialmente, no capítulo da Introdução (Capítulo 1), é apresentada uma contextualização e introdução ao problema no qual se centra este projeto, visando familiarizar o leitor com o estado atual do tema em questão. Além disso, a introdução inclui uma descrição da solução proposta a declaração do problema, os objetivos propostos, as questões de investigações formuladas e finalmente as considerações éticas e contributos.

No capítulo do planeamento (Capítulo 2), são apresentadas as competências a ser desenvolvidas pelo estudante durante o desenvolvimento da dissertação, um cronograma com os principais marcos e tarefas a cumprir e por fim o conjunto dos riscos associados ao

desenvolvimento do projeto, podendo assim fornecer aos leitores uma ideia do processo de planeamento adotado pelo estudar para organização das tarefas e consequente execução das mesmas ao longo do tempo.

Finalmente, a seção do Estado da Arte (Capítulo 3) é descrito o método de pesquisa adotado pelo aluno e feita uma contextualização sobre o estado atual do domínio explicando os conceitos base envolventes no projeto e fazendo uma análise a outras ferramentas similares ao CityLearn tirando algumas conclusões sobre a importância da implementação de uma UI para este tipo de aplicações. Este capítulo pretende servir também como resposta às questões de investigação acima delineadas.

## 2 Planeamento

Para este capítulo são descritas as competências do estudante, tanto positivas como negativas e o seu impacto no projeto e as estratégias a desenvolver para trabalhar sobre as mais negativas de forma a não comprometer a integridade da interface. É feita também uma descrição da metodologia de trabalho adotada e é apresentada uma lista das tarefas a efetuar com uma estimativa temporal das mesmas, o gráfico de Gantt respetivo e os riscos associados ao projeto em geral.

### 2.1 Competências a desenvolver

Para atingir todos os objetivos propostos no desenvolvimento do projeto, foi feito um estudo das competências atuais do estudante, de forma a determinar os pontos fortes e os pontos a melhorar. Dos pontos fortes destacam-se a boa gestão de tempo, demonstrada em outros projetos feitos pelo estudante e o trabalho em equipa e colaboração, uma característica importante considerando que o projeto do CityLearn está dividido em diversos aspetos exigindo assim, comunicação com os outros membros de forma a trocar ideias e desenvolver estratégias para a melhoria do próprio trabalho do estudante.

Alguns dos pontos que se consideraram mais fracos e, por consequência, os mais importantes a desenvolver para o sucesso do projeto foram a proatividade, o pensamento crítico e a autodisciplina, pois o tempo é limitado e devido a isso, é importante ter a capacidade de tirar o maior proveito do mesmo para desenvolver o máximo possível, no entanto, sem comprometer a qualidade do produto final, considerando que este será apresentado aos membros do CityLearn e pode vir a ser utilizado pelos mesmos no quotidiano, exigindo assim um nível mínimo de excelência na interface a desenvolver.

De forma a poder aprimorar os pontos acima descritos, o estudante comprometeu-se a fazer reuniões semanais com o grupo de investigação SoftCPS, no qual está integrado e também, no

seu tempo livre a fazer sessões dedicadas à escrita do relatório ou ao desenvolvimento do projeto em si. Também se propôs a fazer uma entrega intermédia do estado atual do relatório de forma a controlar o trabalho desenvolvido e a facilitar o trabalho dos orientadores, dando assim mais tempo para rever o relatório e, por consequência, dar mais tempo ao estudante para efetuar as correções necessárias.

## 2.2 Metodologia de Trabalho

Devido ao carácter do projeto a ser desenvolvido, no qual é necessário monitorizar o estado atual da interface e melhorar ou alterar o mesmo consoante as necessidades do CityLearn e dos seus utilizadores, a metodologia de trabalho que foi considerada a mais adequada para ser adotada foi a *Action Research*, na qual se baseia em processos cíclicos ao longo do tempo, havendo uma apresentação e avaliação do progresso periódica do projeto aos orientadores e *stakeholders*.

Algo que em grande parte facilitou este processo e permitiu uma constante monitorização do estado da interface a ser desenvolvida, foi o facto de o estudante estar integrado no laboratório de investigação do ISEP, SoftCPS, permitindo ter semanalmente reuniões com os membros para discussão do projeto e futuras adições ao mesmo. Nessas reuniões era feita uma pequena demonstração da interface aos membros do grupo, incluindo os orientadores e, posteriormente, era efetuada uma análise aos componentes da UI apresentados. Dessa análise poderiam surgir questões sobre outras alternativas possíveis ao componente usado ou adições de novos dependendo da funcionalidade em questão.

## 2.3 Tarefas e Marcos

Para o desenvolvimento do projeto foi definido um conjunto de tarefas de forma a listar as ações a tomar para o cumprir com os objetivos do projeto propostos e também para o desenvolvimento das competências do aluno.

Na Tabela 2, é possível ver a lista das tarefas delineadas pelo estudante:

Tabela 2 - Tarefas Principais

Tarefa	Duração	Data Inicio
Definição da metodologia	4 dias	01/10/2024
Definição das competências a desenvolver	6 dias	01/10/2024
Definição da linguagem a utilizar	4 dias	01/10/2024
Definição das Research Questions	14 dias	08/10/2024
Descrever os conceitos base	14 dias	01/11/2024
Analisar ferramentas semelhantes ao CityLearn	14 dias	15/11/2024
Criação da estrutura do menu lateral	7 dias	14/10/2024

Criação da estrutura da página dos KPIs	6 meses	01/11/2024
Criação da estrutura da página dos <i>dashboards</i>	6 meses	01/11/2024
Criação da estrutura da página da vista em mapa	6 meses	01/11/2024
Integração com o <i>backend</i>	1 mês	01/03/2025
Escrita do relatório da dissertação	9 meses	01/10/2024

Com base nas tarefas e durações acima descritas, foi também criado um gráfico de Gantt para uma melhor visualização do fluxo de trabalho do estudante e das precedências existentes entre algumas das tarefas. Este gráfico está apresentado na Figura 3.

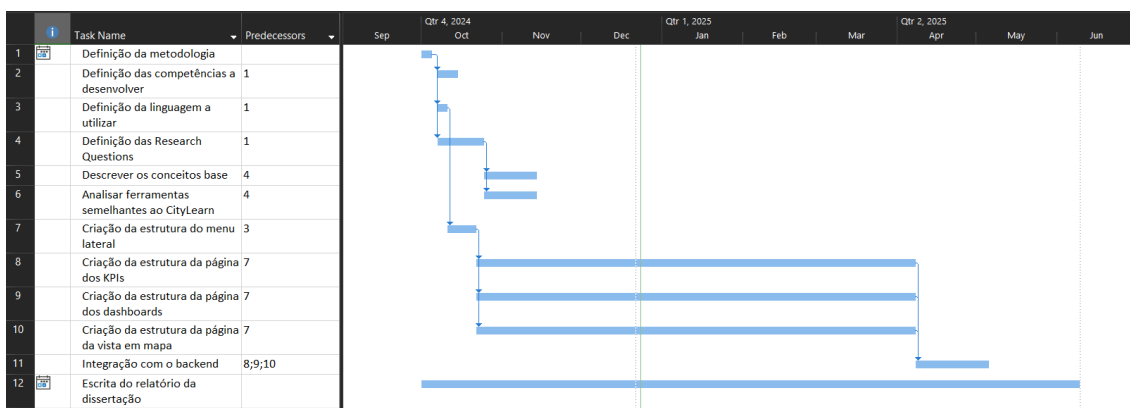


Figura 3 - Gráfico de Gantt com as tarefas definidas

Devido ao cariz volátil do processo de design e implementação de interfaces, as estimativas feitas em termos de duração estão a considerar todo o processo de discussão e alterações que possam vir, não só por decisão dos colaboradores da CityLearn, mas também por decisão do grupo de investigação ou até mesmo do próprio estudante.

Algo também importante a denotar acerca das tarefas acima declaradas, é que o processo de desenvolvimento da UI é um processo cíclico, por consequência da metodologia adota pelo estudante. Isto significa que para cada um dos processos de desenvolvimento da UI, houve sempre um conjunto de pequenas tarefas associadas que aconteciam semanalmente. Este processo é demonstrado pelo diagrama desenhado na Figura 4.

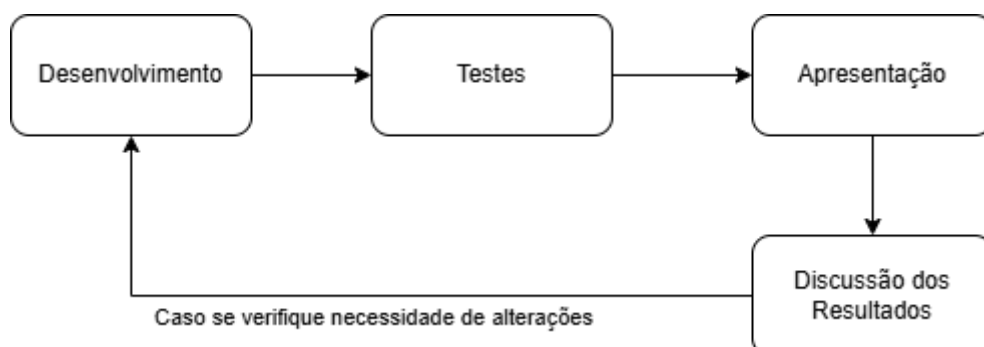


Figura 4 - Fluxo do processo de trabalho nos componentes da UI

Para além das tarefas, foram definidos alguns marcos, tanto obrigatórios, no contexto da cadeira de PREPD e da dissertação, como alguns opcionais definidos pelo estudante e orientadores com o objetivo de poder gerir melhor e controlar o trabalho feito ao longo do tempo. Na Tabela 3 são apresentados os principais marcos definidos:

Tabela 3 - Principais Marcos

Descrição	Data Entrega
Entrega da Versão 0.1 do estado da arte	15/12/2024
Entrega do estado da arte para PREPD	04/01/2025
Apresentação da página dos KPIs	01/02/2025
Apresentação da página dos dashboards	14/02/2025
Apresentação da página da vista em mapa	28/02/2025
Apresentação intermédia da aplicação aos stakeholders	04/2025
Apresentação do protótipo final	05/2025
Entrega do relatório da dissertação	06/2025
Apresentação Final	06/2025

Para alguns dos marcos foi feita uma estimativa mensal, em vez de considerar um dia exato, devido à falta de conhecimento sobre o dia em concreto. É importante notar que estes marcos não são 100% exatos pois podem sofrer alterações, especialmente os relacionados com a apresentação dos componentes da UI.

## 2.4 Riscos Associados

Tal como todos os projetos, este também não é imune a possíveis riscos que podem comprometer o tempo no qual as tarefas são concluídas e até mesmo, em casos mais extremos, na qualidade final do projeto. Em baixo tem apresentada na Tabela 4 uma lista dos possíveis riscos com a sua probabilidade de acontecer e o impacto no projeto.

Tabela 4 - Riscos associados ao projeto

Risco	Descrição	Probabilidade (1-5)	Impacto (1-5)	Efeito
R1	Os <i>stakeholders</i> pedem uma alteração radical no design	3	4	Não será possível cumprir com a deadline a tempo
R2	Existe falta de dados nas simulações	2	4	Menos dados serão apresentados

R3	Falha nos servidores do CityLearn	1	5	Não será possível apresentar resultados
R4	Complexidade inesperada na forma de apresentar alguns dos dados	3	3	Mais tempo será despendido na tarefa, atrasando o projeto
R5	Interrupções devido a problemas de saúde ou emergências pessoais	2	5	Será perdido tempo de trabalho, atrasando o projeto
R6	Falhas de dependência de software ou hardware ou obsolescência técnica	1	5	Afeta a qualidade e integridade da interface

Para os riscos acima descritos, as ações que devem ser tomadas de forma a controlar o efeito dos mesmos na integridade do projetos envolvem uma monitorização periódica dos dados que estão a ser recolhidos para as simulações e do estado dos servidores do CityLearn, podendo assim evitar ao máximo os riscos R2 e R3 e para o risco R1 devem ser discutidas e preparadas previamente, outras opções de design para determinados componentes de UI, de forma a mitigar o trabalho a ser feito no caso de serem pedidas alterações.

Graças ao método de trabalho e ao processo de revisão periódica do projeto, o risco R4 é evitado e o R5 é parcialmente mitigado. Serão necessárias fazer verificações periódicas de atualizações de software e das ferramentas utilizadas para evitar problemas de dependências que causem falhas na integridade da interface como indica o risco R6.



## 3 Estado da Arte

Este capítulo inicia-se com uma descrição da metodologia de pesquisa adota pelo estudante e os critérios de seleção de artigos, seguida por uma exploração dos conceitos fundamentais de tecnologias como Angular, React e Vue, sendo feita uma análise das três, destacando qual a escolhida pelo estudante para o desenvolvimento da interface, e também Bootstrap e a biblioteca Recharts, destacando os seus princípios de funcionamento e o seu potencial na criação de uma interface intuitiva e eficiente para o CityLearn. Em seguida, é apresentada uma análise comparativa de aplicações similares que abordam simulações de sistemas complexos para otimização energética, com o objetivo de contextualizar os avanços tecnológicos e funcionais que sustentam a abordagem proposta.

### 3.1 Metodologia de Pesquisa

Para o processo de pesquisa, o aluno adotou por uma metodologia mais simplificada, utilizando como motor de pesquisa o Google Scholar e dando uso a *queries* como intuito de reduzir ao máximo os resultados apresentados. Para além de *queries* foram aplicados outros filtros como por exemplo, apenas utilizar artigos publicados em inglês e com no máximo 4 a 5 anos, tendo sido feitas algumas exceções para certos artigos.

Devido à diversidade dos conteúdos abordados no estado da arte, houve também uma grande quantidade de artigos encontrados, tendo sido escolhidos pelo estudante apenas os mais relevantes a apresentar. Este foi um dos grandes desafios que o estudante encontrou no seu processo de pesquisa.

Na Tabela 5, é apresentada a lista das *queries* utilizadas para a pesquisa de informações.

Tabela 5 - Lista de *queries* utilizadas

Query	Resultados
"citylearn challenge" AND (2020 OR 2021 OR 2022 OR 2023)	80 artigos
(angular AND react AND vue) AND (frontend OR development) OR (development OR ui development) AND performance	17000 artigos
EV2Gym AND (simulation OR tool)	6 artigos
recharts AND react AND (ui development OR development) AND application AND design	224 artigos

Foi então feita uma outra análise posterior à filtragem inicial, no qual o estudante adotou nas primeiras 2 páginas de resultados de artigos resultantes da pesquisa a 3-Pass-Approach, a qual permite fazer uma análise preliminar nos conteúdos de um artigo e com base nas primeiras leituras, determinar o valor científico dos mesmos para o trabalho. (Schmidl, 2020)

## 3.2 Conceitos Base

Nesta secção serão falados dos conceitos mais importantes que serão abrangidos durante o resto do documento. São linguagens, ferramentas e conceitos que estão fortemente contidos no contexto do projeto desenvolvido.

### 3.2.1 Tecnologias de Desenvolvimento Frontend

#### 3.2.1.1 AngularJS

O AngularJS é uma *framework* de JavaScript de código aberto que é mantida pela Google e por uma comunidade ativa, criada para simplificar o desenvolvimento de aplicações de página única, também conhecidas como *Single Page Applications* (SPAs). Tendo por base JavaScript, o AngularJS permite facilitar imenso o trabalho desenvolvedores tendo como objetivo principal fazer com que as aplicações web sejam mais modulares, simples de manter e escaláveis. Para organizar o desenvolvimento, utiliza o padrão MVC, facilitando alguns processos como manutenção e testes. O AngularJS é frequentemente utilizado para desenvolver aplicações web dinâmicas e interativas. Utiliza uma mistura de HTML, CSS e JavaScript para construir experiências envolventes e viabiliza a elaboração de elementos personalizados no DOM. (“Comparative Analysis of Angularjs and Reactjs,” 2016)

A interação com o *backend* é gerida de maneira eficiente, enquanto as visualizações e controladores têm um papel fundamental na criação da interface. Um dos seus principais recursos é a injeção de dependências, que permite carregar automaticamente os novos

módulos necessários. Além disso, o AngularJS proporciona vinculação de dados em ambas as direções, sincronizando automaticamente as informações entre o modelo e os elementos da visualização. Isso ajuda na criação de aplicações mais ágeis e eficientes, fazendo do AngularJS uma das opções mais populares para o desenvolvimento de SPAs.

No contexto do CityLearn, o uso de AngularJS é benéfico pois ao tomar partido das suas técnicas de detecção de mudanças eficiente e renderização baseada em componentes, garante que apenas as partes necessárias da UI são atualizadas em resposta a mudanças nos dados das simulações, o que oferece aos utilizadores uma experiência rápida, fluída e responsiva. É também capaz de criar uma aplicação altamente escalável permitindo adicionar novos recursos ou ajustar a UI de forma eficiente conforme o CityLearn evolui.

### 3.2.1.2 Vue.js

Vue.js (ou apenas Vue) é uma biblioteca de JavaScript de código aberto desenvolvida por Evan You, um ex-colaborador do Google que, ao trabalhar com Angular em vários projetos, procurou criar uma alternativa mais leve. Foram mantidas algumas funcionalidades úteis como a ligação de dados, mas evitando alguma da complexidade extra encontrada no Angular.

O Vue é definido como uma biblioteca evolutiva para o desenvolvimento de interfaces de utilizador. Isto pode ser realizado de forma progressiva, focando na camada de apresentação, sendo fácil de aprender e de integrar com outras bibliotecas ou projetos já existentes. Além disso, o Vue é muito forte para desenvolver SPAs elaboradas, especialmente ao ser combinado com os seus Single-File Components (SFCs). (Kumpulainen, 2021)

Uma das inovações que o Vue apresenta, assim como o React, é a utilização de um DOM virtual, que reflete o DOM real como objetos JavaScript que podem ser manipulados e atualizados de maneira eficiente, sem a necessidade de recarregar a página da internet. No entanto, uma das qualidades do Vue que o torna verdadeiramente distinto e o diferencia das outras linguagens é a sua leveza em relação ao tamanho. A estrutura tem cerca de 80 KB, enquanto React e Angular apresentam tamanhos muito maiores, em torno de 100 KB e 500 KB, respetivamente.

Essa variação de tamanho pode não aparentar ser muito relevante, mas acaba gerando um grande efeito no desempenho, já que o Vue apresenta um tempo médio de inicialização inferior e uma menor utilização de memória, o que é benéfico, pois oferece diversas vantagens, como poder ser utilizado em dispositivos com recursos escassos.

Para o desenvolvimento de uma UI para uma aplicação como o CityLearn, o Vue destaca-se como uma excelente opção devido à sua eficiência, modularidade e habilidade em criar interações dinâmicas. Com um tempo de inicialização rápido e o uso de DOM virtual, o Vue proporciona atualizações ágeis na interface, sendo ideal para a apresentação em tempo real dos resultados das simulações. A sua arquitetura baseada em componentes possibilita a criação de módulos reutilizáveis, como gráficos e painéis de controlo, simplificando a manutenção e permitindo a fácil expansão da aplicação. Para além disso, o Vue oferece suporte a atualizações reativas e integra-se de forma prática com bibliotecas de visualização, facilitando a criação de gráficos e representações visuais interativas.

### 3.2.1.3 React

Segundo (Chen et al., 2019), React é uma biblioteca em JavaScript que, nos últimos anos, tem conquistado o meio de desenvolvimento *frontend*. Desenvolvido pelos engenheiros do Facebook e lançado como código aberto, o React possibilita que programadores desenvolvam e montem componentes de interface para a internet.

Apresenta ideias inovadoras que estimulam os profissionais a reavaliar métodos convencionais. Desde à muito tempo, os programadores eram orientados a manter HTML, CSS e JavaScript em partes distintas. No entanto, o React sugere modificar a maneira como o desenvolvimento ocorre e, portanto, adota a estratégia de incorporar HTML e CSS diretamente no JavaScript. Embora essa abordagem possa parecer pouco convencional à primeira vista, após experimentá-la, ela alinha-se bem à tendência atual de desenvolvimento em *frontend*, cujo mesmo é baseado em componentes individualizados, mas que trabalham em harmonia entre si, tal como se pode ver noutras linguagens como por exemplo Angular.

Algumas das grandes funcionalidades do React incluem:

**Virtual DOM** – O uso de um DOM virtual em memória oferece significativas vantagens de desempenho, pois ao identificar as alterações na interface antes de enviá-las ao DOM real, possibilita reduzir as interações diretas entre ambos. Devido a isso, diminui-se o efeito de renderizações supérfluas. Isso leva a interfaces mais ágeis e reativas, mesmo em aplicações complexas com diversas interações dinâmicas. No caso do CityLearn, onde as simulações muitas vezes geram dados em tempo real ou grandes quantidades de resultados que devem ser apresentados e atualizados na interface do utilizador, essa questão tornou-se um dos fatores mais relevantes na escolha do React como ferramenta para o desenvolvimento da UI.

**Utilização de JSX** - React apresenta o JSX, uma extensão da sintaxe JavaScript que possibilita a escrita de código semelhante ao HTML dentro de arquivos JavaScript. Facilita a leitura e escrita do código ao unir estrutura e lógica em um só lugar. Isto torna o desenvolvimento mais intuitivo e reduz a separação artificial entre estrutura (HTML), estilo (CSS) e comportamento (JavaScript).

**Integração com Bibliotecas de Visualização** - As simulações muitas vezes produzem dados complexos, como séries temporais, matrizes de múltiplas dimensões ou modelos tridimensionais, que necessitam de interpretação visual para que os utilizadores possam entender os resultados de maneira clara e intuitiva. Para isso, é habitual utilizar bibliotecas específicas para visualização de dados, como D3.js, Chart.js, Plotly, ou até mesmo ferramentas para gráficos 3D, como Three.js e WebGL. Nesse contexto, o uso do React proporciona um ambiente perfeito para incorporar e administrar essas bibliotecas numa aplicação web contemporânea. No que diz respeito ao CityLearn, os dados que são disponibilizados vão desde detalhes sobre o desempenho energético, o comportamento dos edifícios e a eficácia das estratégias de controlo até dados mais abrangentes, como resumos de armazenamento. Esses gráficos e informações podem incluir formas de representação mais básicas, como gráficos de linhas, colunas ou até de áreas. Contudo, são essenciais para entender de que maneira os sistemas de energia urbana podem ser aprimorados.

Em suma, React é uma ferramenta poderosa e flexível para construir uma interface adequada para a apresentação dos resultados das simulações fornecidas pelo CityLearn. As simulações geralmente geram dados dinâmicos e, em alguns casos, em tempo real, o que exige uma interface que seja responsiva, eficiente e fácil de atualizar. A sua eficiência na gestão de atualizações, modularidade, suporte a visualizações interativas e integração com um ecossistema rico de ferramentas tornam-no ideal para lidar com os desafios técnicos e de usabilidade apresentados por este tipo de aplicação.

#### 3.2.1.4 Comparação e decisão

Após entender o funcionamento e capacidades de cada uma das linguagens de desenvolvimento *frontend* acima descritas, foi feita uma análise comparativa entre as 3 de forma a determinar qual a mais adequada para criação da UI para o CityLearn e para arranjar forma de responder à Q11.

No estudo feito em (Diniz et al., 2022), para comparar a performance e capacidade do Angular, React e Vue são seguidas 3 métricas: Tamanho dos pacotes de compilação, Tempo para Interatividade (T<sub>PI</sub>) e Tempo de manipulação do DOM. O T<sub>PI</sub> mede o período entre o fim da última tarefa longa e o início da inatividade, indicando quando a aplicação está pronta para interação do utilizador. Quanto menor o T<sub>PI</sub>, mais rápido a aplicação está disponível para uso e o tempo de manipulação do DOM foi medido pelo tempo entre o estado inicial e o momento em que a estrutura chama o seu gancho de ciclo de vida equivalente após as atualizações do componente.

Inicialmente, é feita uma análise da performance das três linguagens tendo como resultados obtidos os seguintes para o tamanho dos pacotes de compilação e T<sub>PI</sub> representado nas Figuras 5 e 6 respetivamente:

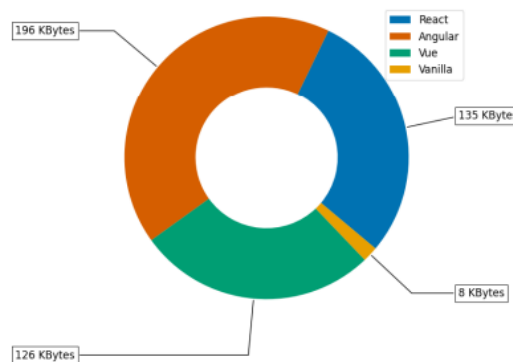


Figura 5 - Análise comparativa do tamanho de pacotes de compilação (Diniz et al., 2022)

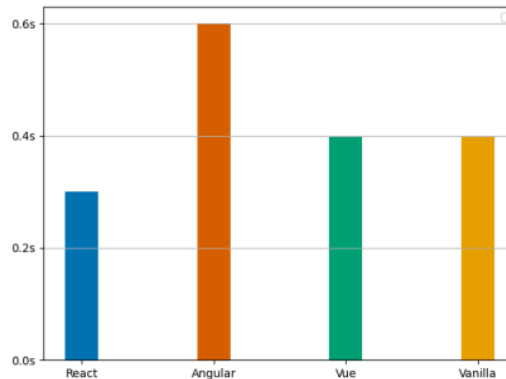


Figura 6 - Análise comparativa do Tpl (Diniz et al., 2022)

Com base nos resultados acima conseguimos chegar à conclusão de que o Angular, devido à sua estrutura baseada no padrão MVC, tem o maior tamanho de pacote entre os frameworks, enquanto React e Vue são mais leves e flexíveis. Em termos de Tpl, embora o carregamento inicial seja naturalmente rápido por não haver chamadas dinâmicas ou APIs, o React obteve o melhor desempenho, superando até o JavaScript nativo. O Vue teve um resultado semelhante ao nativo, enquanto o Angular apresentou o maior tempo, sendo o mais lento para ficar pronto para interação.

Na Figura 7, conseguimos ver os resultados da comparação do tempo de manipulação do DOM obtidos pelo estudo usando como comparação 3 cenários distintos onde são editados 1000, 10000 e 50000 elementos HTML ao mesmo tempo.

	1000	10000	50000	Total
<b>Angular</b>	3,6	29,8	61,0	94,4
<b>React</b>	7,8	51,7	179,4	239,0
<b>Vue</b>	9,7	55,6	267,9	333,3
<b>Vanilla</b>	3,2	34,7	203,8	241,7

Figura 7 - Análise comparativa do tempo de manipulação do DOM (Diniz et al., 2022)

Com base nos dados acima indicados, podemos concluir que em relação ao desempenho dos frameworks, em termos do tempo de manipulação do DOM na maior contagem de elementos, o Angular foi o mais rápido dos três, seguido pelo React e por fim o Vue.

Ao comparar o desempenho geral, Vue e React costumam se destacar em relação ao Angular em aplicações que necessitam de renderizações frequentes da interface, como no caso do CityLearn, por causa do uso de DOM virtual, que melhora as atualizações da interface. No entanto, nos casos em que é preciso realizar operações que requerem a manipulação direta do

DOM real, o Angular destaca-se por ser o mais veloz, devido à sua eficaz implementação de vinculação de dados. (Kumpulainen, 2021)

Apesar de todas as linguagens acima serem consideradas opções viáveis para o desenvolvimento de uma UI para o CityLearn que responda aos requisitos pedidos, a decisão final foi optar pela utilização de React. Para além de que os seus benefícios de desempenho, flexibilidade, grande ecossistema de bibliotecas e facilidade de integração com outras ferramentas o tornam ideal para o desenvolvimento de interfaces dinâmicas e com muitos dados, como as que o CityLearn exigiria, um ponto igualmente importante na tomada de decisão foi o desejo do estudante de querer aprender uma nova linguagem para evoluir na sua carreira profissional, em vez de optar pelo uso de Angular, na qual já se encontrava familiarizado.

### **3.2.2 Biblioteca ReCharts**

Tendo tomada a decisão de escolher React, como linguagem a utilizar no desenvolvimento da UI, outro ponto importante a ter em consideração é qual a melhor forma de representar os dados provenientes das simulações fornecidas pelo CityLearn de maneira que seja facilmente interpretável aos utilizadores, tal como referido na Q12.

Para isso, foi feita uma análise nas ferramentas existentes que possuem integração em React para representação de gráficos e outros elementos visuais. De entre uma grande paleta de opções, a que se destacou foi a biblioteca ReCharts.

A biblioteca Recharts é uma ferramenta de gráficos poderosa e simples de utilizar, desenvolvida especialmente para aplicações em React. Ela disponibiliza uma variedade de componentes gráficos modulares, reutilizáveis e customizáveis para representar dados em diversos formatos, como gráficos de linha, barras, área, radar e mais. O Recharts é desenvolvido com D3.js, oferecendo uma interface mais simples para criar representações gráficas interativas e responsivas. É uma ótima escolha para desenvolvedores que querem adicionar visualizações de dados envolventes e interativas nos seus projetos, utilizando as suas funcionalidades como animações suaves, suporte para acessibilidade e fácil integração com React. Na Figura 8 podemos ver alguns exemplos, inseridos no contexto de uma *dashboard*, dos tipos de gráficos que a biblioteca permite criar.



Figura 8 - Gráficos da biblioteca Recharts, imagem obtida em (Abyl, 2023)

Esta biblioteca é útil na visualização de dados de análise, aumentando assim a legibilidade dos resultados de testes tal como demonstrado em outros trabalhos como (Enhancing Readability, Accessibility, and Shareability of Robot Framework Test Results, 2023) no qual foi uma excelente ferramenta para demonstração de resultados. Também podemos dar uso ao gráficos como forma de detetar certas tendências em dados ou a fazer comparações entre dados como foi realizado no estudo feito em (Web Application for questionnaire Creation and monitoring in React.js, n.d.)

### 3.2.3 Bootstrap

Atualmente, Bootstrap é a estrutura mais utilizada de HTML, CSS e JavaScript para desenvolver sites responsivos e adaptáveis a dispositivos móveis. Com uma das suas principais vantagens sendo o facto de ser *open-source* e completamente gratuito para download e utilização, o Bootstrap é frequentemente utilizado para facilitar e acelerar o desenvolvimento em *frontend*.

A biblioteca apresenta modelos de design apoiados em HTML e CSS que incluem tipografia, formulários, botões, tabelas, navegação, modais, carrosséis de imagens e mais. Além disso, o Bootstrap oferece plug-ins JavaScript, facilitando a implementação de funcionalidades interativas e a criação de layouts responsivos.

Algumas das funcionalidades mais úteis que o Bootstrap oferece são:

**Sistema de grelha altamente responsivo** - O Bootstrap inclui um sistema de grelha fluido, responsivo e baseado no conceito "mobile-first", que se adapta dinamicamente a diferentes tamanhos de dispositivos ou janelas de visualização. Tal como descrito em (Jake Spurlock, 2013) o sistema permite organizar layouts utilizando 12 colunas, ajustando-se automaticamente conforme o tamanho do ecrã aumenta ou diminui, tendo por base os 724px ou 1170px em largura. Na Figura 9 podemos ver de forma mais clara a estrutura base deste sistema de grelha.



Figura 9 - Sistema de grelha do Bootstrap (Jake Spurlock, 2013)

O sistema oferece classes predefinidas para facilitar a criação de layouts flexíveis e intuitivos, além de *mix-íns* poderosos, que permitem gerar layouts semânticos personalizados com maior controlo e flexibilidade.

**Suporte em múltiplos browsers** - O Bootstrap foi criado para ser compatível com todos os navegadores atuais, como Chrome, Firefox, Safari, Edge. Isto é possível, já que o Bootstrap utiliza padrões da web bem definidos, como HTML5, CSS3 e JavaScript, que são amplamente aceites pelos navegadores, além de incluir correções e redireccionamentos para navegadores com menor compatibilidade. Essa vantagem economiza tempo na codificação, pois evita a necessidade de ajustar o código manualmente para cada navegador.

**Diversidade de componentes de UI personalizáveis** – Uma grande vantagem que o Bootstrap traz, é o suporte de componentes de UI próprios. Estes são fundamentais numa *framework* Bootstrap tal como referido por (Suman Aryal BOOTSTRAP-A Front-End Framework For Responsive Web Design, n.d.) pois auxiliam rapidamente na criação de projetos e interfaces com o uso de classes CSS pré-definidas e menos código. Esses componentes são utilizados nos elementos HTML e podem ser ajustados de acordo com as necessidades dos utilizadores. Há uma significativa flexibilidade na personalização dos componentes, possibilitando a utilização de várias categorias de modificadores. Numa aplicação que pode sofrer muitas modificações pelos *stakeholders* ou outros fatores, como uma aplicação de visualização de simuladores, isso representa uma vantagem, não apenas para os seus utilizadores atuais, mas também para aqueles que poderão desenvolver sobre ela no futuro, sendo simples de editar e manipular.

Como o CityLearn é uma plataforma que depende fortemente de visualizações e interatividade para apresentar os resultados de simulações e permitir o ajuste de parâmetros, o uso de Bootstrap complementa o React ao fornecer uma estrutura sólida para estilização e componentes prontos para uso, o que acelera o desenvolvimento e melhora a experiência do utilizador. Alguns exemplos dessas funcionalidades que podem ser aplicadas incluem painéis de controlo intuitivos para modificar parâmetros das simulações, a utilização de sistema em grelha para dispor gráficos e visualizações lado a lado ou empilhados, conforme o tamanho do ecrã, modais para apresentar detalhes ou gráficos específicos, e separadores para classificar dados em categorias.

### 3.3 Estado da Arte

Nesta secção será feita uma descrição do simulador a ser trabalhado nesta dissertação, ou seja, o CityLearn e uma avaliação do estado da arte em algumas aplicações similares ao mesmo e a influencia que a existência de uma UI tem nas que já possuem uma, e também como a implementação de uma UI aprimorada afetaria as mesmas em termos de visibilidade de informação e interação com os utilizadores.

#### 3.3.1 CityLearn

O CityLearn é uma estrutura fundamentada no ambiente OpenAI Gym, que possibilita aos investigadores implementarem, partilharem, replicarem e compararem as suas metodologias de aprendizagem por reforço de forma mais acessível. Por ser uma plataforma modular e de código aberto, ela pode ser facilmente ajustada e personalizada, permitindo a adição de sistemas extras, como os de armazenamento, geração ou consumo de energia, conforme a necessidade. (Vázquez-Canteli et al., 2019)

Ao longo dos anos, investigadores de diversas áreas têm explorado o potencial da inteligência artificial (IA) e de sistemas de controlo distribuído para enfrentar uma variedade de desafios no setor energético, dando origem ao chamado “CityLearn Challenge”. O elevado consumo de eletricidade em áreas urbanas não apenas encarece os preços da energia, mas também gera instabilidade na rede elétrica, podendo resultar em apagões. Além disso, o crescente consumo de energia e as emissões de carbono intensificam consideravelmente a questão das mudanças climáticas.

Algumas das mais relevantes investigações e consequentes descobertas feitas durante os anos foram as seguintes:

##### 3.3.1.1 The CityLearn Challenge 2020

No ano de 2020, onde ocorre a primeira edição do CityLearn Challenge, um total de quatro equipas submeteram os seus agentes ao desafio, tendo obtido os resultados demonstrados na Figura 10. (Vázquez-Canteli et al., 2020)

Alguns dos concorrentes do desafio mostraram que a aprendizagem por reforço pode ultrapassar a eficiência de controladores baseados em regras estabelecidas. Os agentes foram avaliados num conjunto de dados exclusivo, distinto daquele empregue durante a fase de desenvolvimento e treino, demonstrando que a aprendizagem por reforço pode entender dinâmicas anteriormente não vistas em vários sistemas de energia e atuar de maneira mais eficaz do que os controladores convencionais.

Challenge Dataset (Final Scores)								
Rank	Team	Climate Zone	Ramping	1 - Load factor	Avg. Daily peak	Peak demand	Net elec. consumption	Average Score
1	pikapika	1	0.661	0.921	0.902	0.909	0.977	0.874
		2	0.706	0.946	0.864	0.998	0.979	0.899
		3	0.679	0.870	0.830	0.857	0.978	0.843
		4	0.695	0.910	0.892	1.015	0.992	0.901
Average Score								<b>0.879</b>
2	ProtoBuds	1	0.779	1.014	0.982	1.131	1.015	0.984
		2	0.780	0.980	0.959	0.999	1.013	0.946
		3	0.812	0.960	0.939	1.083	1.018	0.962
		4	0.860	0.996	0.991	1.013	1.017	0.976
Average Score								<b>0.967</b>
3	Learned City	1	0.935	1.078	1.071	1.184	1.008	1.055
		2	0.917	1.082	1.050	1.173	1.014	1.048
		3	0.886	1.033	1.003	1.103	1.008	1.007
		4	0.899	1.047	1.012	1.052	0.993	1.001
Average Score								<b>1.028</b>
4	EoRL	1	0.929	1.131	1.150	1.220	1.044	1.095
		2	0.940	1.095	1.105	1.254	1.046	1.088
		3	0.898	1.064	1.053	1.221	1.049	1.057
		4	0.892	1.065	1.069	1.107	1.035	1.034
Average Score								<b>1.068</b>

Figura 10 - Resultados do CityLearn Challenge 2020, (Vázquez-Canteli et al., 2020)

Nos anos posteriores, o foco será concentrado na integração de novos sistemas de energia no CityLearn, como baterias e veículos elétricos, que serão incorporados nas próximas edições do CityLearn Challenge.

### 3.3.1.2 The CityLearn Challenge 2021

Na edição deste ano, foi proposto um desafio de controlo com vários agentes onde as equipas participantes devem administrar a energia consumida por cada um dos edifícios numa microrrede simulada. Ao contrário de problemas mais tradicionais da aprendizagem por reforço, onde há múltiplos episódios, o CityLearn Challenge 2021 possui somente um episódio de quatro anos. Os agentes precisaram de investigar e examinar a política de controlo mais eficiente que pudessem compreender durante esse período de quatro anos. Isto reproduz uma execução real de uma implementação de agentes de RL sem modelos. (Nagy et al., 2021)

Nesta edição participaram 5 equipas com os resultados finais de cada uma sendo os representados na Figura 11.

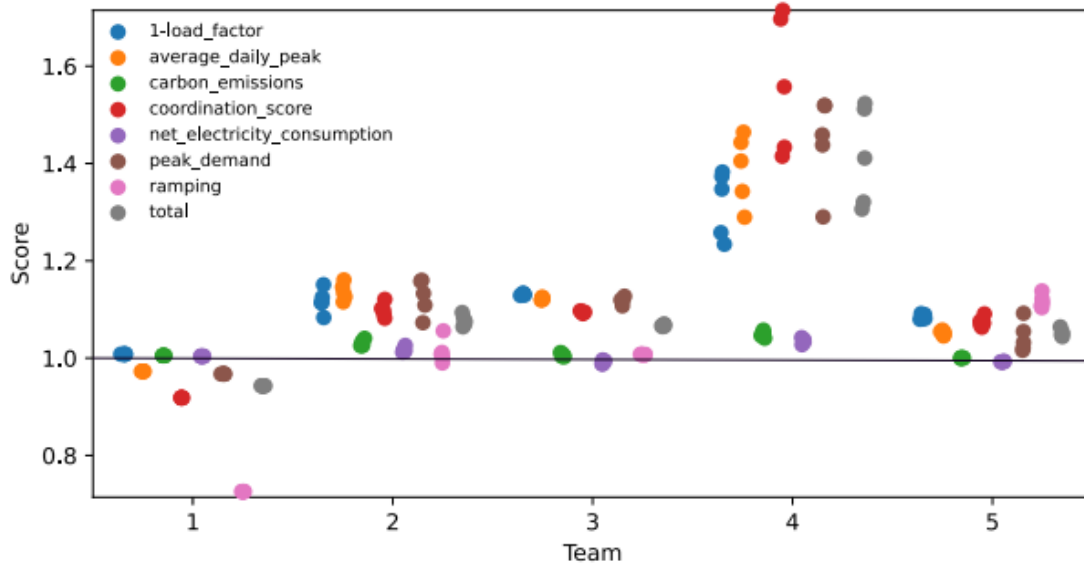


Figura 11 - Resultados do CityLearn Challenge 2021 (Nagy et al., 2021)

Tem em consideração que uma pontuação de 1 é o controlador baseado em regras de referência, isto demonstra que a equipa nº 1 possui os melhores agentes gerais.

A principal conclusão que foi retirada neste ano para estudos futuros foi a necessidade de uma maior demanda por mais estudos em design de recompensas para criar resultados para as microrredes.

### 3.3.1.3 The CityLearn Challenge 2023

No ano de 2023, foi apresentado um algoritmo pela equipa CHESCA (Garmendia et al., n.d.) para coordenar a gestão de energia em edifícios pertencentes a uma mesma comunidade, ou seja, uma área geográfica comum. Esta solução é interpretável e generalizável sem a necessidade de um grande volume de dados de treinamento, integrando heurísticas e métodos tradicionais de otimização. Ainda assim, esta abordagem foi a vencedora do CityLearn Challenge 2023, superando alternativas que utilizam algoritmos mais sofisticados baseados em RL. A Figura 12 mostra os resultados das equipas participantes nesta edição, considerando que quanto menor o valor, melhor o resultado.

Team	Private Cost	Public Cost	Comfort	Emissions	Ramp.	Load	Peak	All-Time Peak	Resilience	Unservd En.
RBC (baseline)	1.124	1.085	2.190	0.994	1.045	0.673	1.432	1.436	0.803	0.750
CHESCA	<b>0.565</b>	0.562	0.132	0.944	0.892	0.951	0.875	0.811	0.715	0.448
Team 2	0.575	<b>0.464</b>	0.304	0.883	0.783	0.869	0.844	0.789	0.398	0.278
Team 3	0.582	0.508	0.203	0.932	0.798	0.911	0.887	0.792	0.445	0.508
CHESCA*	0.548	0.522	0.129	0.930	0.845	0.954	0.873	0.958	0.642	0.316

Figura 12 - Resultados do CityLearn Challenge 2023 (Garmendia et al., n.d.)

Concluindo, nesta edição do CityLearn Challenge foi feita uma exploração da estrutura do algoritmo proposto e examinados os fatores contribuindo para o seu desempenho tanto no conjunto de dados de treino quanto no conjuntos de avaliação anteriormente desconhecidos. Além disso, foram destacadas as limitações da abordagem em relação à qualidade das variáveis de previsão.

Como direção de investigação futura, pretendem-se construir algoritmos para gestão de troca de energia combinando heurística e RL profundo. Algo importante também a atingir no futuro, é arranjar uma forma de conseguir comunicar os resultados a um público mais amplo, de uma forma não tão técnica.

### **3.3.2 EnergyPlus**

EnergyPlus é um programa baseado sobre os recursos e capacidades mais populares do BLAST e DOE-2, outros programas de simulação de energia em edifícios. Ele é constituído por componentes completamente novos, modulares e código estruturado escrito em Fortran 90. (Crawley et al., n.d.)

Contém uma variedade de recursos de simulação inovadores - como etapas de tempo variáveis, sistemas modulares ajustáveis pelo utilizador que se integram a uma simulação de zonas baseada no balanço de calor e massa - além de estruturas de dados de entrada e saída adaptadas para facilitar o desenvolvimento de módulos e interfaces de terceiros.

Alguns dos benefícios gerais trazidos pelo EnergyPlus são:

- As capacidades de simulação englobam diversos aspetos, incluindo simulação integrada, equilíbrio conjunto de transferência de calor e massa, fluxo de ar em múltiplas zonas, circuitos HVAC (sistema flexível e simulação de instalação), conexões para simulação de sistema/planta SPARK e TRNSYS, e algoritmos do novo conjunto de ferramentas de carregamento da ASHRAE;
- As capacidades de entrada, saída e simulação são significativamente mais adaptáveis;
- O EnergyPlus será disponibilizado mais rapidamente do que a inclusão de capacidades similares no DOE-2 ou BLAST, mas proporciona benefícios equivalentes.

No entanto, um dos problemas presentes no EnergyPlus é o facto de ser principalmente um mecanismo de simulação, ou seja, não tinha sido planeado possuir uma interface muito desenvolvida, utilizando elementos básicos no seu design, mas tendo minimamente a capacidade para demonstrar alguns gráficos e modelos como mostra a Figura 13.

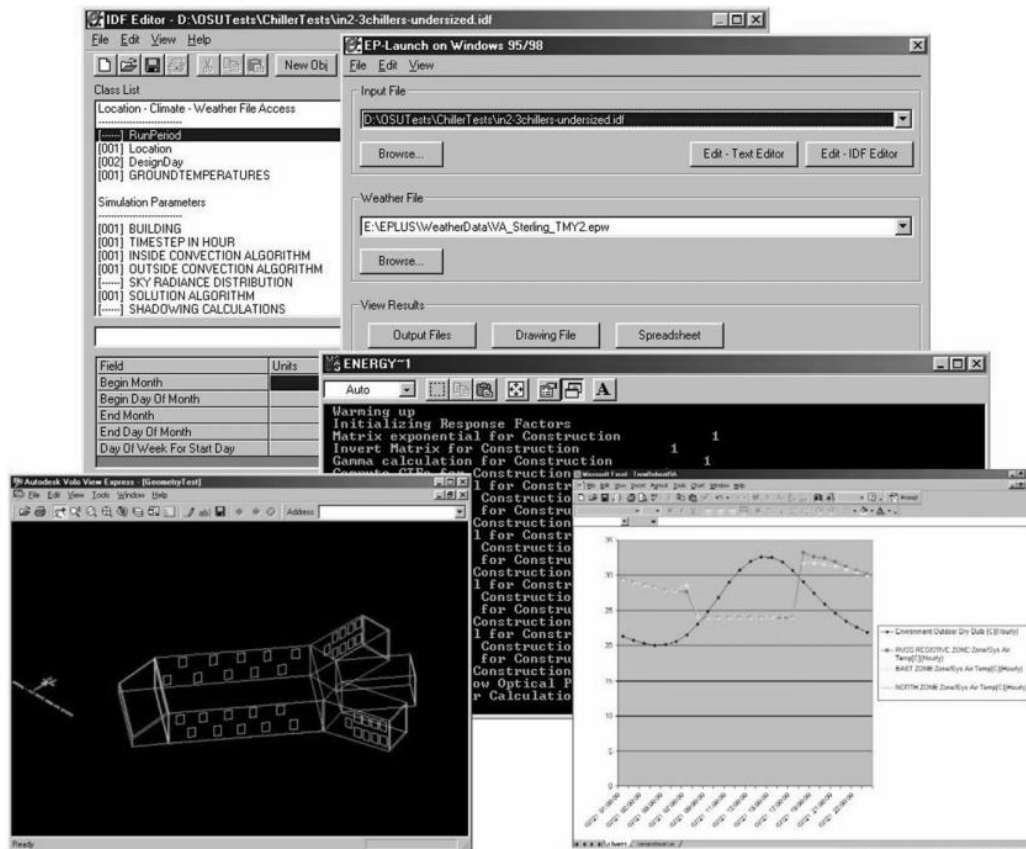


Figura 13 - Interface do EnergyPlus (Crawley et al., n.d.)

### 3.3.3 SustainGym

O SustainGym é um ambiente de simulação desenvolvido para a investigação e criação de soluções sustentáveis em sistemas energéticos, fazendo uso da aprendizagem por reforço (RL, Reinforcement Learning). Assim como o CityLearn, o SustainGym concentra-se na otimização de operações em sistemas complexos, porém possui características distintas em termos de flexibilidade e aplicação a diversos cenários sustentáveis. Foi criado com a finalidade de estabelecer um ambiente padronizado e expansível para o treino de agentes de aprendizagem por reforço, possibilitando a experimentação de várias táticas de otimização e decisão.

O SustainGym é constituído por um conjunto de 5 ambientes RL que modelam realisticamente configurações de sustentabilidade:

- O **EVChargingEnv** modela o problema de programar o carregamento de veículos elétricos (EV) para atender necessidades do utilizador e, ao mesmo tempo, minimizar as emissões de CO2.
- O **ElectricityMarketEnv** modela um sistema de armazenamento de baterias em escala de rede que concorre ao mercado de eletricidade para gerar lucro (através de arbitragem de preços) e reduzir as emissões de CO2.

- O **DatacenterEnv** modela um *datacenter* decidindo sobre uma “curva de capacidade virtual” para mudar de forma flexível empregos em horários com menores emissões de CO2.
- A **CogenEnv** modela uma planta de cogeração de ciclo combinado produzindo vapor e eletricidade para atender ao consumo local e, ao mesmo tempo, minimizar o uso de combustível e os custos de rampa.
- O **BuildingEnv** modela o controlo térmico dos sistemas de energia predial para reduzir o consumo total de eletricidade enquanto satisfaz o requisito de temperatura especificado pelo utilizador.

Uma característica essencial dos ambientes SustainGym é a possibilidade de testar algoritmos RL sob variações realistas e naturais de distribuição externa. Além disso, o SustainGym destaca-se por incluir interações multiagentes, limitações físicas e combinações de ações discretas e contínuas. (Yeh et al., n.d.)

Na Figura 14 podemos ver alguns exemplos de resultados experimentais para os 5 ambientes RL:

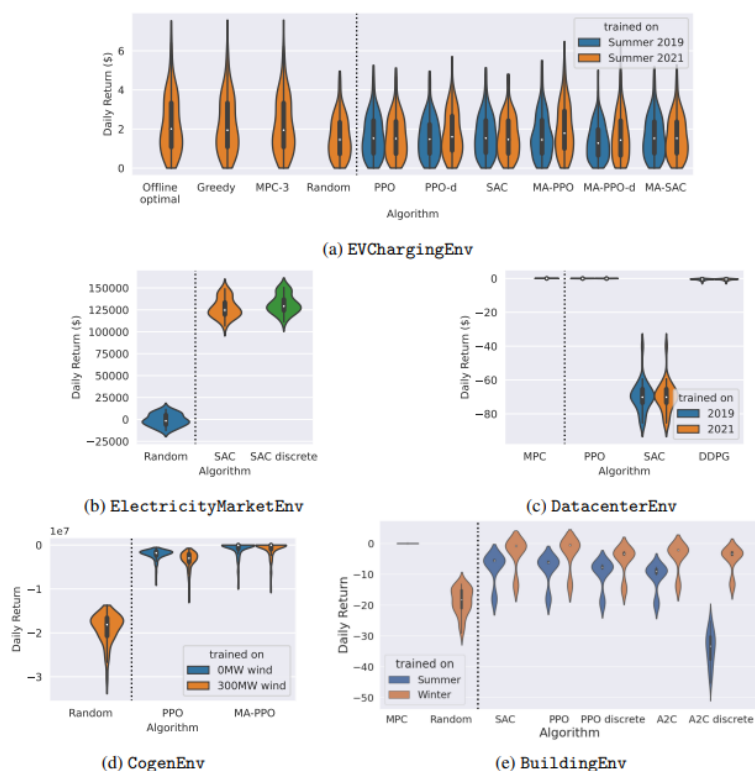


Figura 14 - Resultados experimentais dos 5 ambientes RL (Yeh et al., n.d.)

Algo a ter em conta sobre os resultados acima obtidos é que, o SustainGym não possui uma UI definida, logo os gráficos aqui demonstrados foram feitos no estudo em (Yeh et al., n.d.) posteriormente após a análise e tratamento dos resultados obtidos. A adição de uma interface que permitisse visualizar e manipular de forma clara e concisa os dados acima obtidos seria um

ponto a melhorar para o SustainGym, especialmente para utilizadores que ainda estejam no processo de aprendizagem do funcionamento da ferramenta, será uma excelente adição para os ajudar a interpretar e analisar os resultados obtidos.

### 3.3.4 EV2Gym

O EV2Gym é uma plataforma de simulação realista projetada para criar e testar algoritmos de carregamento inteligente em escala reduzida e ampliada, dentro de um ambiente padronizado. O simulador inclui modelos detalhados de veículos elétricos, estações de carregamento, transformadores elétricos e comportamentos de EV, validados com dados reais.

Com uma interface altamente flexível, o EV2Gym permite que os utilizadores escolham entre estudos de caso prontos ou criem cenários personalizados para atender a necessidades específicas. Além disso, oferece uma ampla variedade de algoritmos, como aprendizagem por reforço, programação matemática e heurísticas, acelerando o desenvolvimento e a análise de novas soluções. Ao combinar ferramentas e padrões, o EV2Gym proporciona um ambiente robusto para elaborar e avaliar algoritmos de carregamento inteligente. (Orfanoudakis et al., 2024)

Na Figura 15, é possível ver como funciona o processo de simulação no EV2Gym.

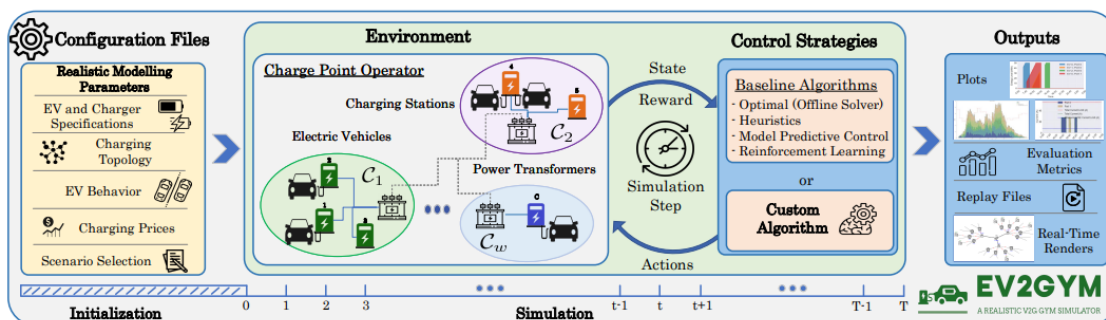


Figura 15 - Processo de simulação no EV2Gym (Orfanoudakis et al., 2024)

Em termos de interfaces visuais, apesar de o EV2Gym contar com a vantagem de ter uma interface altamente personalizável ao dar a possibilidade de escolha aos utilizadores entre os cenários que pode montar, essa interface limita-se ao ato da construção de cenários, e não tanto à apresentação dos resultados dos mesmos. A adição de formas de visualizar os resultados, seja sobre a forma de gráficos de linhas, barras, etc. ou com o uso de tabelas dinâmicas, iriam incrementar substancialmente a qualidade da interface atualmente implementada.

### 3.3.5 Importância da utilização de UIs em aplicações de simulações

Para as aplicações acima referidas, as quais ou não possuem uma interface gráfica satisfatória ou suficiente para atender às necessidades dos utilizadores ou não possuem uma de todo, é

necessário considerar a possibilidade de adotar uma interface para estas devido às grandes vantagens que a inclusão de uma dá não só em termos de valor para os utilizadores, como uma nova janela de oportunidade para evolução dos próprios produtos.

Quando as interfaces são adequadamente elaboradas, elas conseguem gerar emoções positivas e proporcionar uma experiência agradável aos utilizadores, melhorando a usabilidade. As emoções positivas não apenas oferecem alívio psicológico aos utilizadores, mas também podem facilitar a interpretação do conteúdo de design. Os utilizadores valorizam mais os produtos que conseguem proporcionar uma experiência positiva, o que pode influenciar diretamente na usabilidade de um produto. Isto é, a satisfação do utilizador é um fator crucial na relação entre o ser humano e o computador, refletindo-se em questões de utilidade e facilidade de uso de um produto. Isso sugere que a usabilidade é igualmente afetada pela funcionalidade e por experiências estéticas atraentes. (Zhou, n.d.)

De forma a alcançar este efeito nos utilizadores e a assegurar uma interface bem estruturada e idealizada é importante considerar os cinco grandes componentes de design da usabilidades de engenharia tal como descritas pelo ISO 9241-400. (Miraz et al., 2021)

- i. **Segurança e Proteção:** A interface não deve ser prejudicial e deve ajudar a reduzir o erro do utilizador;
- ii. **Eficácia:** a interface deve auxiliar e não atrapalhar a tarefa do utilizador;
- iii. **Eficiência e Funcionalidade:** a interface deve agilizar a tarefa do utilizador;
- iv. **Alegria e diversão:** o uso da interface não deve ser uma tarefa trabalhosa
- v. **Facilidade de aprendizagem e memorização:** a curva de aprendizagem não deve ser íngreme e os procedimentos aprendidos devem ser memoráveis

Nos casos das aplicações como GridLearn, SustainGym e o EV2Gym, o uso de uma interface seguindo os parâmetros acima indicados poderia trazer uma mais-valia para os seus utilizadores, facilitando os procedimentos de investigação e desenvolvimento de avaliação de algoritmos e otimização de sistemas, através de um meio no qual é mais intuitiva e simples a visualização e manipulação destes processos. No entanto, apesar disso, o simulador final a ser escolhido como objeto de estudo foi o CityLearn, devido ao facto de ter mais funcionalidades e potencial de evolução comparativamente às outras aplicações de simulação.

Após esta análise acerca das vantagens e oportunidades que o uso de uma UI fornece às aplicações acima descritas, a Q13 é assim abordada e respondida no contexto do projeto desenvolvido pelo estudante.



## 4 Análise de Requisitos e Design

Este capítulo foca-se na definição dos requisitos funcionais e não funcionais, onde são analisados no contexto da interface descrevendo de que forma estes devem ser seguidos durante a implementação da interface. É feita também uma apresentação da forma como a interface desenvolvida se enquadra em conjunto com os outros componentes do CityLearn presentes e qual a sua importância para o objetivo final do projeto EnergAlze através do uso do modelo C4.

### 4.1 Requisitos Funcionais e Não Funcionais

No processo de desenvolvimento de software, a eficácia do projeto está fortemente atrelada à exatidão e à coleta integral dos requisitos. A identificação dos requisitos funcionais e não funcionais possibilita que especialistas em software identifiquem rapidamente quais são os requisitos mais importantes, sem a necessidade de analisar o sistema na sua totalidade (Tiun et al., 2020).

O levantamento de requisitos foi em grande parte responsabilidade do estudante, tendo também orientação dos membros do grupo SoftCPS de forma a ter uma visão geral de alguns dos requisitos base a ter em conta. Tendo em conta que este é um trabalho de investigação em constante evolução, o estudante teve uma participação ativa no que toca a análise do problema e dos casos de uso levantados, originando, por vezes, a necessidade de alterar ou até de acrescentar requisitos ou novos conceitos para que estes se tornassem mais adequados ao problema em questão o que contribuiu para a definição de novas ideias no projeto. A definição destes requisitos também contribuiu para a publicação de um trabalho melhor para uso na plataforma do CityLearn.

### 4.1.1 Requisitos Funcionais

Os requisitos funcionais são todas as necessidades, características ou funcionalidades esperadas num processo que podem ser atendidos pelo software. De forma geral, um requisito funcional expressa uma ação que deve ser realizada através do sistema, ou seja, um requisito funcional é "o que o sistema deve fazer". São fundamentais no desenvolvimento de software, uma vez que serão um guia na elaboração de todas as funcionalidades do programa, baseadas nas necessidades do utilizador (FatimatuZZahra et al., 2025).

Foi feita uma estruturação e representação dos requisitos e funcionalidades principais que a interface deve implementar através do uso de um diagrama de casos de uso como demonstra a Figura 16.

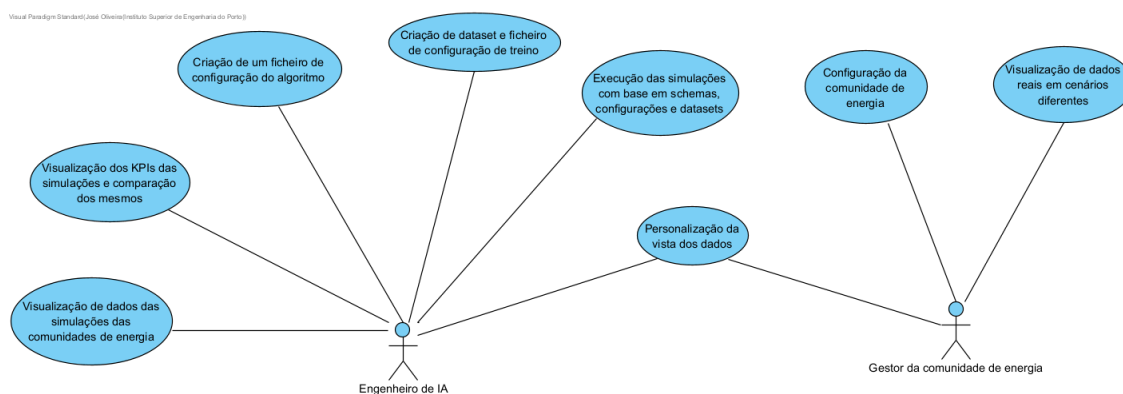


Figura 16 - Casos de uso

Os casos de uso definidos estão divididos por dois atores principais: O engenheiro de IA é responsável pela visualização e análise dos resultados das simulações, visualização dos KPIs das mesmas, pela criação das configurações para executar novas simulações através de ficheiros de configuração e *datasets* e finalmente a execução das simulações com base nessas configurações. O gestor da comunidade de energia tem como principais ações a configuração da comunidade de energia e a visualização dos dados em tempo real. Ambos utilizadores podem customizar a visualização dos dados nas páginas às quais têm acesso.

Os casos de uso descritos representam maioritariamente ações como a visualização de dados ou a criação de métodos para gerar mais dados (através da criação de simulações). Isto devido ao facto de a interface ser dedicada à visualização e análise dos dados provenientes das simulações feitas pelo simulador do CityLearn e de dados reais provenientes de vários cenários.

Para mais facilmente referir aos requisitos estes foram numerados e listados como mostra a Tabela 6:

Tabela 6 - Requisitos Funcionais

Identificador	Requisito
RF1	Visualização de dados das simulações das comunidades de energia
RF2	Visualização dos KPIs das simulações e comparação dos mesmos
RF3	Criação de um ficheiro de configuração do algoritmo
RF4	Criação de dataset e ficheiro de configuração de treino
RF5	Execução das simulações com base em schemas, configurações e datasets
RF6	Configuração da comunidade de energia
RF7	Visualização de dados reais em cenários diferentes
RF8	Personalização da vista dos dados

#### 4.1.2 Requisitos Não Funcionais

Um requisito não funcional pode ser definido como “de que forma” o sistema deve fazer. Uma forma simples de entender o que é um requisito não funcional é ter por base que todos os requisitos não funcionais devem expressar uma premissa ou restrição do sistema. Dessa forma, requisitos não funcionais devem sempre ser mensuráveis, ou seja, deve ser possível verificar se eles estão ou não a ser atendidos pelo software. São utilizados para facilitar a utilização de recursos e garantir a qualidade das características do sistema, de forma que o impacto do requisito não seja diretamente notado ou percebido pelo utilizador (Fatimatuzzahra et al., 2025).

Para a descrição dos requisitos não funcionais foi seguido o modelo FURPS+ que divide os requisitos não funcionais em várias categorias: funcionalidade, usabilidade, confiabilidade, desempenho, suportabilidade. O “+” em FURPS foi adicionado de forma a enfatizar vários atributos como o design, implementação, interface e requisitos físicos.

Igualmente aos requisitos funcionais, estes foram também numerados e listados, estando divididos pelas categorias do modelo FURPS.

##### 4.1.2.1 Funcionalidade

A funcionalidade é onde é feita uma descrição específica de todas as funcionalidades que englobam a aplicação, ou seja, de certa forma é uma descrição dos requisitos funcionais (Joaquim Pessôa Filho, 2024).

Para o caso da UI desenvolvida para o CityLearn, a funcionalidade é representada através dos requisitos funcionais descritos anteriormente e os quais serão abordados com mais detalhe nos capítulos seguintes.

Com base nisso foi apenas retirado um requisito para esta secção como mostra a Tabela 7:

Tabela 7 - Requisitos de funcionalidade

Identificador	Requisito
RNF1	Deve cumprir com todos os requisitos funcionais

#### 4.1.2.2 Usabilidade

A usabilidade avalia as interações da interface com os utilizadores. Neste requisitos procura-se arranjar padrões de uso de forma a garantir consistência nas formas como se utiliza o sistema, prevenindo usos incorretos do mesmo e também noutros fatores como a estética, design, suporte e documentação (Joaquim Pessoa Filho, 2024).

Para existir uma boa usabilidade na interface deve existir uma estrutura intuitiva e fácil de compreender ao utilizador e deve haver uma forma de acesso rápido à documentação do CityLearn, para ajudar na pesquisa de informação relativos a certos processos.

Na Tabela 8 são apresentados os requisitos retirados:

Tabela 8 - Requisitos de usabilidade

Identificador	Requisito
RNF2	A navegação na interface deve ser intuitiva
RNF3	A interface tem uma forma de aceder rapidamente à documentação do CityLearn

#### 4.1.2.3 Confiabilidade

A confiabilidade é outro requisito essencial pois garante aos utilizadores confiança no sistema que estão a utilizar. Na confiabilidade o essencial é definir o quão confiável o sistema precisa de ser, fazendo uma avaliação de várias métricas como a frequência de falhas aceitáveis, o tempo médio entre estas e as formas que o sistema tem de recuperar das mesmas (Joaquim Pessoa Filho, 2024).

Nesta categoria os requisitos relevantes são garantir a disponibilidade do sistema e ser viável de ter disponível em *open-source* pois existe interesse dos criadores do CityLearn para esta possibilidade.

Na Tabela 9 são apresentados os requisitos retirados:

Tabela 9 - Requisitos de confiabilidade

Identificador	Requisito
RNF4	A interface deve estar sempre disponível para utilização
RNF5	A interface deve estar disponível em <i>open-source</i>

#### 4.1.2.4 Desempenho

Para o sucesso da interface a desenvolver é importante que esta tenha um tempo de resposta aceitável, embora não existam especificações em relação ao limite deste tempo, é uma boa

prática e garante maior eficiência. Algumas das métricas medidas neste requisito são a memória máxima utilizada, a quantidade máxima de utilizadores em simultâneo, a carga máxima suportada e a disponibilidade (Joaquim Pessôa Filho, 2024).

O desempenho é um dos requisitos mais importantes no desenvolvimento da interface para o CityLearn pois deve ser possível visualizar e filtrar os dados relativos às simulações com alguma rapidez e eficiência. Deve ser também capaz de suportar um elevado volume de dados relativos a diversas simulações, podendo estas englobar dezenas de edifícios com dados de minuto a minuto.

Na Tabela 10 são apresentados os requisitos retirados:

Tabela 10 - Requisitos de desempenho

Identificador	Requisito
RNF6	A interface deve conseguir filtrar os dados de forma rápida e eficaz
RNF7	A interface deve ser capaz de suportar um elevado volume de dados

#### 4.1.2.5 Suportabilidade

Esta categoria engloba vários aspetos como a testabilidade, adaptabilidade, manutenibilidade, compatibilidade, configurabilidade e escalabilidade.

Em termos de suportabilidade, a interface desenvolvida deve ser facilmente escalável pois é um projeto em evolução contínua e deve ser possível ser implantado externamente.

Na Tabela 11 são apresentados os requisitos retirados:

Tabela 11 - Requisitos de suportabilidade

Identificador	Requisito
RNF8	A interface deve ser escalável
RNF9	A interface deve poder ser implantada externamente

#### 4.1.2.6 Design

Os requisitos de design descrevem restrições relativas a aspetos visuais como por exemplo elementos na interface, botões, cores, fontes, entre outros (Joaquim Pessôa Filho, 2024).

Para a interface desenvolvida as restrições definidas englobam não só garantir a legibilidade dos campos e das datas nos gráficos e também apresentar as unidades corretamente (por exemplo kWh, %, etc.), como também garantir que não há inconsistências nos componentes usados, ou seja, para funcionalidades similares devem ser utilizados componentes similares de forma a não confundir o utilizador.

Na Tabela 12 são apresentados os requisitos retirados:

Tabela 12 - Requisitos de design

Identificador	Requisito
RNF10	Os campos presentes nos gráficos da interface devem ser legíveis
RNF11	As unidades dos campos devem estar corretamente apresentadas
RNF12	Deve ser garantida a consistência nos componentes usados na interface

#### 4.1.2.7 Implementação

Os requisitos de implementação definem restrições como a linguagem a ser utilizada, o ambiente de desenvolvimento ou outros aspectos que dirigem como deve ser implementada a solução (Joaquim Pessoa Filho, 2024).

Em termos de implementação é necessário a aplicação estar em inglês pois esta pode vir a ser utilizada globalmente. Como a interface está contida no projeto EnergAlze, deve também estar integrada e funcionar em conjunto com os outros componentes do projeto.

Na Tabela 13 são apresentados os requisitos retirados:

Tabela 13 - Requisitos de implementação

Identificador	Requisito
RNF13	A interface deve estar disponível em inglês
RNF14	A interface deve estar integrada em conjunto com os outros componentes do projeto

#### 4.1.2.8 Requisitos Físicos

Estes requisitos referem-se a fatores físicos como por exemplo peso, tamanho, etc. São requisitos mais orientados para o hardware utilizado que permita minimamente a produção da solução ao contrário dos outros anteriormente falados que se concentram mais na implementação lógica ou requisitos de software (Joaquim Pessoa Filho, 2024).

Não existem requisitos levantados que digam respeito a esta categoria.

## 4.2 Modelo C4 + 1

O modelo C4+1 é uma abordagem usada para caracterizar sistemas de software através de quatro perspectivas. Cada um dos níveis do modelo foca-se numa visão particular do sistema. É independente de ferramentas e notações, o que implica que pode ser aplicado com qualquer linguagem ou software de desenho de diagramas, pois fornece uma sintaxe que pode simplificar e facilitar a tarefa de conceitualizar e abstrair a funcionalidade de um sistema (Cardoso et al., 2020).

## 4.2.1 Nível 1

O primeiro nível foca-se em enquadrar o sistema de software a ser modelado, apresentando as diversas pessoas ou agentes envolvidos e os sistemas externos que oferecem algum tipo de serviço ao sistema. As relações entre os componentes são ilustradas por linhas, acompanhadas de uma explicação sobre a função do vínculo. Serviços externos ou já implementados são destacados com cores diferentes do sistema de software alvo. Neste nível, informações extras sobre os sistemas de software envolvidos são deixadas de lado, oferecendo uma visão geral do cenário em que o sistema alvo será situado. (Cardoso et al., 2020)

### 4.2.1.1 Vista Lógica

O diagrama da Figura 17, representa a vista lógica de nível 1:

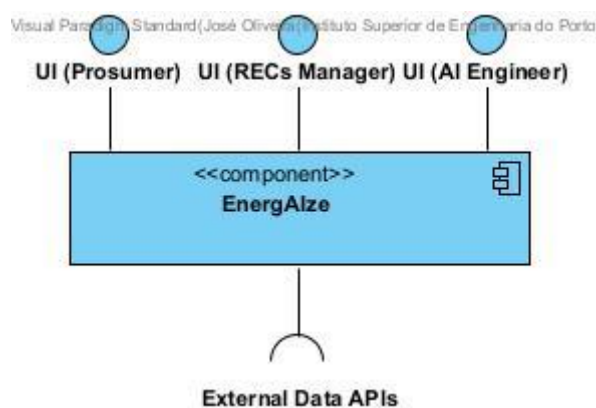


Figura 17 - Vista Lógica - Nível 1

Aqui temos uma vista geral do projeto EnergAlze no qual a interface desenvolvida se encontra inserida. O EnergAlze é um sistema desenvolvido para administrar a energia de locais, que podem uniformemente constituir um REC, impulsionando um modelo fundamentado em IA. O sistema EnergAlze possibilita a administração de energia focada no utilizador e inclui ferramentas que viabilizam o agendamento de energia e flexibilidade (Fonseca et al., 2024). O seu propósito é oferecer um controlo inteligente da flexibilidade energética em RECs. O sistema foca-se na diminuição de picos de demanda de energia, no intercâmbio de energia na rede e na otimização do uso de energia renovável.

É fundamental a adição de interfaces para este sistema devido ao facto de ser importante a visualização do funcionamento dos fluxos de energia ou outras variáveis relevantes e permitir ter uma forma intuitiva e de fácil aprendizagem de gerir os mesmos.

As interfaces acedidas pelo RECs Manager e pelo AI Engineer foram as desenvolvidas nesta dissertação, estando ambas presentes na solução implementada. As funcionalidades de cada uma destas foram divididas entre diferentes páginas, cujo acesso às mesmas está restringido ao seu respetivo utilizador. Estas interfaces possibilitam que os gestores das REC do sistema determinem em quais das RECs o sistema irá operar; caso contrário, os dados necessários para a previsão de flexibilidade não estariam acessíveis. Elas também possibilitam que os

engenheiros de IA utilizem previsões passadas do componente deste projeto para reeducar os algoritmos de programação e otimização da flexibilidade energética.

### 4.2.2 Nível 2

O segundo nível detalha o sistema em componentes, requeridos para disponibilizar os serviços que o EnergyAlze proporciona. Novamente, informações complementares, como a organização interna dos elementos, são deixadas de lado, apresentando apenas uma descrição funcional de cada recipiente e as conexões entre eles. (Cardoso et al., 2020)

#### 4.2.2.1 Vista Lógica

Na Figura 18 está representada a vista lógica de nível 2:

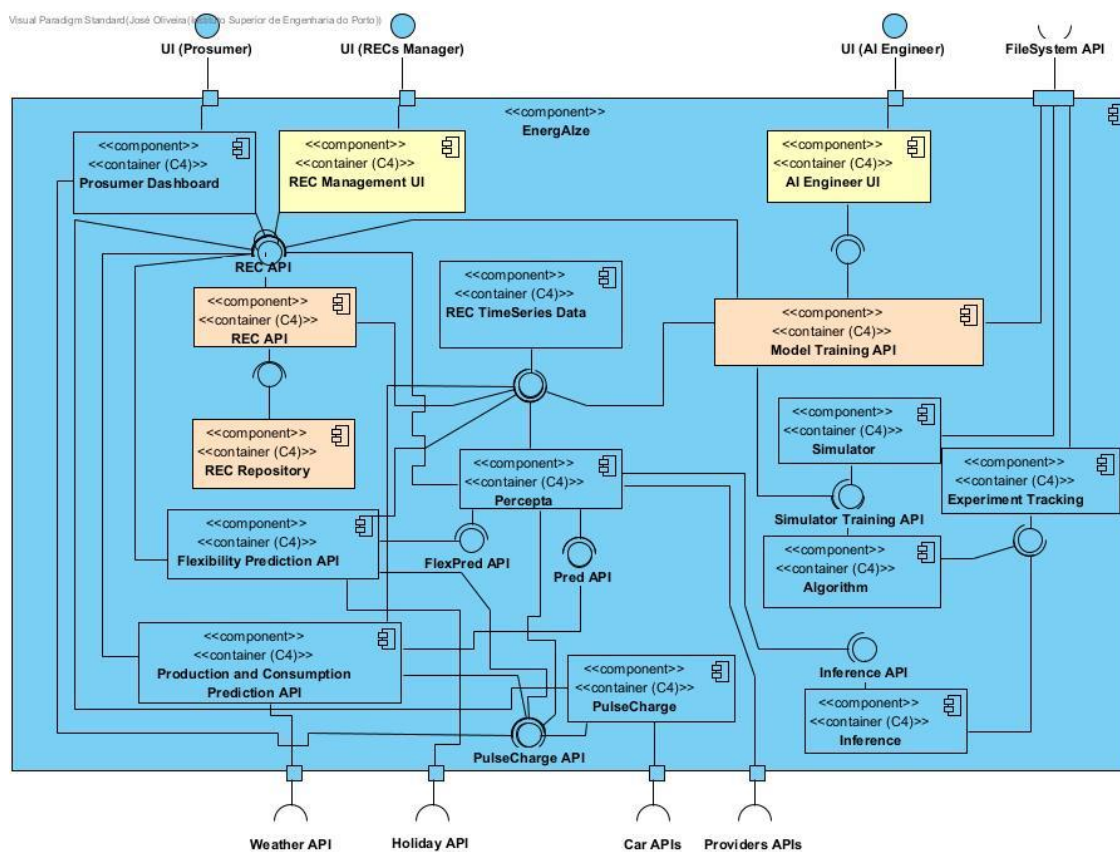


Figura 18 - Vista Lógica - Nível 2

Na Figura 18 é possível visualizar o projeto EnergyAlze em toda a sua extensão. Os componentes trabalhados pelo estudante, o ponto central da dissertação, REC Management UI e a AI Engineer UI, foram destacados com cor amarela

Outros componentes foram destacados a laranja devido ao facto de trabalharem em conjunto com a interface e terem sido feitas algumas ações para garantir o bom funcionamento da mesma:

- **REC API** – Este componente permite obter toda a logica da comunidade de energia, ou seja, o que são casas, que equipamentos tem, que utilizadores há, que comunidade é, etc. É aqui que é obtida toda a informação base da comunidade.
- **REC Repository** – Este componente é a base de dados que guarda toda a informação obtida na REC API. Na comunicação com a interface o objetivo deste componente é fornecer os dados de vários cenários para assim visualizar e analisar os mesmos.
- **Model Training API** – Este componente tem como objetivo treinar modelos, gerar *datasets*, fazer configurações de simulação, etc. No contexto da interface a comunicação com este é feita para a ajuda na definição de configurações de treino que serão posteriormente utilizadas para executar novas simulações.

#### 4.2.2.2 Vista Física

Na Figura 19 está representada a vista física de nível 2:

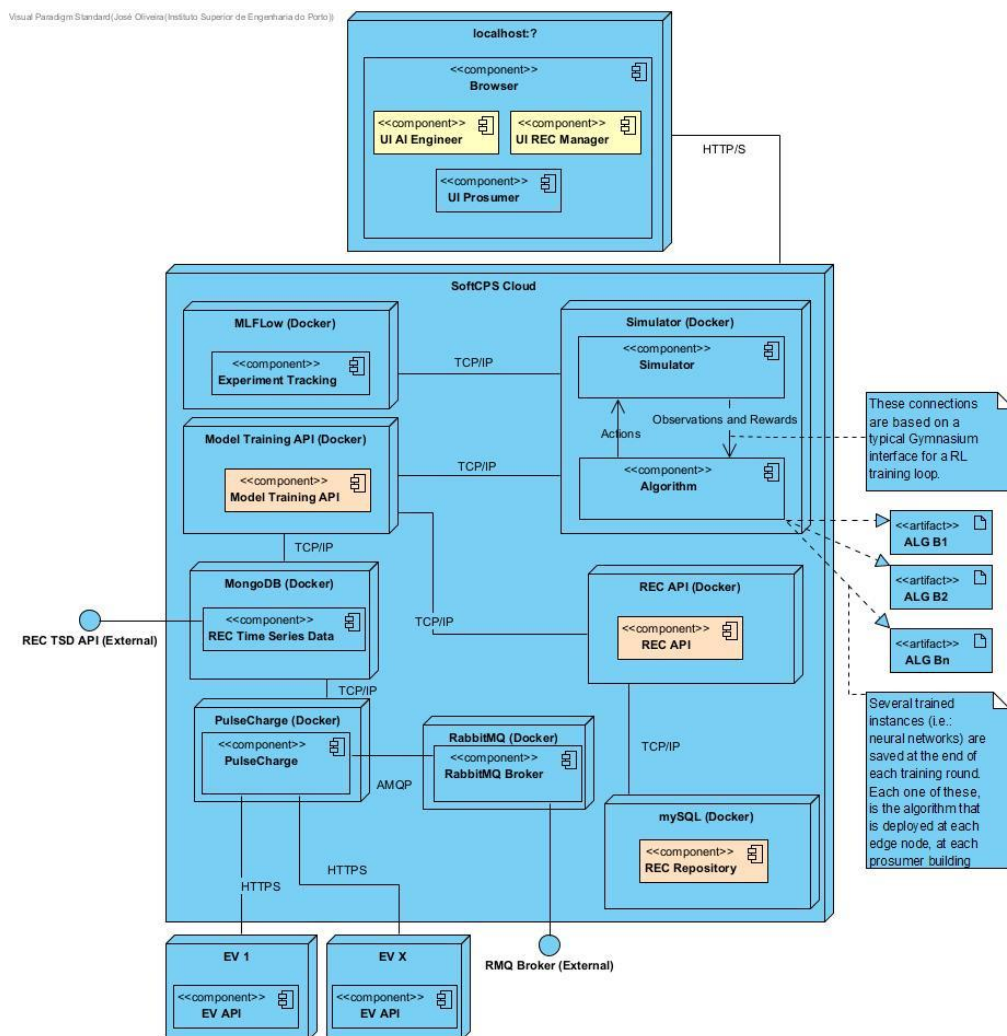


Figura 19 - Vista Física - Nível 2

Aqui encontram-se os componentes constituintes do projeto organizados pelos seus respetivos locais onde se encontram em funcionamento, seguindo a mesma regra de destaque que a vista lógica de nível 2. As interfaces podem ser acedidas pelos utilizadores localmente através do

browser ou acedendo pelo IP da máquina virtual onde se encontrar integrada, fazendo a ligação com os componentes necessários do projeto EnergyAlze, situados na *cloud* do grupo SoftCPS através de pedidos HTTP. Estes componentes por si mesmos estão contidos em contentores armazenados em Docker.

As interfaces estão também contidas numa das máquinas virtuais do ISEP, tendo sido criado um fluxo de implantação para a atualização da mesma sempre que fossem feitas alterações no código. O objetivo disto é garantir que a interface possa mais facilmente ser distribuída e escalável para futuramente ser integrável no CityLearn. Na secção 5.3 deste documento é descrito o processo de implantação com mais detalhe.

#### 4.2.2.3 Vista de Processos

Nesta vista são apresentados os casos de uso existentes através de um diagrama SSD para cada um que demonstra o fluxo da interação entre os utilizadores e o sistema.

#### **UC1 - Visualização de dados das simulações das comunidades de energia**

A Figura 20 demonstra o SSD com o fluxo para a visualização dos dados:

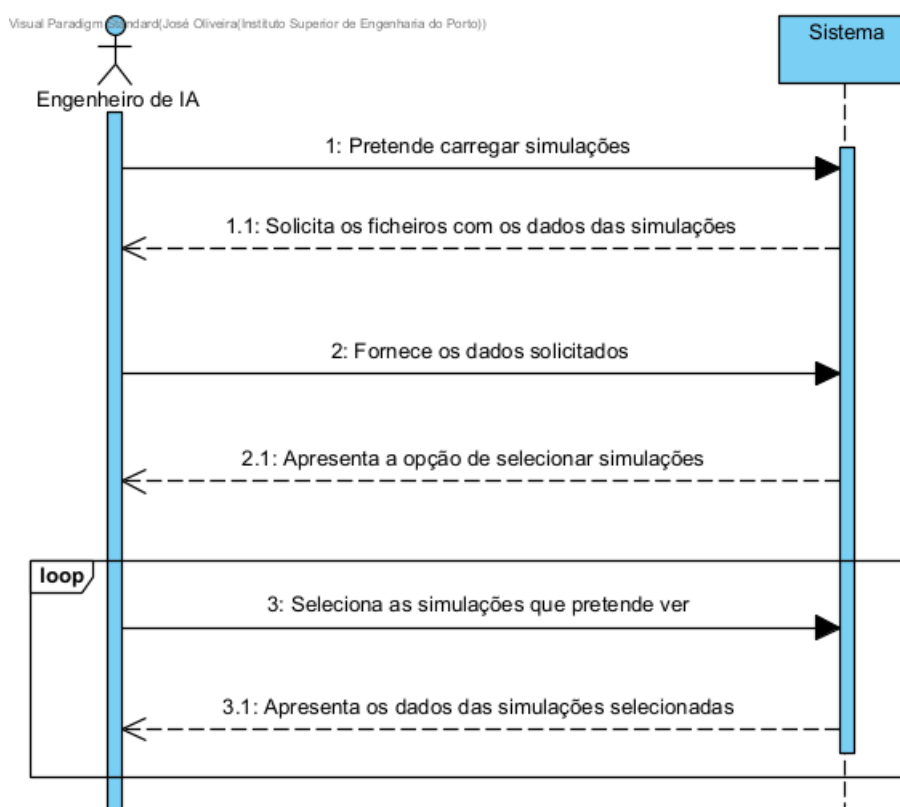


Figura 20 - SSD: Visualização de dados das simulações das comunidades de energia

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA começa o processo ao carregar numa opção para fazer o upload de dados de simulações. Após selecionar a pasta com os dados necessários, o sistema apresenta ao utilizador uma opção para selecionar as simulações que pretende visualizar. O engenheiro

pode agora seleccionar quais são as simulações que pretende visualizar e confirmar a seleção. O sistema apresenta por fim um conjunto de separadores, um por cada simulação selecionada, e dentro de cada um destes, são apresentados os dados dos componentes presentes na simulação correspondente. Este processo de seleção e apresentação dos resultados pode ser repetido as vezes que o engenheiro quiser até o mesmo sair da página.

**Pré-condições:** Devem existir dados válidos para visualização

**UC2 - Visualização dos KPIs das simulações e comparação dos mesmos**

A Figura 21 demonstra o SSD com o fluxo da visualização e comparação de KPIs:

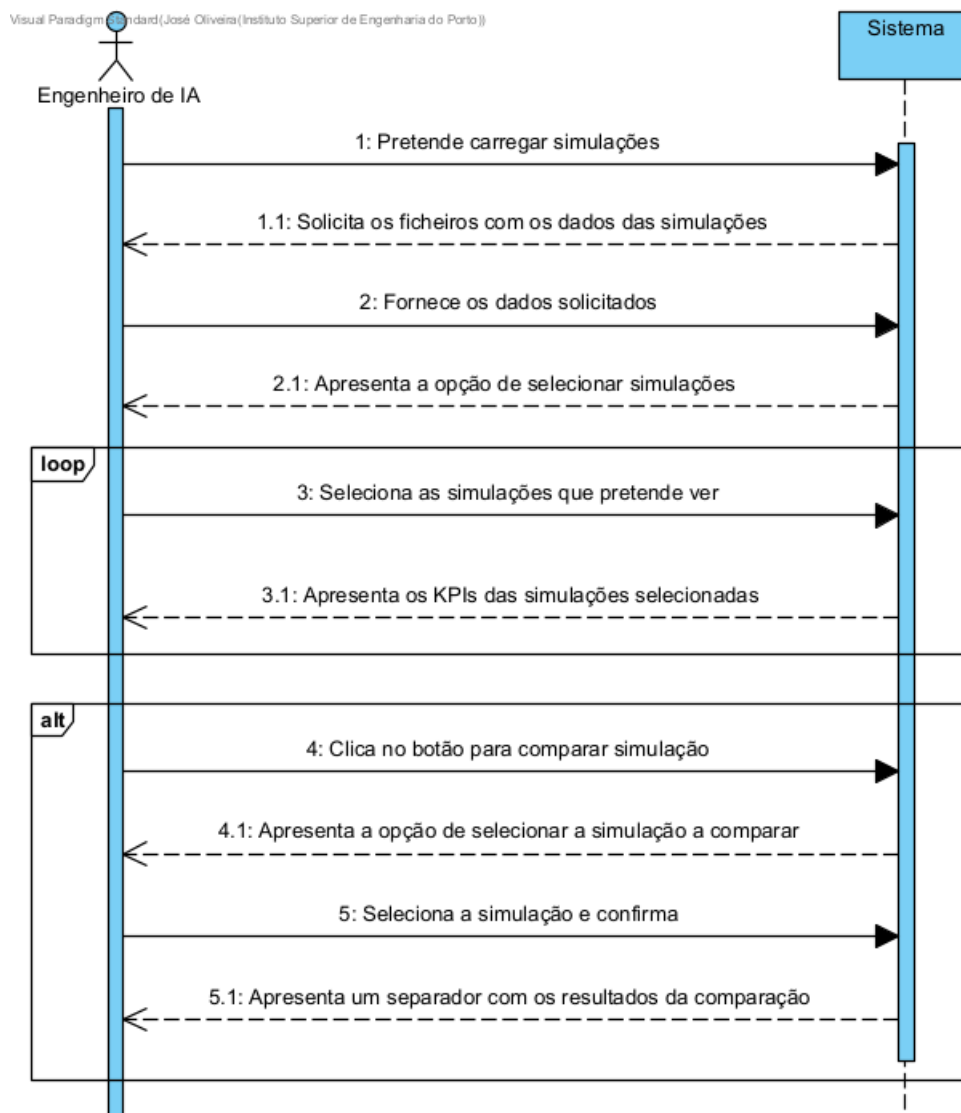


Figura 21 - SSD: Visualização dos KPIs das simulações e comparação dos mesmos

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA começa o processo ao carregar numa opção para fazer o upload de dados de simulações. Após seleccionar a pasta com os dados necessários, o sistema apresenta

ao utilizador uma opção para seleccionar as simulações que pretende visualizar. O engenheiro pode agora seleccionar quais são as simulações que pretende visualizar e confirmar a selecção. O sistema apresenta por fim um conjunto de separadores, um por cada simulação seleccionada, e dentro de cada um destes, são apresentados os dados dos KPIs da simulação correspondente. Este processo de selecção e apresentação dos resultados pode ser repetido as vezes que o gestor quiser até o mesmo sair da página. Para além disso, o utilizador tem a opção de comparar os resultados entre comparações. Ao seleccionar a opção de “Comparar” de uma simulação, este poderá seleccionar a simulação com a qual quer comparar KPIs e após confirmação, aparecerá um novo separador com os resultados da comparação.

**Pré-condições:** Devem existir dados válidos para visualização.

**UC3 - Criação de um ficheiro de configuração do algoritmo**

A Figura 22 demonstra o SSD com o fluxo do processo de criação do ficheiro:

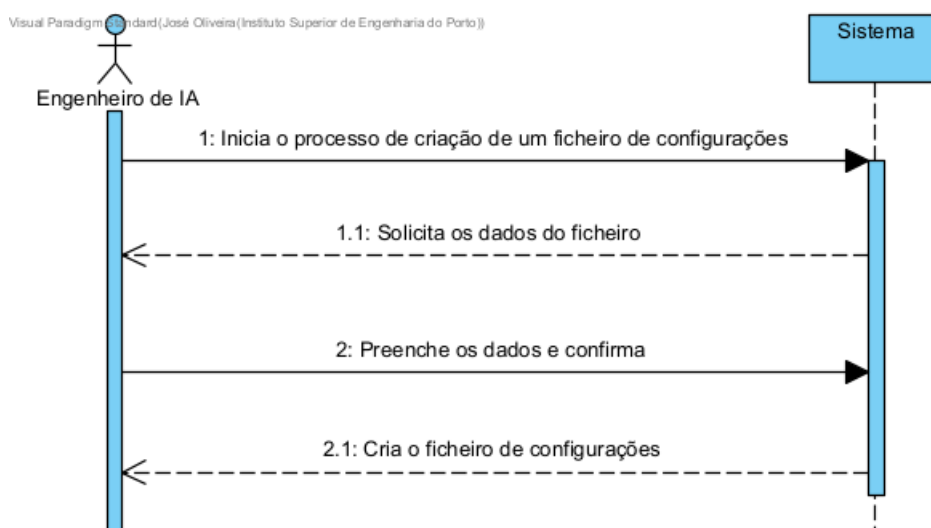


Figura 22 - SSD: Criação de um ficheiro de configuração do algoritmo

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA começa o processo de criação de um ficheiro de configurações. O sistema solicita um conjunto de dados acerca do mesmo. O engenheiro preenche os dados solicitados e confirma os mesmos. Após a validação dos dados inseridos, o ficheiro de configurações é criado e o utilizador recebe uma notificação a confirmar o sucesso da operação.

**Pós-condições:** O ficheiro de configurações é criado e o utilizador é notificado.

**UC4 - Criação de *dataset* e ficheiro de configuração de treino**

A Figura 23 demonstra o SSD com o fluxo da criação do *dataset*:

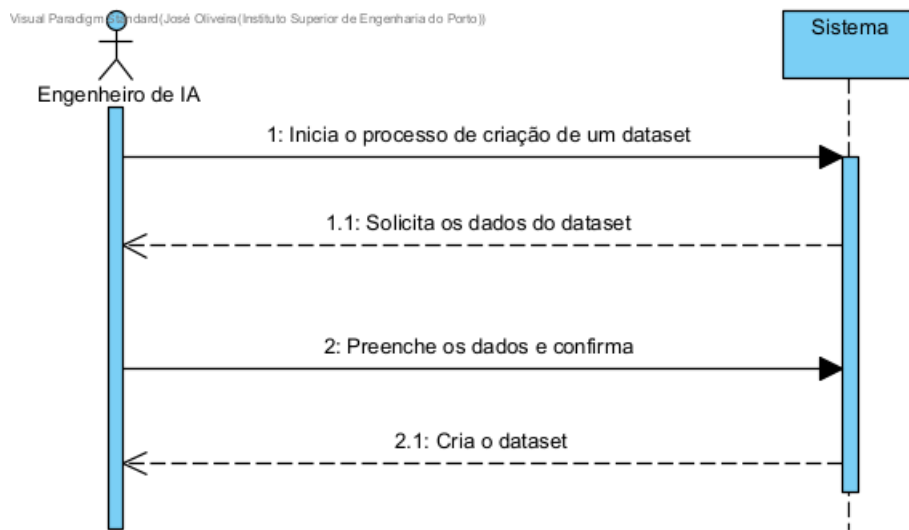


Figura 23 - SSD: Criação de *dataset* e ficheiro de configuração de treino

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA começa o processo de criação de um *dataset*. O sistema solicita um conjunto de dados acerca do mesmo. O engenheiro preenche os dados solicitados e confirma os mesmos. Após a validação dos dados inseridos, o *dataset* é criado e o utilizador recebe uma notificação a confirmar o sucesso da operação.

**Pós-condições:** O *dataset* é criado e o utilizador é notificado.

**UC5 - Execução das simulações com base em *schemas*, configurações e *datasets***

A Figura 24 demonstra o SSD com o fluxo do processo de execução das simulações:

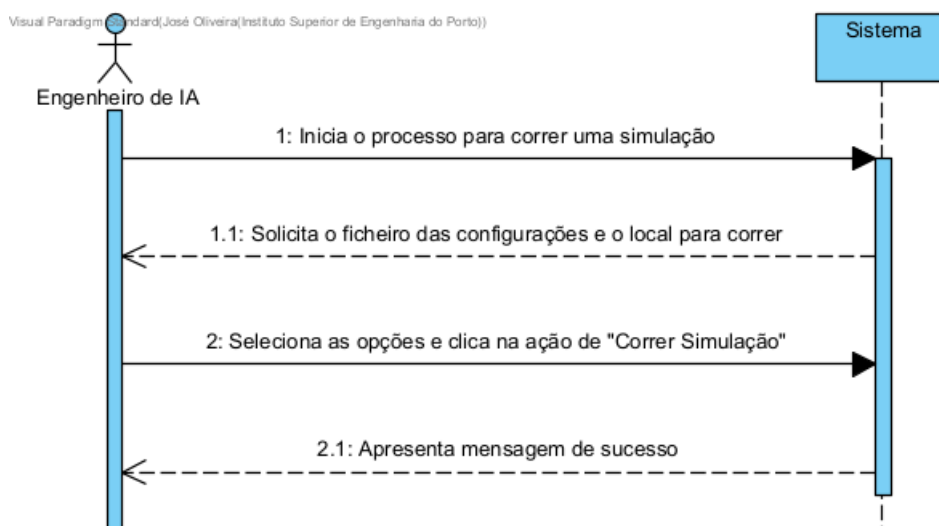


Figura 24 - SSD: Execução das simulações com base em *schemas*, configurações e *datasets*

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA inicia o processo entrando na página respetiva para correr simulações. O sistema solicita a escolha do ficheiro de configurações a utilizar e o local onde será colocada a simulação a correr. O engenheiro escolhe o ficheiro de configurações a utilizar e o local e confirma o processo clicando no botão “Run Simulation”. É então criada uma simulação com base nos dados do ficheiro seleccionado e a mesma é colocada a correr num contentor situado no local indicado sendo o utilizador notificado do sucesso da operação.

**Pré-condições:** Deve existir um ficheiro com informações relevantes para criar a simulação.

**Pós-condições:** A simulação fica a correr no contentor indicado e o utilizador é notificado.

### **UC6 - Configuração da comunidade de energia**

A Figura 25 demonstra o SSD com o fluxo do processo de configuração:

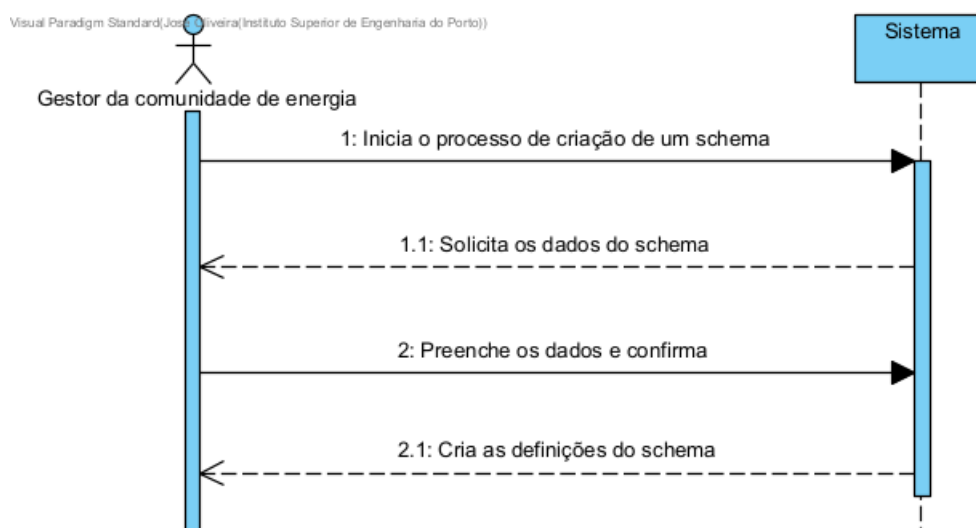


Figura 25 - SSD: Configuração da comunidade de energia

**Ator Principal:** Gestor da comunidade de energia

**Descrição:** O gestor da comunidade de energia começa o processo de criação de um *schema*. O sistema solicita então um conjunto de dados acerca do mesmo. O gestor preenche os dados solicitados e confirma os mesmos. Após a validação dos dados inseridos, as definições do *schema* são criadas sendo o utilizador notificado do sucesso da operação.

**Pós-condições:** As definições do *schema* são criadas e o utilizador é notificado.

### **UC7 - Visualização de dados reais em cenários diferentes**

A Figura 26 demonstra o SSD com o fluxo da visualização dos dados:

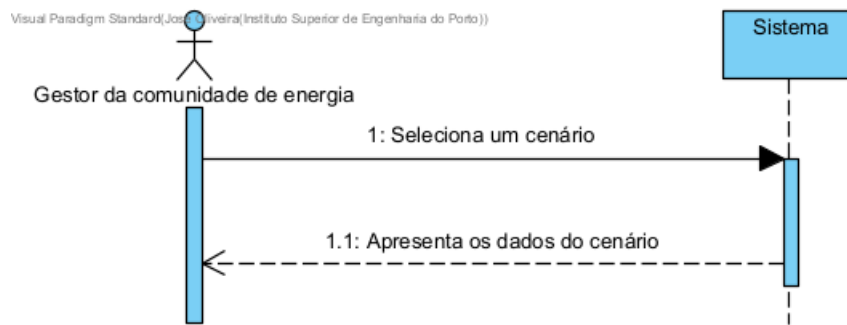


Figura 26 - SSD: Visualização de dados reais em cenários diferentes

**Ator Principal:** Gestor da comunidade de energia

**Descrição:** O gestor da comunidade de energia ao entrar na página pode escolher entre os cenários disponíveis, cada um deles estando dentro de um separador próprio. O sistema faz a ligação com a base de dados e faz um pedido conforme o separador onde o utilizador se encontrar para obter os dados do cenário em questão.

**Pré-condições:** Deve haver ligação com a base de dados onde se encontram os dados dos cenários

**UC8 - Personalização da vista dos dados**

A Figura 27 demonstra o SSD com o fluxo da personalização da vista dos dados:

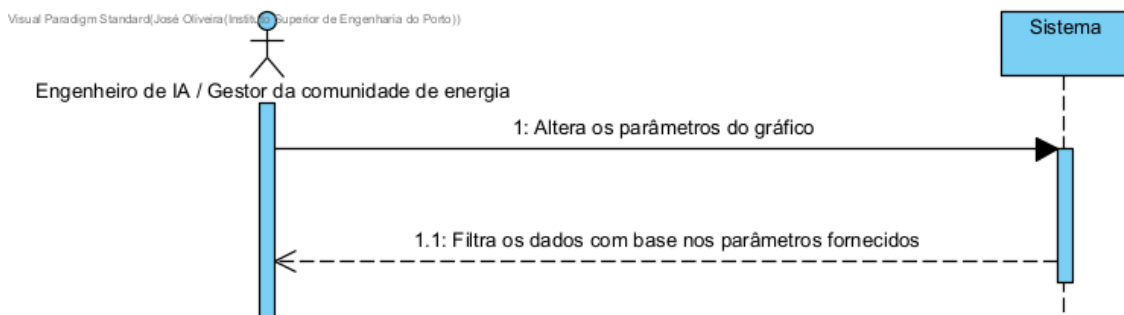


Figura 27 - SSD: Personalização da vista dos dados

**Ator Principal:** Engenheiro de IA / Gestor da comunidade de energia

**Descrição:** O engenheiro/gestor altera os parâmetros de um gráfico seja o intervalo de tempo entre cada valor, o intervalo de datas dos quais o mesmo quer ver informação ou até mesmo quais os dados em si que estão visíveis e o sistema faz a filtragem dos dados consoante os parâmetros indicados pelo utilizador, apresentando em tempo real o gráfico com as devidas alterações.

**Pré-condições:** O gráfico deve estar presente e visível na página.

### 4.2.3 Nível 3

Os contentores, por sua vez, consistem em componentes, que é a perspetiva na qual o modelo C4 se foca no terceiro nível. A cooperação entre os vários componentes deve garantir o serviço ou a funcionalidade do contentor mencionada no segundo nível do modelo C4. (Cardoso et al., 2020)

#### 4.2.3.1 Vista Lógica

Na Figura 28 está representada a vista lógica de nível 3:

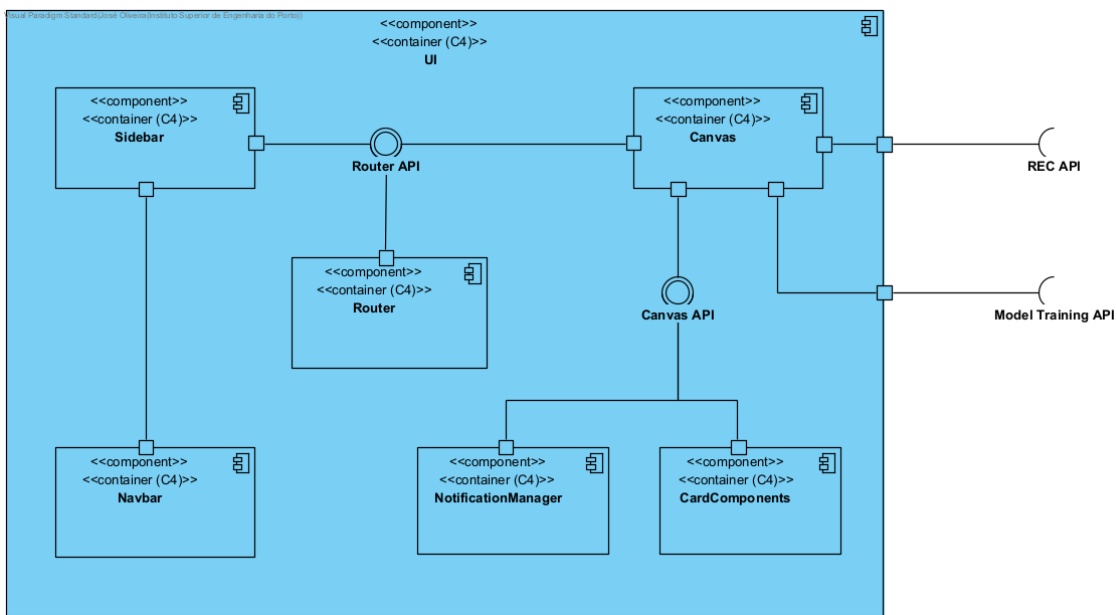


Figura 28 - Vista Lógica - Nível 3

Aqui é feita uma representação da estrutura da interface. Como tanto a interface destinada ao engenheiro de IA e a destinada ao gestor da comunidade de energia possuem as mesmas características, estão ambas representadas pelo mesmo diagrama.

A interface na sua base é constituída por um menu lateral, sendo este o componente da Sidebar com a lista das páginas às quais o utilizador pode aceder, o componente da Navbar que representa a secção em que se pode ver o nome da página atual e futuramente terá opções de *login/logout* e pelo componente do Canvas, que representa a zona central que irá refletir o conteúdo da página selecionada pelo utilizador no menu lateral. Este componente sendo o central da interface é também o que comunica com as APIs responsáveis pela comunicação com a base de dados e pela criação das simulações.

O componente do Router é o que funciona como elo de ligação entre a Sidebar e o Canvas, tendo a responsabilidade de fazer o roteamento entre as páginas.

Os componentes NotificationManager e CardComponents estão também associados ao Canvas, sendo o primeiro responsável por tratar das notificações de sucesso e erro ao efetuar

determinadas operações na interface e o segundo representa os componentes que contém os cartões com os gráficos interativos que estão presentes nas secções de visualização de dados.

#### 4.2.3.2 Vista de Processos

Relativamente à vista de processos do nível 3, foram escolhidos alguns dos requisitos funcionais para elaborar em mais detalhe um diagrama de sequência de forma a demonstrar para além do funcionamento do fluxo, as interações com outros componentes do projeto OPEVA. Para além disso estão descritas aqui algumas das decisões de design tomadas ao longo do tempo para alguns dos componentes feitos para os requisitos funcionais.

#### **Visualização de dados das simulações das comunidades de energia**

A principal decisão de design a discutir para este requisito foi relativa à melhor forma de apresentar os gráficos dos diversos elementos em conjunto com a lista lateral dos mesmos. Inicialmente foi pensado adicionar vários gráficos e apresentar todos estes acompanhados por um *scroll* vertical de forma a ter acesso a todos os gráficos de uma vez ao selecionar determinado edifício como demonstra a Figura 29.

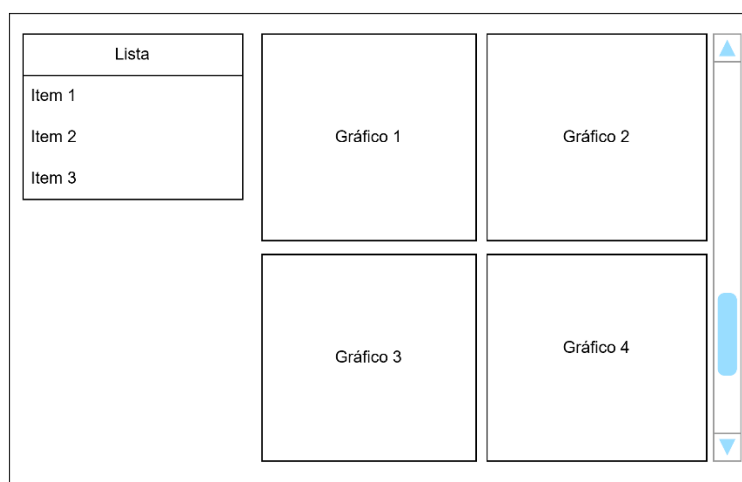


Figura 29 - Estrutura inicial para a representação de gráficos

No final foi decidido que essa alternativa iria pôr em causa a qualidade e a visibilidade dos dados devido ao facto de não serem facilmente perceptíveis as informações o que não cumpre com o requisito RNF10. Acabou-se por adotar uma estrutura com menos gráficos cujo conteúdo deve ser selecionado ao detalhe na lista lateral, mas que permite assim uma visualização mais clara e detalhada dos gráficos.

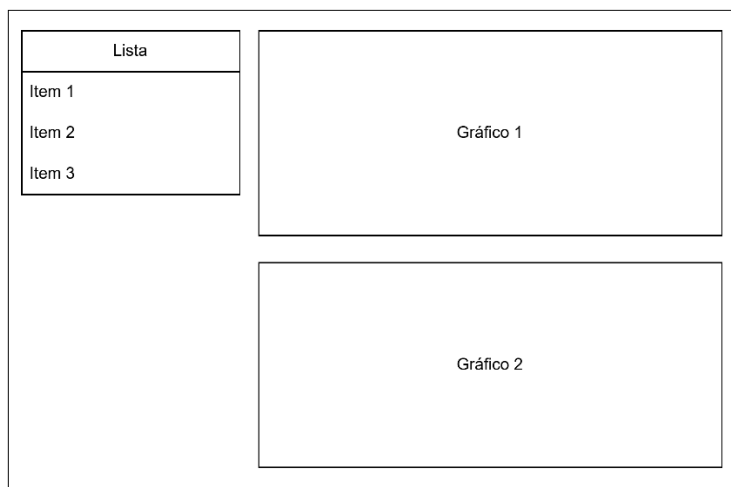


Figura 30 - Estrutura decidida para a representação de gráficos

### **Visualização dos KPIs das simulações e comparação dos mesmos**

Para este requisito uma das alternativas de design posta em questão foi a melhor forma de apresentar os resultados das comparações. Foi decidido utilizar uma tabela, tal como é feito para apresentar os resultados dos KPIs devido a ser mais fácil de interpretar os resultados das comparações pois tendo um formato similar ao original, torna-se mais simples detetar os pontos onde se pode comparar os resultados.

A outra opção posta em causa foi a utilização de um gráfico. No entanto, esta ideia foi descartada pela quantidade elevada de dados disponíveis para comparação, levando a que o gráfico de resultados tivesse um grande número de variáveis a apresentar tornando difícil de ler e analisar. Mesmo utilizando lógica como limitar a visibilidade de certos resultados, continua a ser uma forma de apresentação pouco pratica e menos legível.

### **Criação de um ficheiro de configuração do algoritmo**

Para este requisito não foram tomadas quaisquer decisões de design.

### **Criação de dataset e ficheiro de configuração de treino**

Para as ações e observações foi implementada uma simples lista com *checkboxes* para cada um dos valores das ações e observações existentes. Para as ações isto resolvia a situação, mas para as observações implicava problemas para a seleção de quais destas se encontravam partilhadas.

Uma solução possível era a criação de uma terceira lista com os mesmos valores das observações, mas com o objetivo de selecionar apenas se seriam partilhadas ou não. Esta ideia acabou por ser descartada pois um dos objetivos era permitir ter ambas as funcionalidades na mesma lista de forma a evitar criar componentes adicionais. A solução final foi a criação de uma estrutura de tabela com colunas para o nome da observação, uma coluna com *checkboxes* para marcar as ativas e outra para marcar as partilhadas. Esta estrutura foi usada para as ações mesmo não sendo necessária de forma a cumprir com o requisito RNF12.

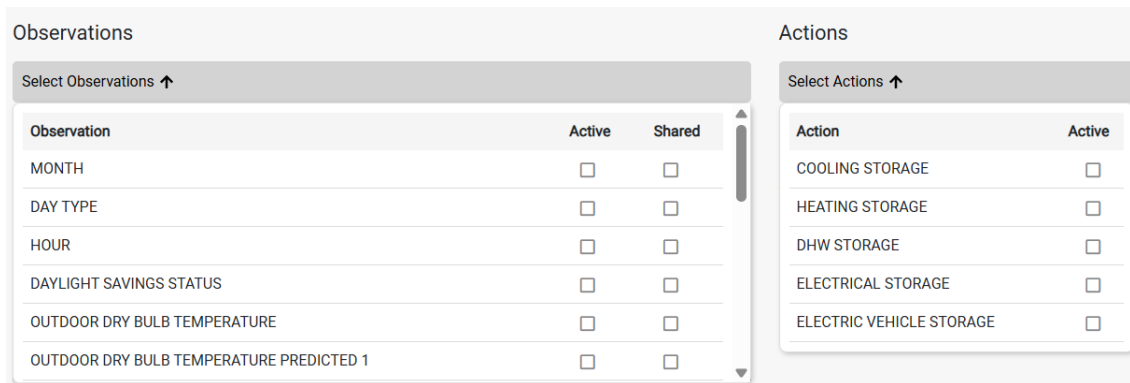


Figura 31 - Estrutura para seleção de observações e ações

### Execução das simulações com base em *schemas*, configurações e *datasets*

O diagrama da Figura 32 representa as interações adicionais feitas com outros componentes do projeto, as quais são essenciais para o funcionamento deste requisito.

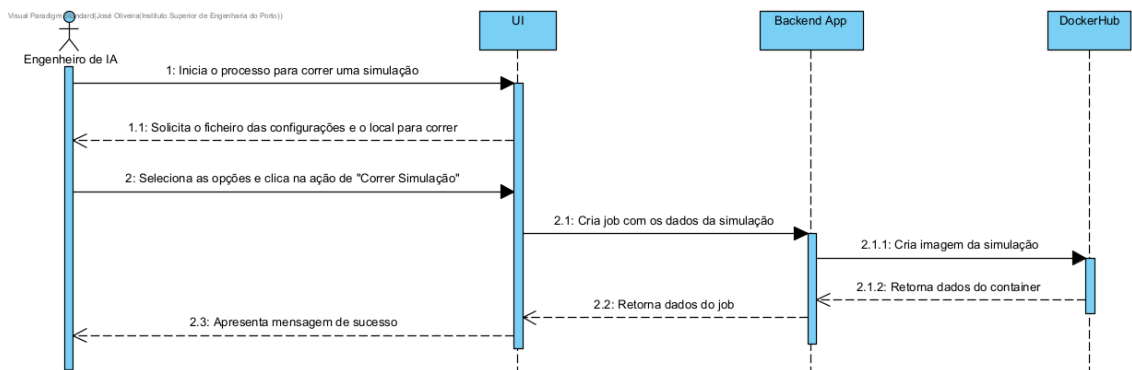


Figura 32 - SD: Execução das simulações com base em *schemas*, configurações e *datasets*

**Ator Principal:** Engenheiro de IA

**Descrição:** O engenheiro de IA inicia o processo entrando na página respetiva para correr simulações. O sistema solicita a escolha do ficheiro de configurações a utilizar e o local onde será colocada a simulação a correr. O engenheiro escolhe o ficheiro de configurações a utilizar e o local e confirma o processo clicando no botão "Run Simulation". É chamado um *endpoint* para a aplicação de *backend* e na mesma é criado um job com os dados da simulação. A partir desse job é criada a imagem da simulação e a mesma é colocada a correr num contentor situado no local indicado. Finalmente é apresentada ao engenheiro uma mensagem de sucesso a indicar que a simulação se encontra a correr.

**Pré-condições:** Deve existir um ficheiro com informações relevantes para criar a simulação.

**Pós-condições:** A simulação fica a correr no contentor indicado.

### Configuração da comunidade de energia

Relativamente a este requisito, o ponto no qual houve decisões de design foi na forma de representar os valores da curva de eficiência para cargas e descargas no componente dos carregadores.

Inicialmente foi considerado simplificar o processo e escrever os valores de carga/descarga numa caixa de texto. No entanto, acabou por se provar pouco intuitivo para os utilizadores, devido também ao tamanho reduzido da janela, sendo pouco prático para ler os valores na sua totalidade e decidiu-se tomar uma alternativa mais clara. Foi feita então uma estrutura estilo tabela com dois valores (mínimo e máximo) para cada um dos intervalos. Assim sempre que o utilizador pretender adicionar uma nova linha com um intervalo clica no botão “Add Row” ou se pretender remover clica no botão do lado direito para remover essa linha.

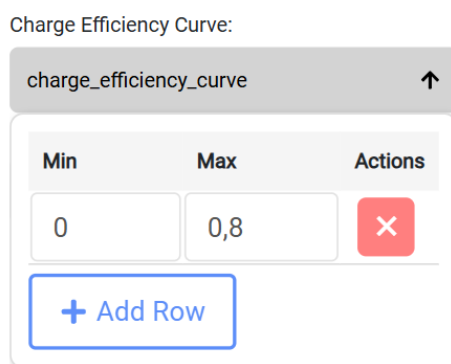


Figura 33 - Componente para array de pares

### Visualização de dados reais em cenários diferentes

Para este requisito foi seguida a mesma estrutura definida para a visualização dos dados das simulações das comunidades de energia pois possuem uma lógica similar, fazendo sentido seguir a mesma estrutura de forma a garantir consistência no design, respeitando assim o RNF12.

O diagrama da Figura 34, representa a interação adicional feita com outros componentes do projeto, a qual é essencial para o funcionamento deste requisito.

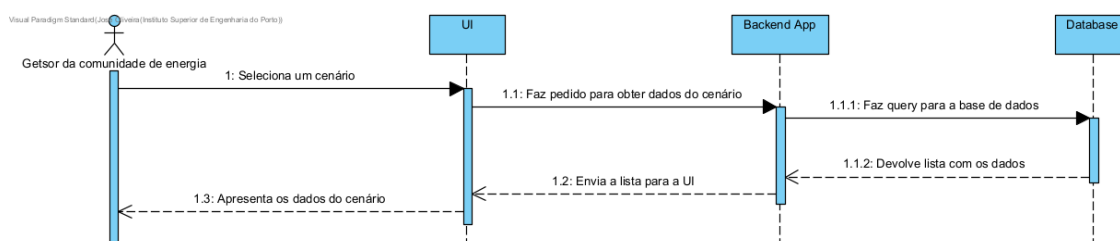


Figura 34 - SD: Visualização de dados reais em cenários diferentes

**Ator Principal:** Gestor da comunidade de energia

**Descrição:** O gestor da comunidade de energia seleciona um cenário do qual o utilizador pretende visualizar informação. Após a seleção ser feita, na UI é apresentado um ícone de

*loading* enquanto é feito um pedido HTTP para a aplicação em *backend*. A aplicação recebe o pedido e com base no cenário a visualizar, faz uma *query* para a base de dados respetiva em MongoDB. A base de dados envia então os dados pedidos e a aplicação em *backend* envia estes de volta para a UI. Finalmente na UI são recebidos os dados do cenário e é construída a estrutura que permitirá visualizar e analisar os mesmos.

**Pré-condições:** Deve existir uma ligação à base de dados.

### **Personalização da vista dos dados**

Para este requisito uma das questões que surgiram foi relativo à forma de representação do *slider* para filtragem da data inicial e final. A ideia inicial e a que foi utilizada no final foi a criação de um *slider* no qual o utilizador pode arrastar ambas pontas para definir a data inicial e final.

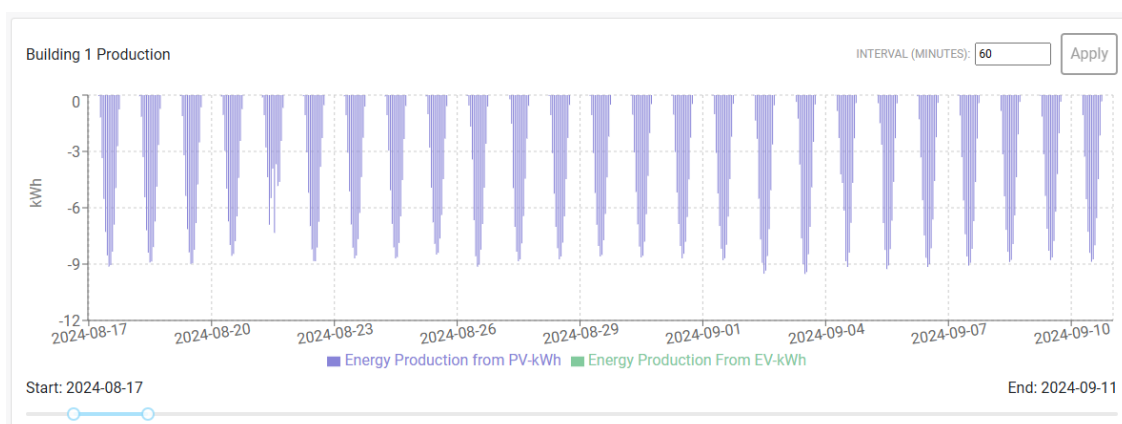


Figura 35 - Gráfico com *slider*

Alternativamente, foi considerada a utilização de um elemento Brush presente na biblioteca Recharts como mostra a Figura 36, mas devido à forma como o elemento funciona em conjunto com o gráfico e ao tipo de dados usados para representar o eixo do X sendo este *strings* com o valor das datas, não era praticável a sua utilização.

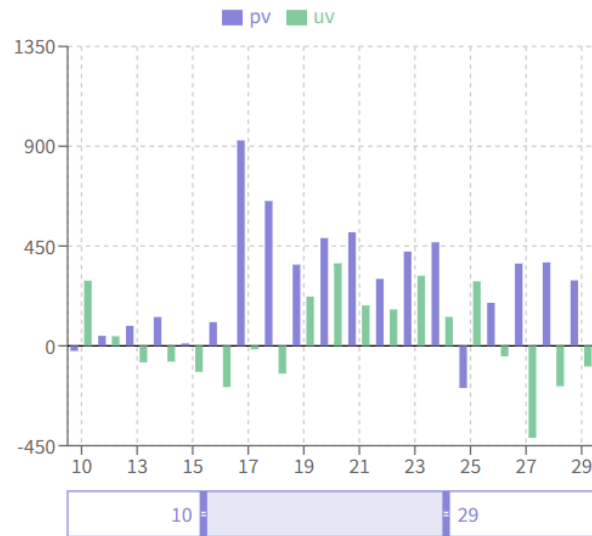


Figura 36 - Gráfico com o componente Brush

Outra questão pertinente foi decidir a forma de representar os possíveis intervalos de escolha na alteração de intervalos. Foi considerado utilizar uma lista com todas as possíveis opções, mas isso acabou por ser descartado pois certos casos tinham um grande número de opções possíveis. De forma a dar total liberdade ao utilizador de escolher o intervalo a ver, foi decidida a utilização de uma caixa de texto numérica com um botão para aplicar as alterações.



Figura 37 - Filtro de intervalos

#### 4.2.4 Nível 4

Finalmente, neste nível, cada um dos elementos é expandido para demonstrar como é codificado, utilizando diagramas de classes UML, diagramas de relações entre entidades, estrutura de código ou outros semelhantes. Esse nível não é obrigatório e pode ser aplicado somente a componentes extremamente complexos que requerem uma descrição mais detalhada. (Jokste & Kampars, 2024)

Para este projeto em questão, não existe complexidade suficiente para justificar a representação de diagramas neste nível.

## 5 Implementação da Solução

Neste capítulo é descrito todo o processo para a implementação da solução proposta ao problema. Começa com uma descrição sucinta da estrutura da organização do projeto e os seus componentes, de seguida é feita uma descrição da estrutura da interface em si e do funcionamento desta passando por uma descrição de cada uma das páginas constituintes dela explicando o seu propósito e como o utilizador pode interagir com ela. Finalmente é falado acerca do processo de implantação desenvolvido para disponibilizar a interface externamente e são abordadas as avaliações das métricas de acessibilidade e os testes feitos para garantir a qualidade e integridade da interface.

### 5.1 Estrutura do projeto

Para ajudar o estudante a elaborar de forma mais eficiente uma interface que permita alcançar todos os requisitos pretendidos, esta baseia-se num *template* de React com a adição de Bootstrap integrado chamado “React Light Bootstrap Dashboard”. Este *template* ofereceu uma base sólida sobre a qual elaborar o resto da interface tornando-a facilmente maleável e escalável.

A estrutura de pastas que engloba o projeto encontra-se organizada como demonstra a Figura 38:

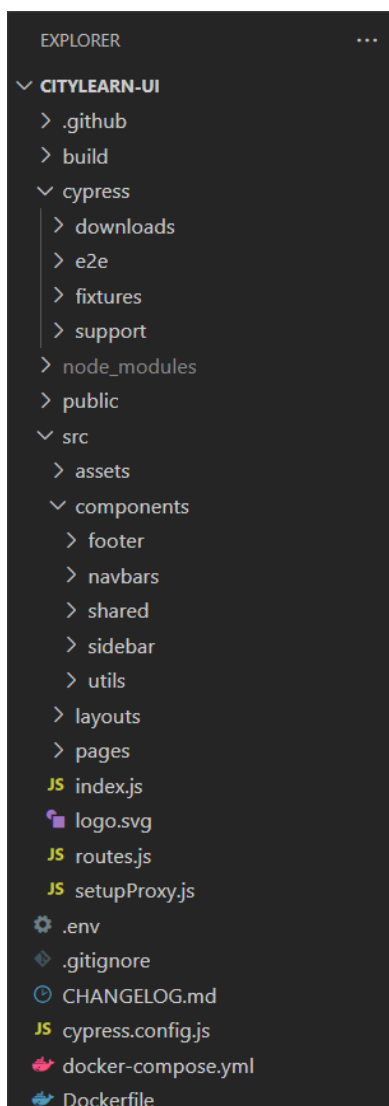


Figura 38 - Estrutura do projeto

Algumas das pastas e ficheiros presentes nomeadamente as pastas “layouts”, “assets” em alguns elementos na pasta “components” como os referentes à barra lateral, barra superior (também conhecida como *navbar*) e footer já faziam parte do *template* original tendo como objetivo principal estabelecer uma base para a construção da interface inicial e a partir daí, mais facilmente adicionar novas páginas e funcionalidades. Existe também um ficheiro dedicado ao roteamento que faz o elo de ligação entre a barra lateral e as páginas que serão visíveis na zona central.

Para facilitar o desenvolvimento do projeto foi feita uma divisão em pastas, cada uma reservada a componentes com determinadas funcionalidades para assim ser mais escalável e permitir um melhor entendimento da estrutura da interface.

Na pasta “pages” está contido o código relativo às páginas presentes na interface, estando cada uma dentro de uma outra pasta com o nome da página. Dentro dessa pasta existe um ficheiro

no formato .js com o componente da página e em certos casos, outros ficheiros que sejam essenciais para o funcionamento da página.

Na pasta “shared” estão contidos os componentes que são partilhados por outros componentes. Isto permite agrupar vários componentes numa só pasta em vez de repetir os componentes em vários locais.

Na pasta “utils” estão contidos os componentes que são utilizados como elementos auxiliares em algumas páginas. De certa forma são similares aos elementos na pasta “shared” mas têm usos mais particulares.

Na pasta “cypress” estão situados todos os ficheiros e pastas relativas ao processo de testes realizados no projeto.

Para além das pastas foi criado um ficheiro adicional chamado “setupProxy.js” cujo objetivo é de fazer a interface funcionar como proxy para principalmente resolver problemas relacionados com CORS que ocorreram durante o desenvolvimento. Este problema ocorria devido ao facto de aplicações web normalmente executarem requisições *cross-origin* HTTP ao solicitar um recurso que tenha uma origem diferente da sua própria origem, sendo este o caso durante o processo para fazer ligação com a base de dados externa em MongoDB.

## 5.2 Estrutura da interface

A interface segue um formato típico de um *dashboard* administrativo o qual é constituído por um menu lateral do lado esquerdo com a lista das páginas às quais o utilizador pode aceder, uma secção superior na qual se pode ver o nome da página atual e na secção inferior existe um footer com a marcação de copyright.

Devido a ser ainda uma fase inicial do desenvolvimento da interface, a ideia futura será ter múltiplos tipos de utilizador disponíveis e dar acesso a páginas diferentes. De momento para demonstrar essa intenção inicial, as páginas às quais cada tipo de utilizador terá acesso estão separadas pelo nome do tipo como demonstra a Figura 39.

Finalmente na zona central, ocupando a maior parte do ecrã, a zona que irá refletir o conteúdo da página selecionada pelo utilizador no menu lateral. Este pode ser considerado o núcleo da aplicação, onde a maioria das interações do utilizador irão acontecer.



Figura 39 - Estrutura base da interface

### 5.2.1 Página “REC Dashboards”

Esta é a página na qual o utilizador irá entrar sempre que abrir a aplicação inicialmente. É responsável pela visualização dos dados relativos a dados provenientes de simulações e corresponde ao requisito funcional RF1. O utilizador inicialmente seleciona a opção “Upload Simulations” que abre o explorador de ficheiros e pede como input uma pasta.

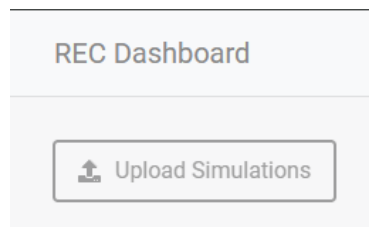


Figura 40 - Botão "Upload Simulations"

Essa pasta será a que tem os dados das simulações e deve seguir um formato específico. A pasta deve ter o nome “SimulationData” e no seu interior uma lista de outras pastas, cada uma com o nome relativo à simulação correspondente.

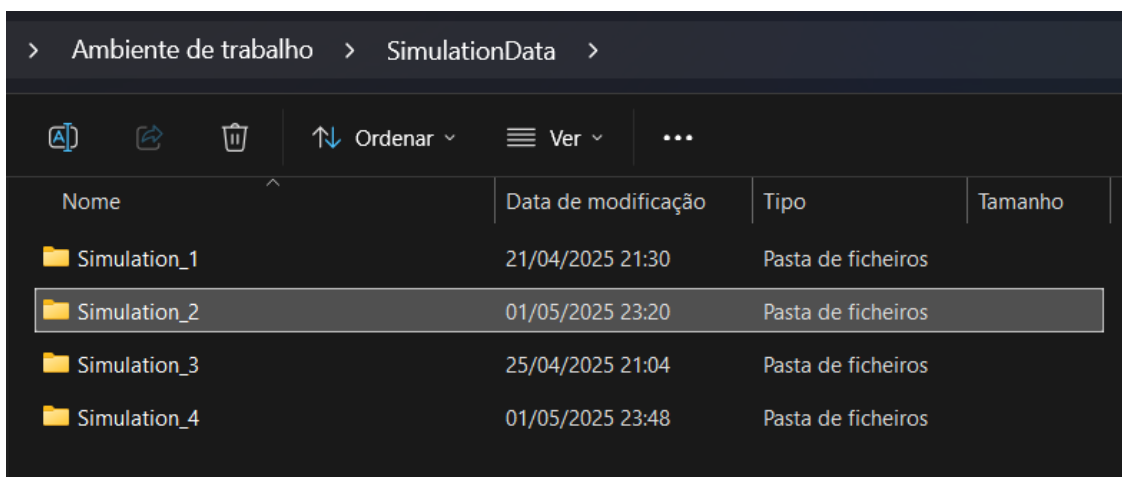


Figura 41 - Estrutura de pastas das simulações

Dentro de cada uma destas pastas está um conjunto de ficheiros no formato CSV, com os dados relativos aos edifícios, veículos elétricos, etc. É importante ter um especial cuidado com a nomenclatura para esses ficheiros de dados também. Deve seguir o nome “exported\_data\_building\_X” para definir os edifícios, seguido da definição do componente caso se refira a um, ou seja, “\_battery” ou “\_charger\_X\_Y” sendo X o número do edifício e Y o número do componente do edifício e terminando sempre com a definição do episódio “\_epZ”, sendo Z o número do episódio.

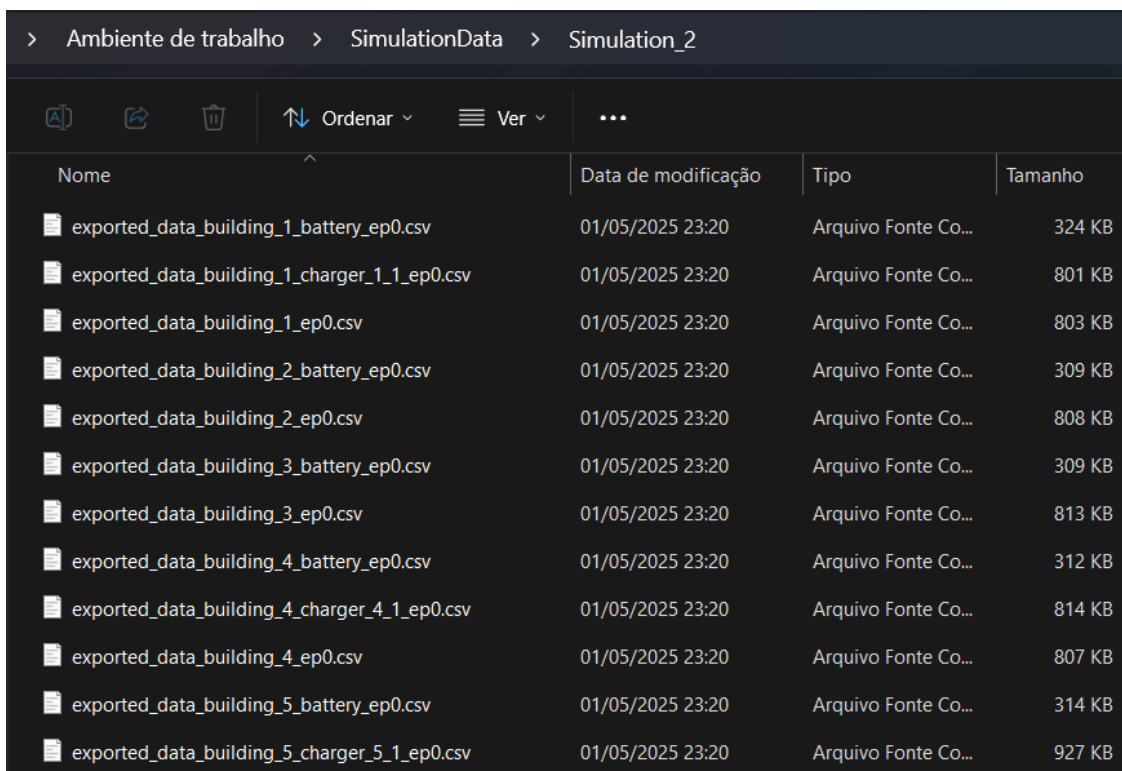


Figura 42 - Lista de ficheiros com os dados da simulação

Um ponto importante a ter em conta no upload da pasta com os dados é relativo ao facto de os dados ao estarem a ser importados a partir de uma pasta para o browser, por questões de segurança este pede sempre confirmação da ação do utilizador como está exemplificado na Figura 43.

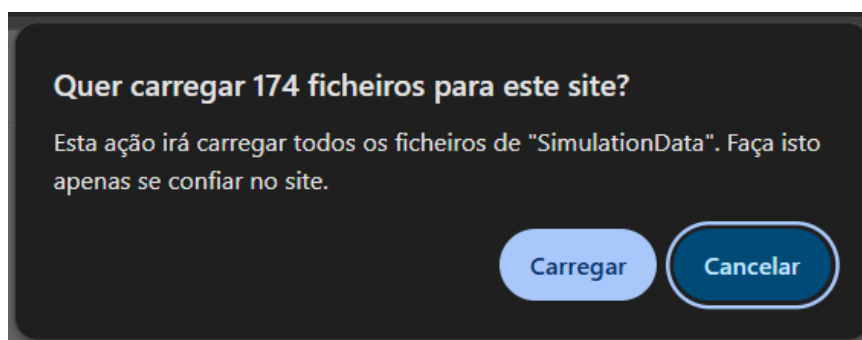


Figura 43 - Aviso do browser no upload de ficheiros

Após ser feito o upload da pasta com dados válidos, aparece do lado direito um botão “Select Simulations” que abre uma pequena janela onde o utilizador pode selecionar as simulações que pretende visualizar. O botão de confirmar fica inativo até ser selecionada pelo menos uma simulação.



Figura 44 - Janela com o botão "Select Simulations"

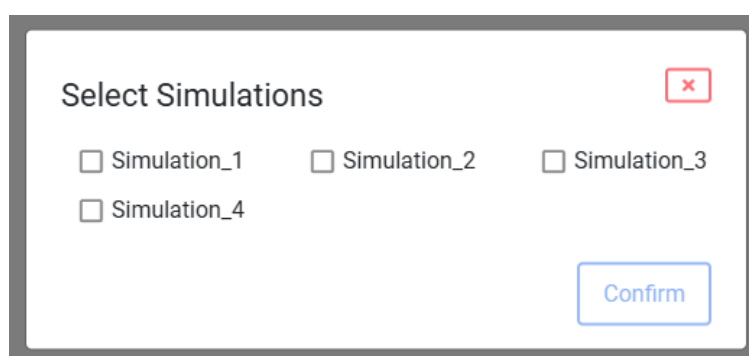


Figura 45 - Janela de seleção de simulações visíveis

Após a confirmação da escolha das simulações a visualizar, aparece no ecrã um conjunto de separadores, relativos a cada uma das simulações escolhidas.

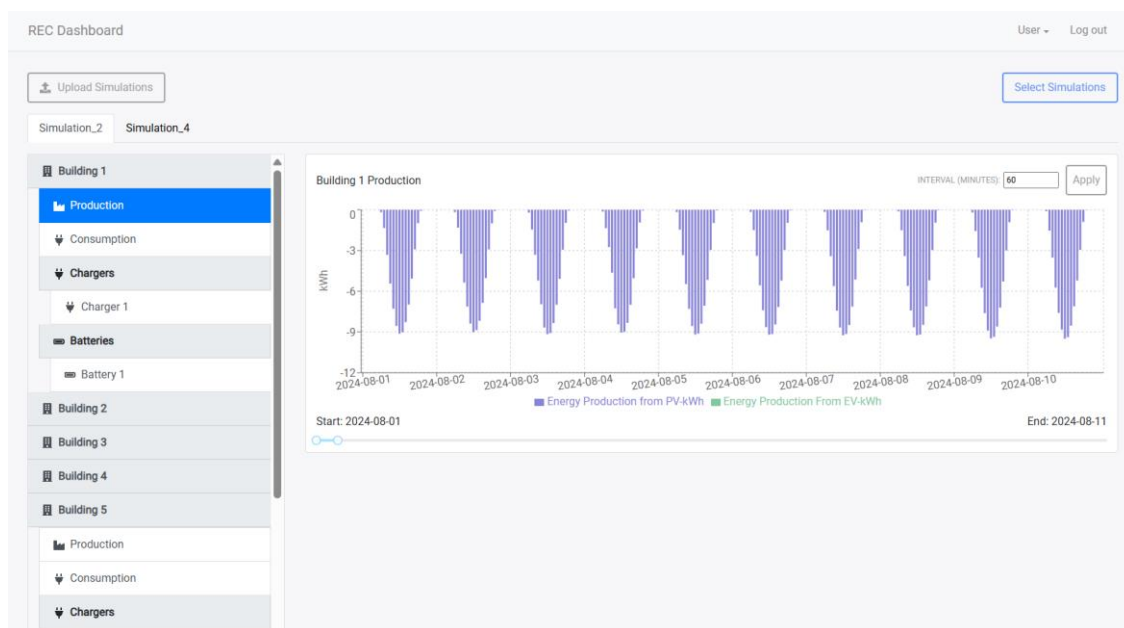


Figura 46 - Detalhes da simulação

Tal como demonstra a Figura 46, dentro de cada separador vemos uma lista do lado esquerdo do ecrã com o conjunto dos elementos presentes na simulação como os edifícios e dentro de cada edifício conseguimos ver um conjunto de itens relativos aos componentes do mesmo. Estes componentes estão divididos em: Consumption Production, Batteries e Chargers. Do lado direito do ecrã é possível ver o gráfico relativo ao componente selecionado.

Existem outros componentes como os veículos elétricos e os dados relativos a preços, com os seus respetivos gráficos colocados por baixo dos gráficos relativos aos edifícios como se pode ver na Figura 47.

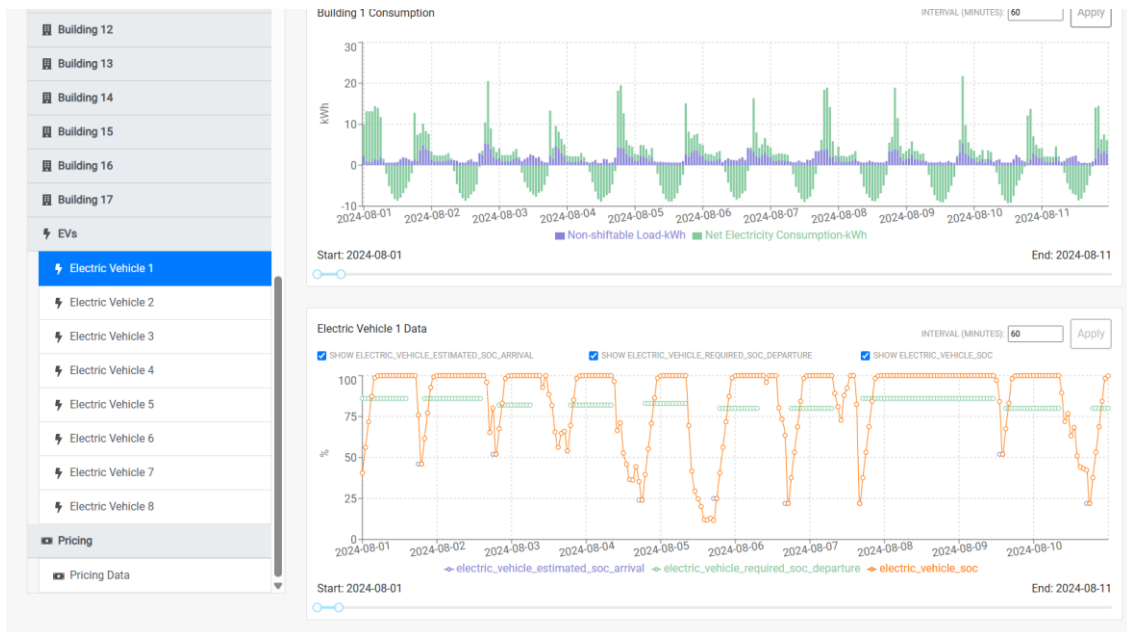


Figura 47 - Gráficos adicionais das simulações

Um ponto importante a ter em conta foi a forma como os gráficos deveriam estar posicionados. Devido ao nível de detalhe de alguns gráficos e de ter necessidade de manter os dados visíveis e fáceis de interpretar, foi decidido utilizar uma estrutura na qual o gráfico superior pode apresentar puramente dados relativos a consumo ou produção de um edifício e o gráfico inferior pode apresentar todos os outros. Assim reduz-se o número de gráficos presentes ao mesmo tempo para dois, sendo cada um deles grande o suficiente para demonstrar todos os dados necessário de forma a não ficarem apertados no ecrã. Foram feitas tentativas que permitissem apresentar mais do que dois gráficos, mas nenhuma delas garantia uma boa visibilidade da informação.

Tal como referido anteriormente, também há a necessidade de considerar os episódios existentes, pois estes referem-se a diferentes instâncias das simulações. Tendo isso em conta, nos casos de simulações com vários episódios acima da estrutura referida anteriormente foi colocada uma caixa para seleccionar qual o episódio que o utilizador que ver, para assim ser mais fácil de separar os resultados de episódios e permitir uma análise mais intuitiva dos dados sem sobrecarregar o utilizador com informação e tornar a análise mais confusa.

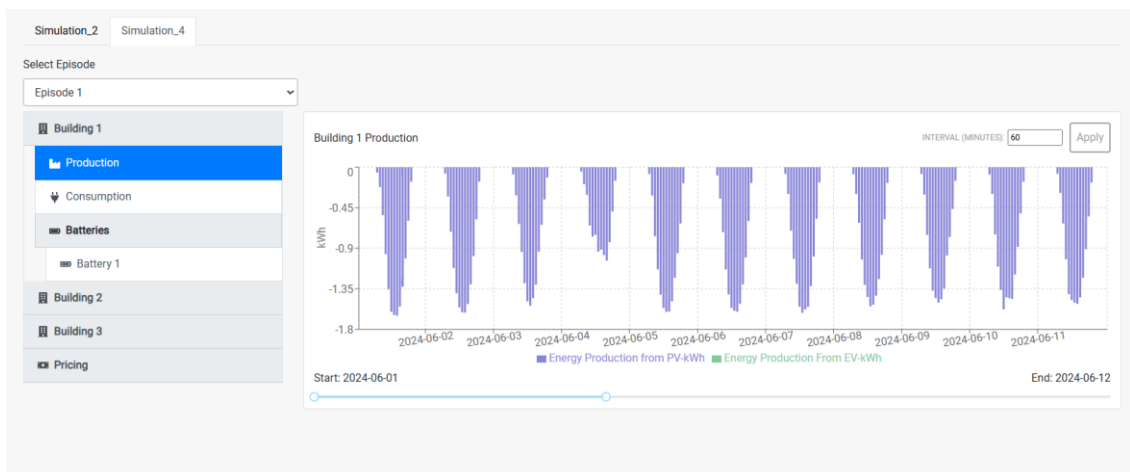


Figura 48 - Divisão dos dados em episódios

## 5.2.2 Página “Use Cases”

Esta página tem como objetivo mostrar ao utilizador dados reais provenientes de cenários diversos e corresponde ao requisito funcional RF7. Para isto foi feita uma ligação com uma base de dados externa em MongoDB a qual continha os dados relativos aos cenários disponíveis.

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
R-H-01	831.49 kB	27 K	194.00 B	2	1.54 MB
R-H-02	815.10 kB	27 K	187.00 B	2	1.54 MB
R-H-03	610.30 kB	18 K	192.00 B	2	1.02 MB
R-H-04	856.06 kB	27 K	194.00 B	2	1.54 MB
R-H-05	761.86 kB	27 K	194.00 B	2	1.54 MB

Figura 49 - Estrutura dos dados na base de dados MongoDB

Para fazer a ligação foi utilizado um proxy o qual permitiu fazer a ligação com a base de dados a partir do browser utilizando como intermédio um programa em *backend* com o propósito de estabelecer essa ligação. Após o pedido ser feito e os dados recebidos, estes aparecem no respetivo separador de cada cenário.

Tal como na página dos REC Dashboards, a estrutura dos dados é similar tendo uma lista lateral com a lista de edifícios e os seus componentes internos. Alguns dos gráficos são similares aos existentes na página dos REC Dashboards como os gráficos da produção e consumo como se pode visualizar na Figura 50.

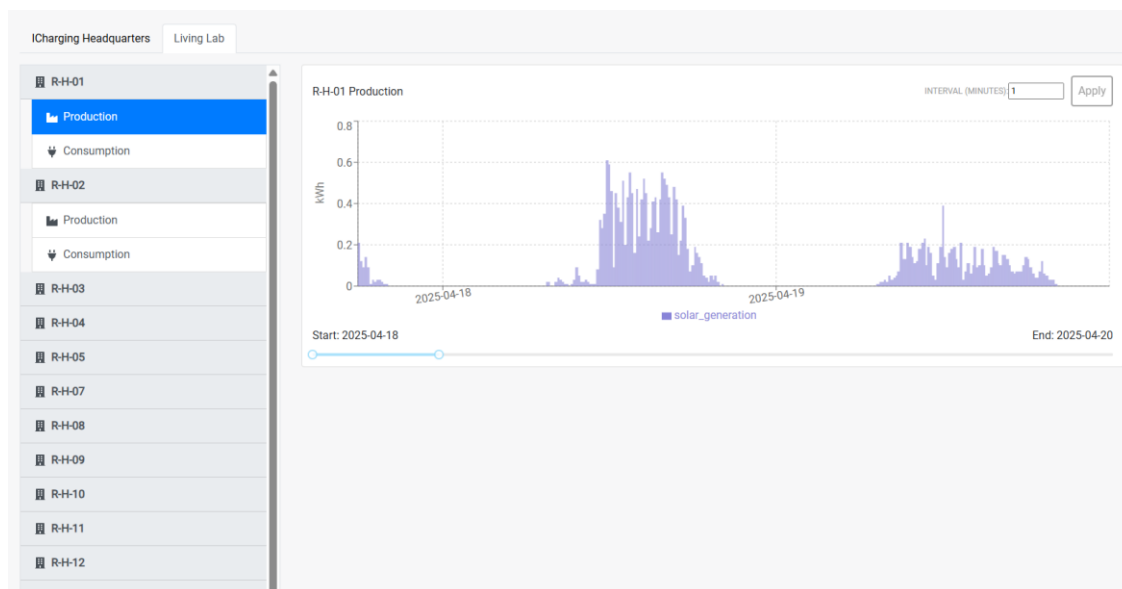


Figura 50 - Visualização dos dados dos cenários

### 5.2.3 Página “KPIs”

Apresentando uma estrutura e lógica similares à existente na página dos REC Dashboards, a página dos KPIs é responsável por apresentar os dados relativos aos KPIs de simulações e corresponde ao requisito funcional RF2.

Seguindo o mesmo processo de fazer o upload da pasta com os dados das simulações e seleccionar quais as simulações que o utilizador pretender analisar, os dados que aparecem terão um aspeto similar ao presente na Figura 51.

Simulation_1   Simulation_2   Simulation_3   Simulation_4																
KPIs for Simulation_1																
Parsed from exported_kpis.csv																
KPI	DISTRICT	BUILDING_1	BUILDING_2	BUILDING_3	BUILDING_4	BUILDING_5	BUILDING_6	BUILDING_7	BUILDING_8	BUILDING_9	BUILDING_10	BUILDING_11	BUILDING_12	BUILDING_13	BUILDING_14	BU
all_time_peak_average	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.0
annual_normalized_unserved_energy_total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
carbon_emissions_total	2.717	3.18	1.16	1.267	4.314	3.232	1.107	4.326	1.177	1.221	2.38	1.142	11.322	1.152	1.132	5.0
cost_total	2.911	3.371	1.222	1.32	4.031	3.438	1.189	4.586	1.229	1.303	2.579	1.196	12.531	1.205	1.194	6.0
daily_one_minus_load_factor_average	1.023															
daily_peak_average	0.002															
discomfort_cold_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
electricity_consumption_total	2.773	3.128	1.202	1.322	3.992	3.237	1.146	4.372	1.221	1.279	2.449	1.181	11.726	1.187	1.172	6.0
monthly_one_minus_load_factor_average	1.056															
ramping_average	0.0															

Figura 51 - Visualização dos KPIs de uma simulação

Aqui, são visíveis os dados relativos aos KPIs da simulação selecionada em formato de tabela sendo que as linhas representam cada um dos KPIs avaliados e as colunas representam cada um dos edifícios aos quais o KPI se refere.

Quando estão visíveis dados de mais de uma simulação, ou seja, existem múltiplos separadores visíveis na página, acima da tabela com os resultados existe um botão do lado direito do ecrã “Compare” o qual permite fazer o fluxo para comparar a simulação atualmente ativa com outra dentro das que o utilizador escolheu como visíveis.

Ao clicar no botão irá abrir uma pequena janela semelhante à de selecionar as simulações para visualizar dados, com a única diferença de que apenas pode ser selecionada uma das simulações da lista para efetuar a comparação entre as duas. Para este efeito determinou-se mais eficiente e intuitivo o uso de *radio buttons* ao contrário de múltiplas *checkboxes* forçando assim ao utilizador a escolha de apenas uma das simulações disponíveis.

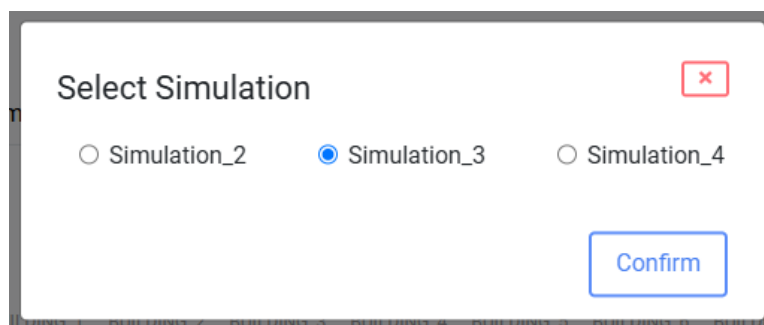


Figura 52 - Janela de seleção da simulação a comparar

Após a confirmação da simulação a comparar com a atual é criado um novo separador do lado direito da lista de separadores denominado “Comparison” o qual guarda os resultados da comparação feita.

KPI	DISTRICT	BUILDING_1	BUILDING_2	BUILDING_3	BUILDING_4	BUILDING_5	BUILDING_6	BUILDING_7	BUILDING_8	BUILDING_9	BUILDING_10	BUILDING_11	BUILDING_12	BUILDING_13	BUILDING_14	BU
all_time_peak_average	<b>0.200</b>															
annual_normalized_unserved_energy_total	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
carbon_emissions_total	0.000	<b>2.000</b>	0.000	<b>0.421</b>	0.000	0.000	0.000	0.000	0.000	0.000	<b>3.000</b>	0.000	<b>-3.000</b>	<b>1.000</b>	0.000	<b>-3</b>
cost_total	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
daily_one_minus_load_factor_average	0.000															
daily_peak_average	0.000															
discomfort_cold_delta_average	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_cold_delta_maximum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_cold_delta_minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	<b>7.000</b>	0.000
discomfort_hot_delta_average	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_hot_delta_maximum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_hot_delta_minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
electricity_consumption_total	0.000	<b>-0.570</b>	0.000	0.000	0.000	0.000	0.000	0.000	<b>-1.000</b>	<b>1.000</b>	0.000	0.000	0.000	0.000	0.000	<b>-3</b>
monthly_one_minus_load_factor_average	0.000															
ramping_average	0.000															

Figura 53 - Separador com os resultados da comparação

Os resultados presentes são o resultado de um simples processo de comparação: Ao carregar no botão de comparar estando na Simulação X e escolhendo a Simulação Y da lista de simulações disponíveis, o separador tem os resultados de Y – X, ou seja, todos os dados dos KPIs da Simulação X são subtraídos ao dados equivalentes da Simulação Y.

Dependendo do valor dos resultados finais são feitos alguns tratamentos para uma melhor visualização do resultados: os campos cujo resultado da subtração seja um número positivos ficam destacados a bold e a cor verde, se o resultado for um número negativo ficam igualmente destacados, mas a cor vermelha e caso o resultado final seja o valor 0, os valores permanecem a cor preta sem qualquer destaque aplicado.

#### 5.2.4 Página “Datasets”

Esta página é responsável pela criação de *datasets* consoante um conjunto de atributos definidos pelo utilizadore corresponde ao requisito funcional RF4.

A definição dos dados dos *datasets* está dividida em duas secções principais. Na primeira secção são preenchidas algumas informações acerca do *dataset* como o nome, o cenário de demonstração ao qual este pertence, chamado de *site* no contexto da interface, um conjunto de informações base e também a seleção das observações e ações.

Dataset Name: dataset\_teste

Site: PulseCharge

Base Info

Random Seed: 2022

Simulation Start Time Step: 0

Episode Time Steps: 1

Central Agent  Rolling Episode Split  Random Episode Split

Observations

Observation	Active	Shared
MONTH	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DAY TYPE	<input type="checkbox"/>	<input checked="" type="checkbox"/>
HOUR	<input type="checkbox"/>	<input type="checkbox"/>
DAYLIGHT SAVINGS STATUS	<input type="checkbox"/>	<input type="checkbox"/>
OUTDOOR DRY BULB TEMPERATURE	<input type="checkbox"/>	<input type="checkbox"/>

Actions

Agent Config (.json): Escolher ficheiro | Nenhum ficheiro selecionado

Simulation parameters: Root Directory: datasets, Simulation End Time Step: 8759, Seconds per Time Step: 3600

Figura 54 - Formulário de preenchimento dos dados do *dataset*

Na segunda secção, são definidos os dados relativos aos agentes e às funções de recompensa. Esta secção é mais simples pois apenas são necessárias seleções para o tipo de agente e função de recompensa e o upload de um ficheiro JSON com as configurações do agente.

Agent Info

Agent Type: citylearn.agents.rbc.BasicBatteryRBC

Agent Config (.json): Escolher ficheiro | Nenhum ficheiro selecionado

Reward Function Info

Reward Function Type: citylearn.reward\_function.RewardFunction

- citylearn.reward\_function.RewardFunction
- citylearn.reward\_function.MARL
- citylearn.reward\_function.IndependentSACReward
- citylearn.reward\_function.SolarPenaltyReward
- citylearn.reward\_function.ComfortReward
- citylearn.reward\_function.SolarPenaltyAndComfortReward
- citylearn.reward\_function.V2GPenaltyReward

Figura 55 - Secção dos dados do agente e função de recompensa

Ao seleccionar o ficheiro de configurações aparece por baixo uma pré-visualização do conteúdo do ficheiro a qual é editável caso sejam pretendidas fazer alterações de última hora ao ficheiro de configurações.

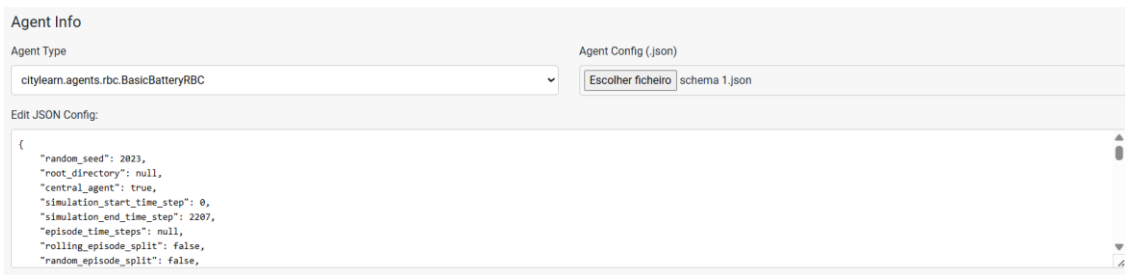


Figura 56 - Caixa de texto de pré-visualização do JSON

Finalmente, tem uma pequena secção para definir o valor do período entre cada um dos passos temporais do *dataset*, juntamente com o intervalo da data e hora inicial e final dos dados que farão parte do *dataset*.

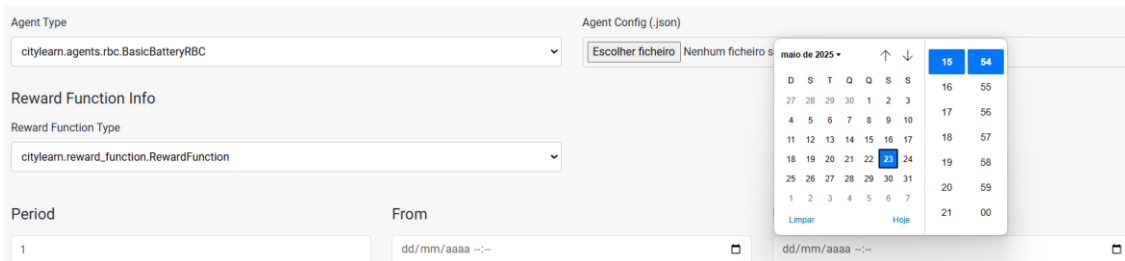


Figura 57 - Campos do período e data e hora inicial e final

Após os dados estarem preenchidos, quando o utilizador clica no botão “Save Dataset”, é chamado um *endpoint* que irá fazer a construção do *dataset* com base nos dados preenchidos. Esses dados estão organizados numa estrutura dividida em diversas propriedades. Grande parte destas são relativas aos dados preenchidos na primeira secção através das caixas de texto como por exemplo “*random\_seed*”, “*root\_directory*”, etc. No entanto, existem outros casos que não possuem uma representação tão direta dos resultados como:

- **observations** – Representa todas as observações existentes. Consoante os valores selecionados nas *checkboxes* correspondentes, aparecem os valores como “*true*” ou “*false*”.
- **actions** – Representa todas as ações existentes. Consoante os valores selecionados nas *checkboxes* correspondentes, aparecem os valores como “*true*” ou “*false*”.
- **agent** – Representa os dados relativamente ao agente, nomeadamente o tipo e dados da configuração.
- **reward\_function** – Representa os dados relativamente à função de recompensa.

Após o processo ser concluído, o utilizador recebe uma notificação no canto superior direito do ecrã a indicar que a operação foi concluída com sucesso.

## 5.2.5 Página “Schema”

A página Schema é responsável pela criação dinâmica do esquema do *schema* consoante um conjunto de atributos os quais devem ser definidos pelo utilizador e associando este a um site, também definido pelo utilizador. Está página é utilizada para facilitar a criação de *schemas* para os utilizadores e corresponde ao requisito funcional RF6.

Em termos de design a página é constituída por uma caixa de texto na qual o utilizador define o nome do site ao qual vai associar o *schema* e temos um *canvas* no qual a montagem da estrutura de elementos do *schema* ocorre. Para o desenvolvimento desta funcionalidade foi utilizado como ferramenta auxiliar a biblioteca ReactFlow, que permitiu criar um componente que permite a criação dinâmica de elementos no *schema* seguindo uma estrutura que faz lembrar um editor de diagramas, similar a algo que poderemos encontrar em aplicações como o Visual Paradigm.

Do lado esquerdo temos a lista de componentes existentes no sistema que podem ser arrastados para o *canvas* de forma a criar um componente. Os edifícios são os componentes principais que representam os edifícios nas simulações. Podem ser adicionados a estes diversos equipamentos como carregadores, equipamentos de aquecimento/arrefecimento, entre outros. Tanto os edifícios como os equipamentos têm um conjunto de campos que devem ser preenchidos com as especificidades necessárias para a simulação. Para além disso, é possível fazer a associação de equipamentos aos edifícios ligando os dois componentes juntos através de uma linha e outras ações como alterar o nome dos componentes e fazer a ação de copiar-colar outros componentes para facilitar o preenchimento de dados para múltiplos componentes similares.

Na Figura 58, é possível ver o *canvas* preenchido com alguns componentes.

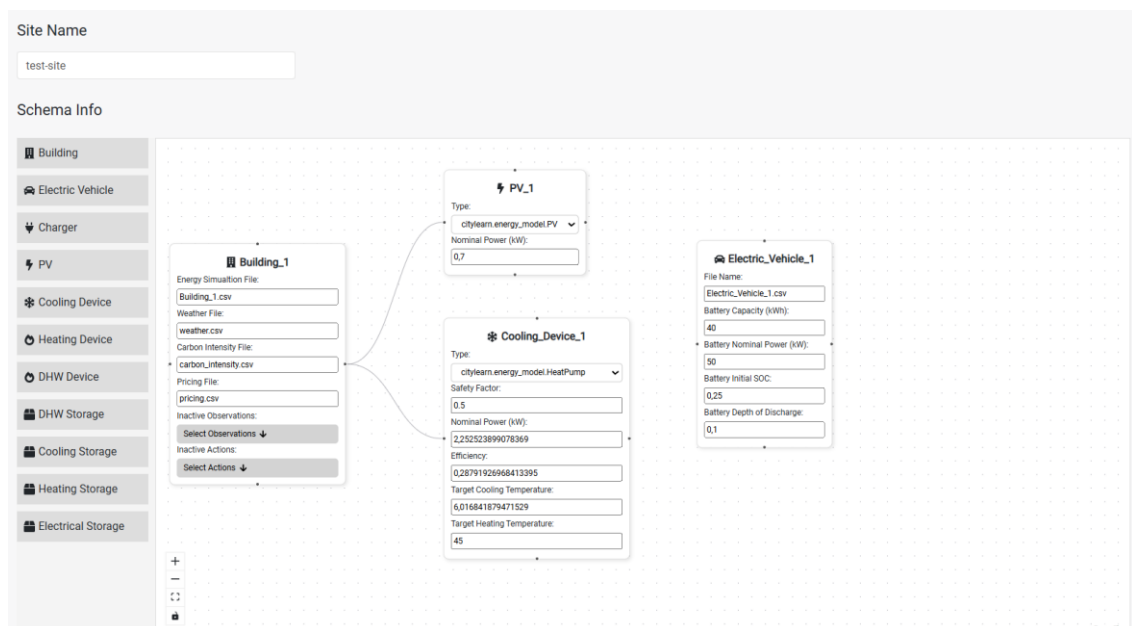


Figura 58 - Canvas com componentes e as suas ligações

Quando se clica no botão “Save Schema”, é chamado um *endpoint* que irá fazer a criação do site com o nome fornecido e associar um *schema* com os dados do esquema desenhado no *canvas*. Esses dados estão organizados numa estrutura dividida em duas propriedades principais: "electric\_vehicles\_def" e "buildings". A primeira é relativa aos dados dos veículos elétricos definidos no esquema enquanto a segunda é relativa aos edifícios e aos equipamentos associados a estes.

### 5.2.6 Página “Experiment Configs”

Nesta página, o objetivo é a criação de um ficheiro de configurações para guardar as configurações do *dataset*, do *schema* e do algoritmo, cumprindo com o requisito funcional RF3 e assim com base neste, criar um teste.

Os campos presentes nesta página são o nome do ficheiro ao qual as configurações vão ficar associadas e os parâmetros das configurações, que devem estar num formato JSON válido.

Figura 59 - Formulário do ficheiro de configuração

Ambos os campos devem ser preenchidos senão o botão “Save Config” permanecerá inativo. Após os campos estarem preenchidos, ao clicar no botão é chamado um *endpoint* para guardar os dados relativos ao ficheiro de configuração. Após o pedido ser concluído aparece uma mensagem para o utilizador a indicar que a operação foi concluída com sucesso.

### 5.2.7 Página “Run Simulations”

Nesta página, o objetivo principal é colocar simulações a correr em contentores externos no DockerHub com base em configurações definidas para as mesmas, sejam estas a partir de ficheiros de configuração, *schemas* tais como os definidos na sua página própria ou com base em *datasets*. Esta página permite a concretização do requisito funcional RF5.

Os campos principais a serem preenchidos nesta página são o nome do ficheiro no qual estão guardadas as configurações para correr a simulação e o contentor no qual a simulação irá correr. Ambos os campos têm uma lista de valores com possíveis opções que estão guardadas

externamente. Para obter estes valores é chamado um *endpoint* para cada campo para o preencher com a lista de valores.

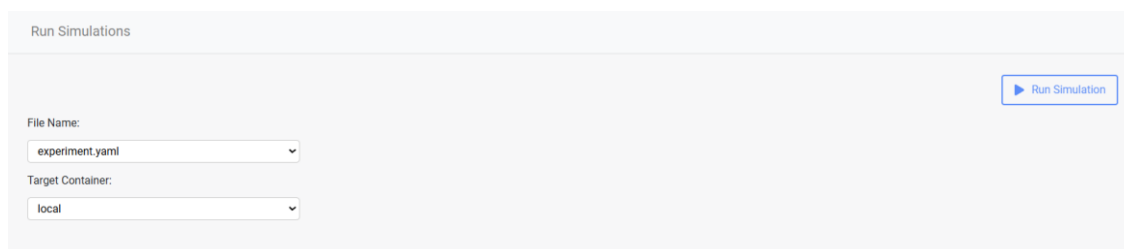


Figura 60 - Página "Run Simulations"

Após o utilizador seleccionar ambos os valores que pretende utilizar, quando o utilizador clica no botão "Run Simulation", é chamado um outro *endpoint* cujo objetivo é criar a instância da simulação e colocar a mesma a correr. Após o processo estar concluído com sucesso, aparece no canto superior direito uma mensagem de sucesso.

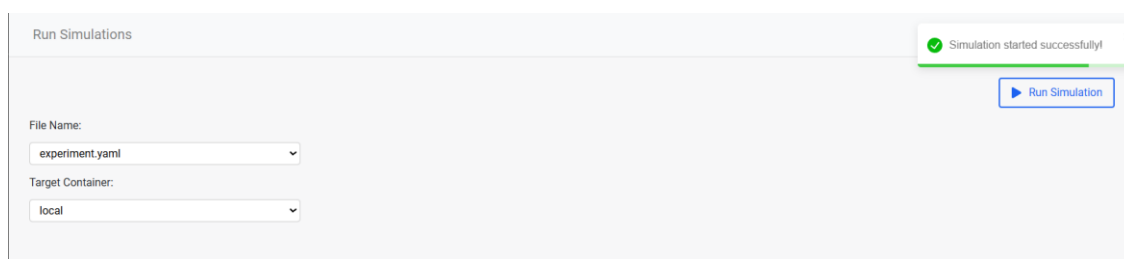


Figura 61 - Mensagem de sucesso

### 5.2.8 Card Components

Um estilo de componentes bastante utilizado ao longo da interface são os chamados de "Cards", cujo objetivo é apresentar um gráfico e permitir alterar os parâmetros do mesmo de forma a dar liberdade ao utilizador de filtrar os dados que quer ver e da forma que lhe for mais conveniente. Estes são utilizados em todas as páginas que envolvem a visualização de gráficos.

Todos os gráficos são constituídos por um conjunto de componentes comuns:

- **XAxis** – Representa o eixo do X no gráfico. A este está associado o *timestamp* dos resultados cujos *ticks* são visíveis sempre que este chega à meia-noite ou à hora mais próxima desta. Representam sempre uma data no formato YYYY-MM-DD, marcando assim um novo dia.
- **YAxis** – Representa o eixo do Y no gráfico. Está associado aos valores possíveis dos campos do gráficos. Podem ser adicionados múltiplos para representar diferentes escalas de valores (por exemplo valores de consumos e percentagens).
- **Tooltip** – Representa a caixa de texto que aparece quando se passa o rato por cima dos dados do gráfico.

- **Legend** – Representa a legenda do gráfico. Permite facilmente ao utilizador saber a correspondência entre os campos e a sua representação no gráfico.

O código presente no Excerto de Código 1, mostra o esqueleto geral de um gráfico da interface, utilizando como exemplo um gráfico de produção.

#### Excerto de Código 1 - Esqueleto geral de um gráfico

```
<ResponsiveContentor width="100%" height={300}>
  <ComposedChart data={aggregatedData} stackOffset="sign">
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis
      dataKey="Time Step"
      ticks={xAxisTicks}
      tickFormatter={(tick) => tick.slice(0, 10)}
      angle={-8}
    />
    <YAxis label={{ value: "kWh", angle: -90, position:
"insideLeft" }} />
    <Tooltip
      labelFormatter={(label) => {
        const date = new Date(label);
        return date.toLocaleString("en-US", {
          month: "short",
          day: "2-digit",
          year: "numeric",
          hour: "2-digit",
          minute: "2-digit",
          hour12: false,
        });
      }}
      formatter={(value, name) => [`${value.toFixed(3)} kWh`,
name]}
    />
    <Legend />
    <Bar dataKey="Energy Production from PV-kWh" stackId="a"
fill="#8884d8" />
    <Bar dataKey="Energy Production From EV-kWh" stackId="a"
fill="#82ca9d" />
  </ComposedChart>
</ResponsiveContentor>
```

#### 5.2.8.1 Métodos de filtragem dos dados

Um ponto importante comum a todos os gráficos presentes na interface é que todos eles têm alguma lógica relativamente à forma como se pode filtrar e visualizar os dados que estão a ser apresentados no momento. Esta apresenta-se sobre três aspetos diferentes: a alteração do intervalo de tempo entre cada “step”, a filtragem dos dados a serem visualizados através de *checkboxes* e o filtro das datas inicial e final do período no qual se consegue ver dados, representado por um *slider*.

#### 5.2.8.2 Alteração do intervalo de tempo entre “steps”

Este filtro é utilizado especialmente quando existe um maior volume de dados e o utilizador pretende alterar o intervalo de tempo entre cada “step”. Um ponto importante a ter em conta é o facto de que só podem ser escolhidos valores que não sejam superiores a 60 ou inferiores

ao do intervalo base do ficheiro, ou seja, se os dados de input inicialmente tiverem um intervalo entre cada “step” de 10 minutos por exemplo, o utilizador não pode escolher ver os dados em intervalos de 5 minutos, pois não corresponderiam a valores corretos.

Para além destas restrições, existem uma restrição adicional na qual apenas são considerados válidos valores múltiplos do valor base. Por exemplo: com um intervalo base de 15 min, as opções possíveis de intervalos são 15, 30, 45 e 60 minutos. Nas Figuras 62 e 63, temos um exemplo onde tendo um intervalo base 1 minuto, são vistos os dados em intervalos de 10 e 40 minutos respetivamente.

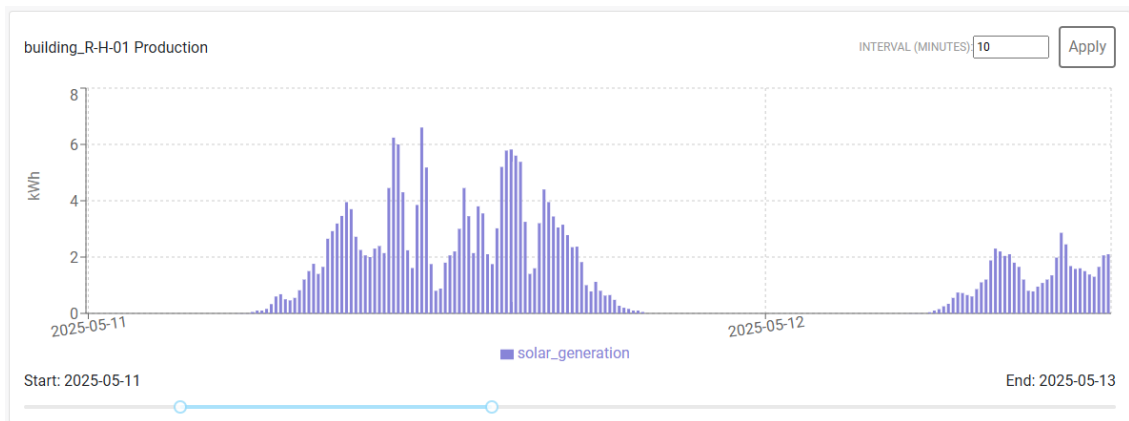


Figura 62 - Representação dos dados com intervalo de 10 minutos

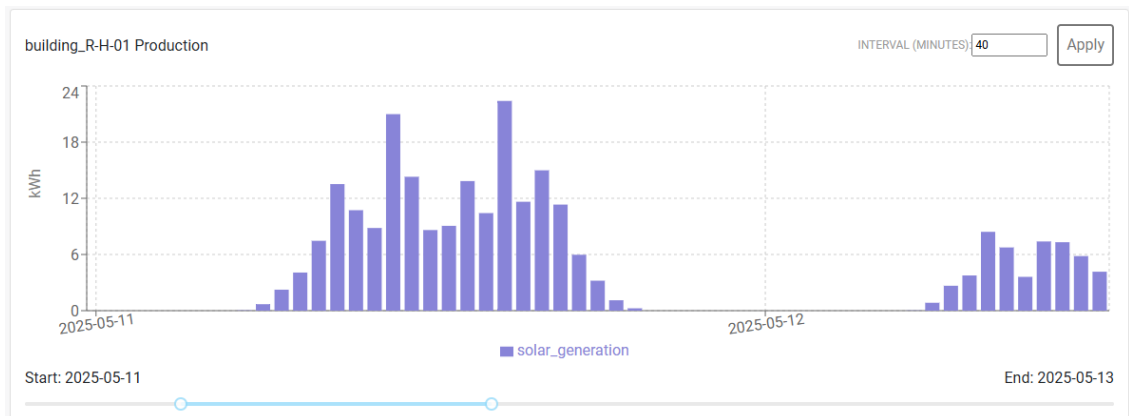


Figura 63 - Representação dos dados com intervalo de 40 minutos

A forma como os dados são calculados é utilizando o intervalo base e a partir daí utilizá-lo como base para calcular os outros. O método para calcular o intervalo base é o presente no Excerto de Código 2.

#### Excerto de Código 2 - Método para determinar intervalo base

```
const baseIntervalMinutes = updatedData.length > 1
  ? Math.round((updatedData[1].timestamp - updatedData[0].timestamp)
    / (60 * 1000))
  : 1;
```

Tendo o intervalo base, é possível calcular outros desde que sejam múltiplos do intervalo base de forma a evitar cálculos errados.

No código abaixo está feita a estrutura do filtro do intervalo, o qual é constituído por uma caixa de texto e um botão para aplicar as alterações. O botão ficará inativo caso o valor do intervalo não seja um múltiplo, ou seja, inferior ao intervalo base ou superior a 60. Para o utilizador saber qual é o intervalo base para aqueles dados existe uma *tooltip* que indica o valor do mesmo.

#### Excerto de Código 3 - Estrutura do filtro de intervalos

```
<div style={{ marginBottom: "1rem" }}>
  <label>
    <span title={`Base interval from data: ${baseIntervalMinutes}
minute (s) `}>
      Interval (minutes):
    </span>
    <input type="number" min={baseIntervalMinutes} max={60}
value={intervalInput} onChange={(e) => {
  const val = parseInt(e.target.value);
  setIntervalInput(Number.isNaN(val) ? baseIntervalMinutes :
val);
}}
style={{ width: "80px", marginRight: "10px", marginLeft: "5px"
}}/>
  </label>
  <Button className="p-2" variant="secondary"
onClick={handleApplyInterval} disabled={ intervalInput <
baseIntervalMinutes || intervalInput > 60
}>
    Apply
  </Button>
</div>
```

Após a aplicação do intervalo, todos os dados serão recalculados consoante o intervalo definido. Um ponto importante a considerar no cálculo do novo valor das variáveis é o tipo de valor a ser calculado, isto é, se o valor for um valor simples como um consumo ou um valor de energia estes podem simplesmente ser somados. No entanto, se for uma percentagem por exemplo, a soma já não faz sentido ser utilizada e deve ser feita a média dos valores.

O Excerto de Código 4 mostra um caso em que ambas lógicas existem simultaneamente: soma para os valores de x e média para os valores de y.

#### Excerto de Código 4 - Lógica de soma/média de valores por intervalo

```
const aggregateGroup = (timestamp, group) => {
  const sumKeys = ['Charger Consumption-kWh', 'Charger Production-
kWh'];
  const avgKeys = ['EV Estimated SOC Arrival-%', 'EV Required SOC
Departure-%', 'EV SOC-%'];

  const aggregated = { timestamp, 'Time Step': group[0]['Time Step']
};

  sumKeys.forEach(key => {
```

```

        const values = group.map(item => item[key]).filter(val => val
        !== null);
        aggregated[key] = values.length > 0
            ? values.reduce((sum, v) => sum + v, 0)
            : null;
    });

    avgKeys.forEach(key => {
        const values = group.map(item => item[key]).filter(val => val
        !== null);
        aggregated[key] = values.length > 0
            ? values.reduce((sum, v) => sum + v, 0) / values.length
            : null;
    });

    return aggregated;
};

```

Este método é um método auxiliar utilizando em conjunto com o método principal de agregação cujo objetivo é organizar os dados em grupos com base no intervalo atualmente definido. Esses grupos são posteriormente colocados no método auxiliar para realizar a soma/média dos valores.

### 5.2.8.3 Visualização dos campos

A visualização dos campos do gráfico é alterada através das *checkboxes* situadas acima do gráfico. É um simples caso de selecionar/desselecionar um campo para que este apareça/desapareça do ecrã.

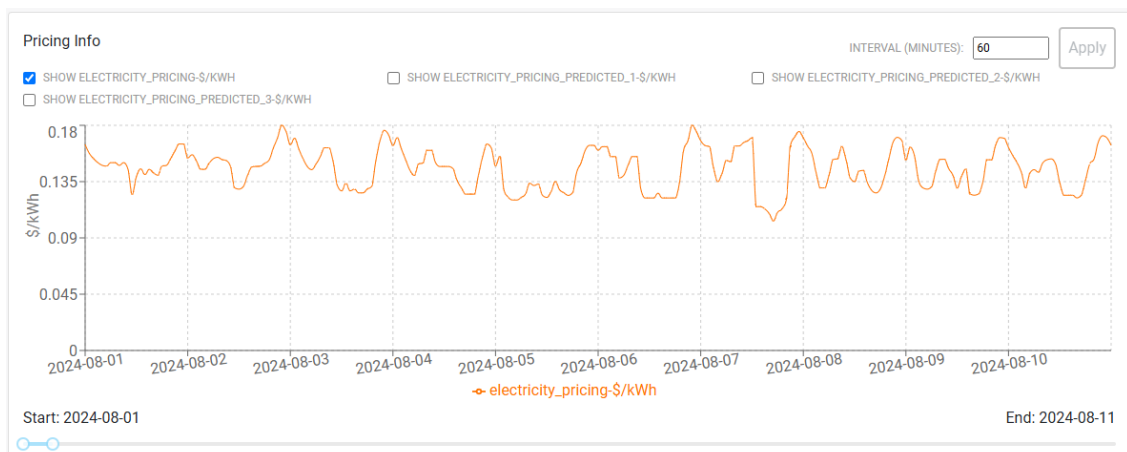


Figura 64 - Exemplo do filtro de visualização de campos

O processo para garantir este comportamento começa pela declaração das variáveis presentes no gráfico e o método para ativar/desativar a visibilidade das mesmas como demonstra o Excerto de Código 5.

#### Excerto de Código 5 - Inicialização dos campos visíveis

```

const [visibleSeries, setVisibleSeries] = useState({

```

```

    'electricity_pricing-$/kWh': true,
    'electricity_pricing_predicted_1-$/kWh': true,
    'electricity_pricing_predicted_2-$/kWh': true,
    'electricity_pricing_predicted_3-$/kWh': true
  });

const handleCheckboxChange = (seriesKey) => {
  setVisibleSeries((prev) => ({
    ...prev,
    [seriesKey]: !prev[seriesKey]
  }));
};

```

Com base nesses campos o próximo passo é construir a estrutura com as *checkboxes* para permitir ao utilizador de ativar ou desativar o campo.

#### Excerto de Código 6 - *Checkboxes* com a visibilidade dos campos

```

<div className='d-flex align-items-center mb-2'>
  <Row style={{ width: "100%" }}>
    {Object.keys(visibleSeries).map((key) => (
      <Col key={key} md={4}>
        <label className='d-flex align-items-center'>
          <input
            type="checkbox"
            checked={visibleSeries[key]}
            onChange={() => handleCheckboxChange(key)}
          />
          <span className='ml-2'>Show {key}</span>
        </label>
      </Col>
    ))}
  </Row>
</div>

```

Finalmente, na declaração dos componentes do gráfico basta adicionar a condição para quando a visibilidade do campo em questão estiver com o valor “true”, este seja apresentado no gráfico.

#### Excerto de Código 7 - Controlo da visibilidade dos elementos do gráfico

```

{visibleSeries['electricity_pricing-$/kWh'] && (
  <Line type="monotone" dataKey="electricity_pricing-$/kWh"
  stroke="#FF7300" dot={false} />
)}
{visibleSeries['electricity_pricing_predicted_1-$/kWh'] && (
  <Line type="monotone" dataKey="electricity_pricing_predicted_1-
  $/kWh" stroke="#8884d8" dot={false} />
)}
{visibleSeries['electricity_pricing_predicted_2-$/kWh'] && (
  <Line type="monotone" dataKey="electricity_pricing_predicted_2-
  $/kWh" stroke="#82ca9d" dot={false} />
)}
{visibleSeries['electricity_pricing_predicted_3-$/kWh'] && (
  <Line type="monotone" dataKey="electricity_pricing_predicted_3-
  $/kWh" stroke="#0088FE" dot={false} />
)}

```

#### 5.2.8.4 Alteração da data inicial e final

Se for intenção do utilizador modificar o período de tempo no qual existem dados basta utilizar o *slider* por baixo do gráfico. O utilizador pode arrastar as duas pontas para alterar a data inicial e final e o gráfico filtra os resultados automaticamente consoante os valores. Para ter sempre em conta o período no qual o utilizador se encontra, no lado esquerdo e direito estão presentes os valores atuais da data inicial e final respetivamente.

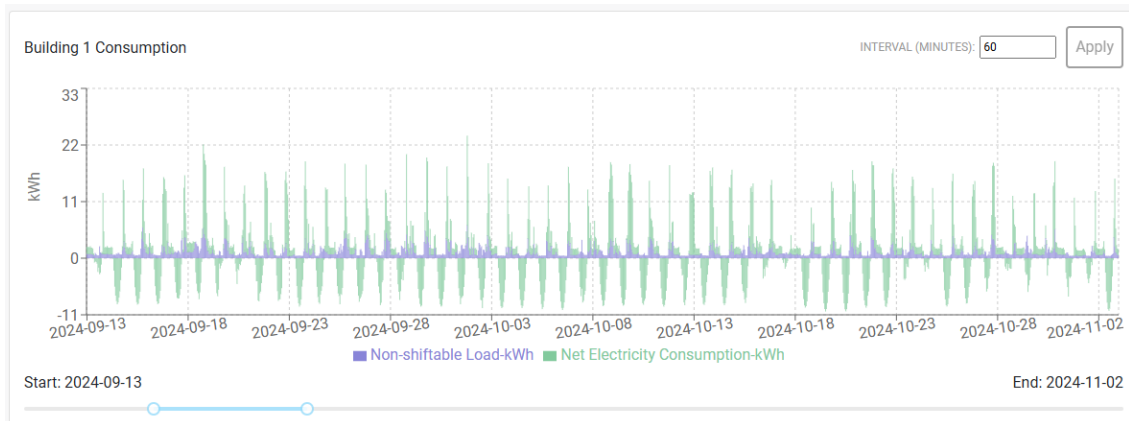


Figura 65 - Exemplo do filtro da data inicial e final

Para isto foi criado um componente "DateRangeSlider" que representa o *slider* em questão e regista os valores da data inicial e final atualmente selecionadas e envia esses dados para o componente do gráfico. Este componente fica associado ao gráfico no qual esteja inserido e permite assim interagir com este para alterar o intervalo de datas atualmente visível.

#### Excerto de Código 8 - Componente DateRangeSlider

```
const DateRangeSlider = ({ minTimestamp, maxTimestamp, sliderValues,
onSliderChange }) => {
  const formatDate = (timestamp) => dayjs(timestamp).format("YYYY-MM-DD");

  const step = 24 * 60 * 60 * 1000; // Step = 1 day

  return (
    <div className="p-6 w-600px mx-auto">
      <div className="d-flex justify-content-between mb-2">
        <span>Start: {formatDate(sliderValues[0])}</span>
        <span>End: {formatDate(sliderValues[1])}</span>
      </div>

      <Slider
        range
        min={minTimestamp}
        max={maxTimestamp}
        value={sliderValues}
        onChange={onSliderChange}
        step={step}
      />
    </div>
  );
};
```

```
};
```

Após ocorrer uma alteração no *slider*, o método `handleSliderChange` é chamado para alterar o valor da variável que contém os valores atuais dos *sliders* que serão utilizados para filtrar os dados atuais do gráfico como demonstra o Excerto de Código 9.

Excerto de Código 9 - Método para filtrar dados com base no valor do slider

```
const handleSliderChange = (values) => setSliderValues(values);

const filteredData = updatedData.filter(
  (item) => item.timestamp >= sliderValues[0] && item.timestamp <=
  sliderValues[1]
);
```

#### 5.2.8.5 Card Production e Consumption

Os gráficos relativos à produção e consumo são os mais simples de entre os outros gráficos existentes. São constituídos por barras sobrepostas representando dois campos ao mesmo tempo por cada *timestamp*. De forma a não ocorrerem erros de cálculo tendo um campo com valor positivo e outro com valor negativo é adicionada ao gráfico a propriedade `stackOffset="sign"` garantindo assim que mesmo que isso aconteça, os valores ficam separados a partir da linha com valor 0.

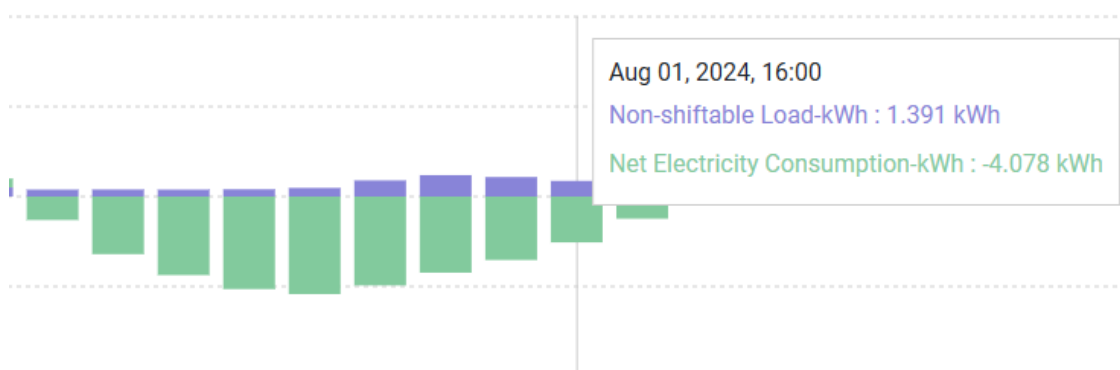


Figura 66 – Separação de valores positivos e negativos

Um exemplo do gráfico de produção está representado na Figura 67.

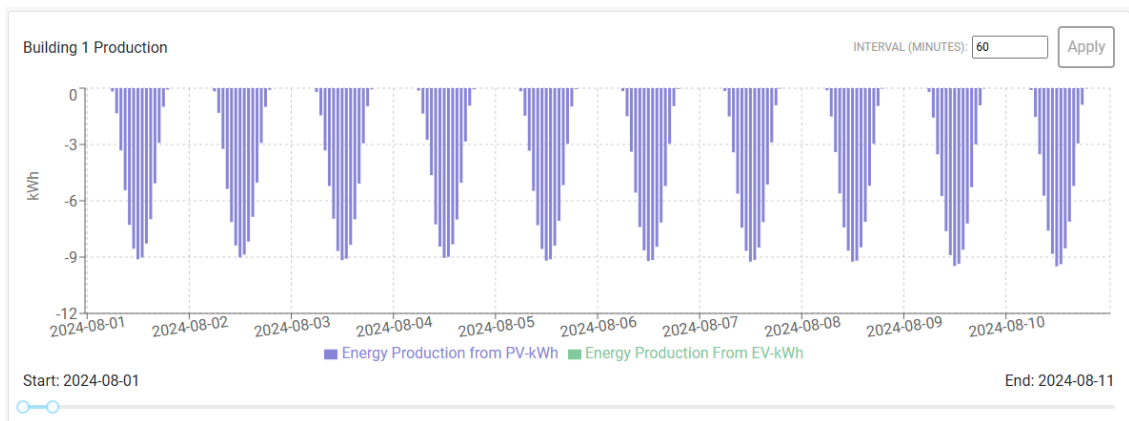


Figura 67 - Gráfico de produção

Os gráficos de consumo seguem exatamente a mesma estrutura com a diferença de que os dados apresentados se referem ao *Non-Shiftable Load* e à *Net Electricity Consumption*.

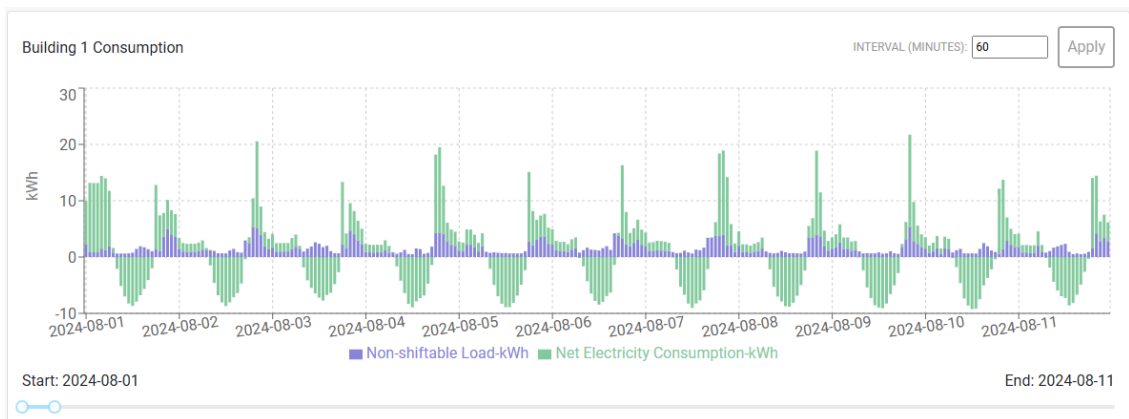


Figura 68 - Gráfico de consumo

#### 5.2.8.6 Card Charger

Os gráficos relativos aos dados dos carregadores exigiram a adição de vários tipos de indicadores para os seus campos, utilizando barras para valores de consumo e produção, linhas para os valores do SOC e linhas de referência para os momentos em que um veículo parte ou retorna para a estação de carregamento.

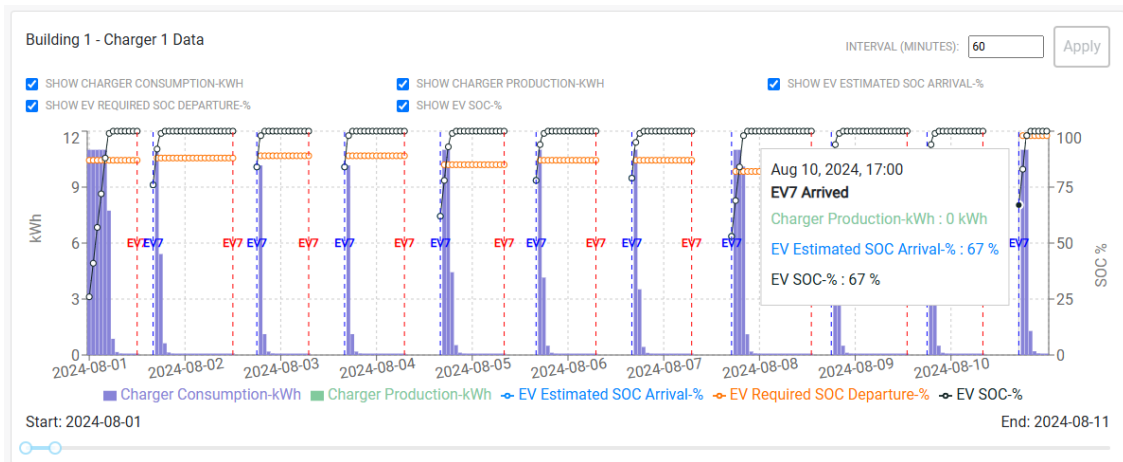


Figura 69 - Gráfico dos carregadores

Devido à possibilidade de as linhas de referência não serem muito perceptíveis em cenários que as mesmas fossem sobrepostas aos valores das barras, foi adicionada à *tooltip* uma lógica adicional para criar uma mensagem. São recolhidos todos os marcadores de chegada e partida e conforme o seu tipo são adicionados à mensagem da *tooltip* com mostra o Excerto de Código 10.

Excerto de Código 10 - Mensagem de partida/chegada de um veículo elétrico

```
// Find matching arrival and departure EVs
const arrivedEVs = arrivalMarkers
  .filter(marker => marker.timestep === label)
  .map(marker => formatEVName(marker.EVName));

const departedEVs = departureMarkers
  .filter(marker => marker.timestep === label)
  .map(marker => formatEVName(marker.EVName));

// Create message if an EV arrives or departs
let evMessage = "";
if (arrivedEVs.length > 0) {
  evMessage += `${arrivedEVs.join(", ")} Arrived`;
}
if (departedEVs.length > 0) {
  evMessage += `${departedEVs.join(", ")} Departed`;
}

return (
  <>
  <span>{formattedDate}</span>
  {evMessage && <span style={{ fontWeight: "bold", marginTop:
"4px" }}><br />{evMessage}</span>}
  </>
);
```

### 5.2.8.7 Card Battery

Nos gráficos relacionados com as baterias os dados a apresentar são relativamente simples, sendo o valor da descarga da bateria em kWh e utilizando barras e a percentagem do SOC utilizando uma linha.

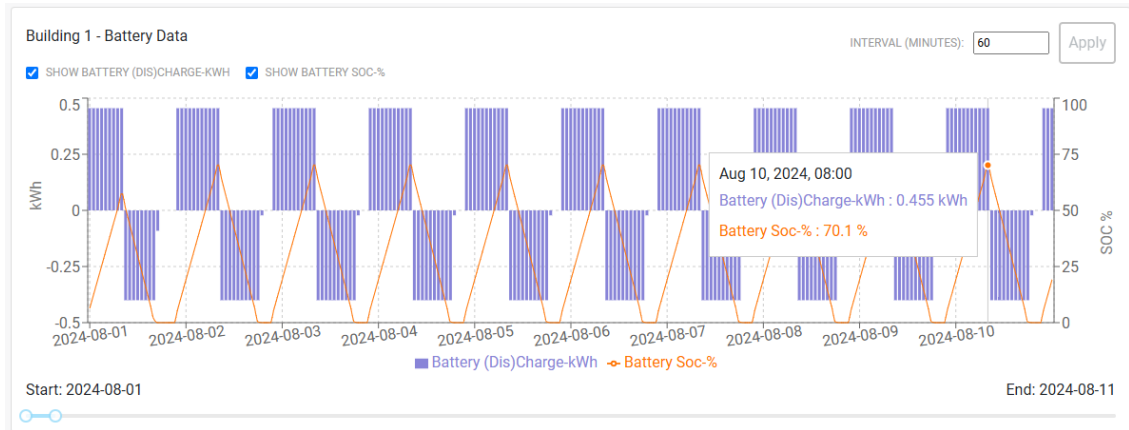


Figura 70 - Gráfico das baterias

Como ambas variáveis têm intervalos máximos diferentes, na definição do gráfico são declarados dos dois eixos Y tal como demonstra o Excerto de Código 11.

#### Excerto de Código 11 - Múltiplos eixos Y

```
<YAxis label={{ value: 'kWh', angle: -90, position: 'insideLeft' }} />  
<YAxis  
  yAxisId="right"  
  orientation="right"  
  domain={{ [0, 100] }}  
  label={{ value: 'SOC %', angle: -90, position: 'insideRight' }}  
>
```

### 5.2.8.8 Card EV

Para os gráficos dos veículos elétricos são utilizadas três linhas para a representação dos campos. Todos estes são representados com percentagens, por isso neste caso não é necessário definir mais do que um eixo Y.

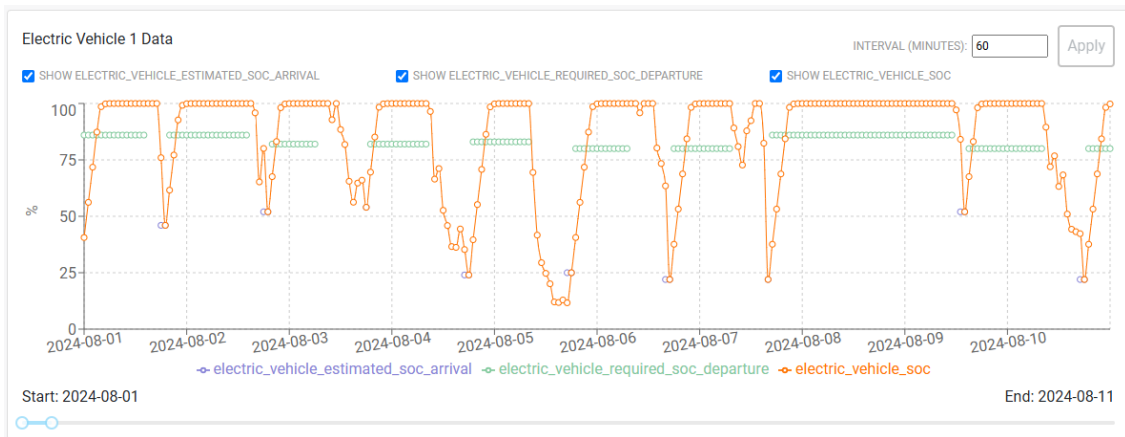


Figura 71 - Gráfico dos veículos elétricos

### 5.2.8.9 Card Pricing

Os gráficos de preços têm formatos similares aos dos veículos elétricos, com a diferença de serem apresentadas quatro linhas, com a primeira sendo referente a valor dos preços do momento e as três outras linhas serem os valores previstos nos próximos 1, 2 e 3 meses respectivamente.

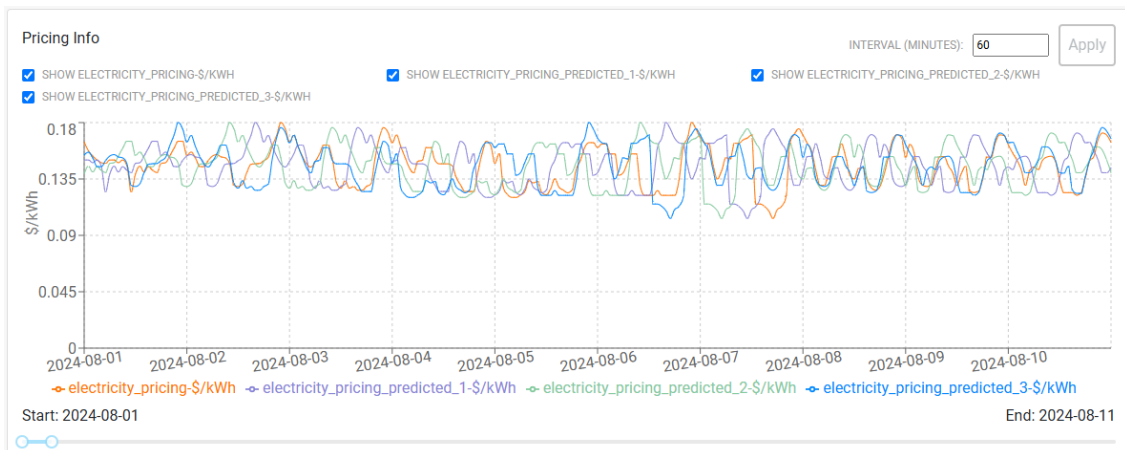


Figura 72 - Gráfico dos preços

## 5.3 Implantação da interface

Para fazer a integração com os outros componentes do projeto e colocar a interface a funcionar num ambiente real, foi feito adicionalmente um processo de implantação de forma a ter a interface disponível numa das máquinas virtuais do ISEP. Para este processo, foram criados ficheiros dockerfile e docker.compose para conseguir que a interface ficasse integrada num Docker.

De seguida foi necessário estabelecer uma conexão com uma máquina virtual do ISEP. Foi utilizado o PuTTY para esse efeito. Na Figura 73 conseguimos ver a interface desenvolvida inserida na máquina virtual juntamente com outros componentes do projeto.

```
opeva_softcps@softcps: /opt/opeva_infra_services
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

73 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Wed May 21 10:32:35 2025 from 193.136.62.5
opeva_softcps@softcps:~$ cd /opt/opeva_infra_services/
opeva_softcps@softcps:/opt/opeva_infra_services$ ls
citylearn_ui  opeva_backend_api  rabbitmq_broker
mlflow       percepta            README.md
mongodb      pulsecharge        simulation_algorithm_training
opeva_softcps@softcps:/opt/opeva_infra_services$
```

Figura 73 - Interface do puTTY

Após conectar com sucesso na máquina foi executado um comando para correr o ficheiro `docker.compose` para criar uma imagem e respetivo contentor dentro da máquina.

O fluxo do processo de implantação torna-se automático pois sempre que é feito um `commit` para o repositório do GitHub a cada dia é atualizado automaticamente o contentor da máquina virtual através de uma `watchtower` definida que verifica a cada 24 horas se existe alguma atualização nos dados e caso isso se verifique efetua as ações para atualizar. O diagrama da Figura 74 representa este fluxo na sua totalidade.

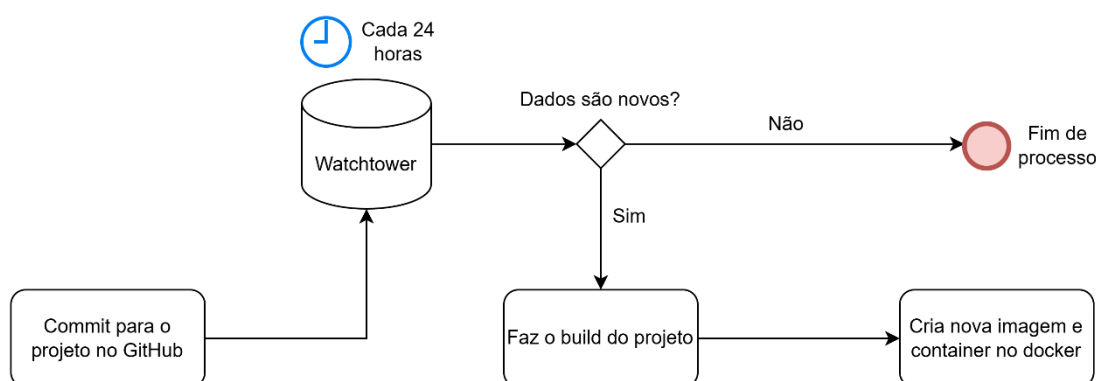


Figura 74 - Fluxo do processo de atualização da imagem no Docker

Tendo sido o processo de implantação concluído com sucesso, em vez de estar disponível apenas localmente, agora também é possível acessar à interface através do endereço IP da máquina virtual na qual esteja inserida.

## 5.4 Avaliação de Métricas

Seguir métricas no design de interfaces é crucial porque ajuda a construir experiências eficazes, centradas no utilizador e mensuráveis. Sem métricas, as decisões de design baseiam-se apenas em suposições ou na estética e não em se realmente resolvem os problemas do utilizador ou atendem aos objetivos do negócio.

Como medidor de métricas para avaliar a qualidade da interface desenvolvida foi utilizado o WCAG pois a acessibilidade é um dos fatores mais importantes de uma interface. Seguindo as métricas do WCAG será possível para a interface oferecer uma experiência melhor e mais acessível para todos os utilizadores, não só aqueles que sofrem de algum tipo de deficiência.

### 5.4.1 WCAG

Diferentes conjuntos de diretrizes foram sugeridos para assegurar a acessibilidade de sites. O conjunto de diretrizes mais conhecido são as Diretrizes de Acessibilidade para Conteúdo Web (WCAG) do World Wide Web Consortium (W3C). Essas diretrizes são vistas como orientações globais padrão que podem ser utilizadas por desenvolvedores de sites durante o design e a criação de páginas web, esperando-se que utilizadores com deficiência recebam os mesmos benefícios dos serviços oferecidos pelo site. (Akram et al., 2023)

A primeira versão do padrão WCAG (WCAG 1.0) incluía 14 diretrizes, 64 pontos de verificação e três níveis de conformidade:

- O **Nível A** é um requisito básico para que alguns utilizadores com deficiência possam acessar e utilizar conteúdo da web
- O **Nível AA** indica acessibilidade geral e a remoção de barreiras significativas ao acesso ao conteúdo
- O **Nível AAA** oferece melhorias e aprimoramentos na acessibilidade da web para alguns utilizadores com deficiência.

Para que um site declare estar em conformidade com um desses níveis, todas as suas páginas necessitam de cumprir com todos os critérios de sucesso para esse nível e os precedentes. (Chatziemmanouil & Katsanos, 2024)

### 5.4.2 WAVE

Para avaliar a interface relativamente ao WCAG foi utilizada uma ferramenta chamada WAVE. O WAVE é um recurso automatizado criado pela WebAIM que possibilita aos utilizadores

introduzir o URL de um site em funcionamento. O seu objetivo é auxiliar programadores web a avaliar a acessibilidade de uma página web específica para torná-la mais acessível. Ele insere diversos ícones numa página na web que possibilitam que utilizadores e especialistas identifiquem potenciais problemas de acessibilidade. Ícones vermelhos sinalizam problemas de acessibilidade, ícones amarelos sinalizam alertas, ícones verdes sinalizam recursos de acessibilidade e todos os ícones azul-claros sinalizam elementos estruturais, semânticos ou de navegação. (Alsaeedi, 2020)

Para aplicar o WAVE na interface foi utilizada uma extensão do Google Chrome que permite facilmente analisar uma página simplesmente fazendo click direito nesta e clicando na opção “WAVE this page”.

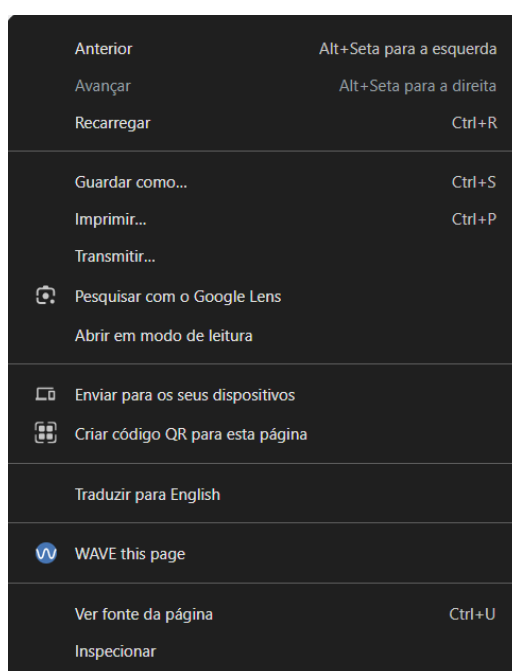


Figura 75 - Extensão do Chrome para uso do WAVE

Após fazer a análise da página, é apresentada uma estrutura como mostra a Figura 76 onde se pode ver todos os detalhes da análise.

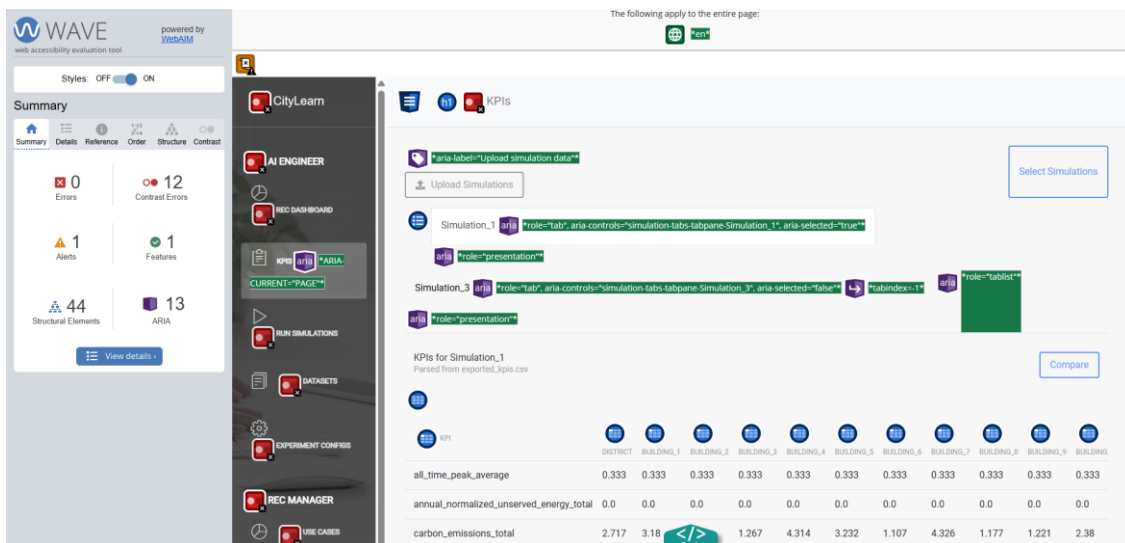


Figura 76 - Resultados da análise da página

Podemos assim analisar os resultados obtidos através da ferramenta. Para o exemplo da página de KPIs, podemos retirar as seguintes conclusões:

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 12 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 1 alerta, inofensivo para os níveis do WCAG
- **Níveis WCAG cumpridos:** A

Para as outras páginas da interface os resultados analisados podem se resumir ao seguinte:

#### 5.4.2.1 Página REC Dashboards

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 27 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 1 alerta, inofensivo para os níveis do WCAG
- **Níveis WCAG cumpridos:** A

#### 5.4.2.2 Página Run Simulations

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 12 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 1 alerta, inofensivo para os níveis do WCAG
- **Níveis WCAG cumpridos:** A

#### 5.4.2.3 Página Datasets

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 12 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 2 alertas, um deles relativo às diretrizes 1.3.1 e 2.4.1 do WCAG
- **Níveis WCAG cumpridos:** A, com alguns alertas

#### 5.4.2.4 Página Experiment Configs

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 12 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 1 alerta, inofensivo para os níveis do WCAG

- **Níveis WCAG cumpridos:** A

#### 5.4.2.5 Página Use Cases

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 16 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 1 alerta, inofensivo para os níveis do WCAG
- **Níveis WCAG cumpridos:** A

#### 5.4.2.6 Página Schema

- **Erros:** 0 erros, ou seja, cumpre com todos os parâmetros do nível A
- **Erros de Contraste:** 12 erros de contraste, não cumpre com a diretriz 1.4.3 do nível AA
- **Alertas:** 62 alertas, maioritariamente relativos às diretrizes 1.1.1, 1.3.1 e 3.3.2 do WCAG nível A e a diretriz 2.4.6 do nível AA
- **Níveis WCAG cumpridos:** A, com alguns alertas

Após a análise extensiva de cada página em cada diretriz, podemos tirar algumas conclusões relevantes:

- A interface possui um número significativo de erros de contraste. Apesar de estes estarem associados à diretriz 1.4.3 do nível AA, deve ser algo a ter em conta no futuro para melhorar na interface.
- Existem alguns alertas, especialmente relativos às diretrizes 1.1.1, 1.3.1, 2.4.1 e 3.3.2 do WCAG nível A e a diretriz 2.4.6 do nível AA. Deve haver um cuidado especial no futuro de forma a resolver estes alertas.

De uma forma geral a interface cumpre com o nível A do WCAG tendo os níveis mínimos aceitáveis de acessibilidade. No entanto é de notar que é necessário tomar ações relativas aos alertas encontrados, de forma a evitar que possam tornar-se possíveis erros no futuro.

## 5.5 Testes

Outro fator importante a ter em conta durante o desenvolvimento da solução, foi a necessidade de desenvolver formas de testar as funcionalidades de forma a verificar se cumpriam com os requisitos delineados. Foram então feitos um conjunto de testes E2E (End-To-End) utilizando a ferramenta Cypress.

### 5.5.1 Testes E2E (Cypress)

O Cypress é uma ferramenta de automação *open-source* e gratuita para testes de software. É desenvolvida em JavaScript e utilizada por instituições globais de prestígio, como a NASA e a DHL. Trata-se de uma estrutura de testes E2E, desenvolvida por programadores com o objetivo de validar aplicações web.

Foca-se em remover incongruências nos testes, assegurando que os utilizadores conseguem redigir, depurar e correr testes no browser sem precisar de configurações ou pacotes extras. Opera como uma aplicação autónoma e pode ser instalado em sistemas operacionais macOS,

Unix/Linux e Windows utilizando aplicações hifenizadas ou ferramentas de linha de comando. O Cypress foi criado principalmente para programadores que codificam em JavaScript, pois pode ser utilizado para testar qualquer aplicação executada num browser. (Al-Ajily, 2022)

### 5.5.2 Descrição dos Testes

Para iniciar o processo dos testes foi necessário fazer a instalação do Cypress no projeto utilizando o comando `npm install cypress --save-dev`. Após ser concluída a instalação, foi utilizado o comando `npx cypress open`, para aceder à interface do Cypress. Este será o local onde os testes serão realizados.

Em primeiro lugar foi necessário escolher o tipo de testes a fazer, para o caso em questão são os testes E2E, e em seguida foi solicitada a escolha do browser onde os testes serão feitos. Foi escolhido fazer os testes no Google Chrome por ser dos browsers mais utilizados. Após as configurações iniciais estarem concluídas os testes E2E ficam automaticamente associados à pasta “e2e” criada dentro da pasta “cypress” no projeto. Aparece posteriormente uma interface com mostra a Figura 77, na qual estão agrupados todos os ficheiros de teste feitos.

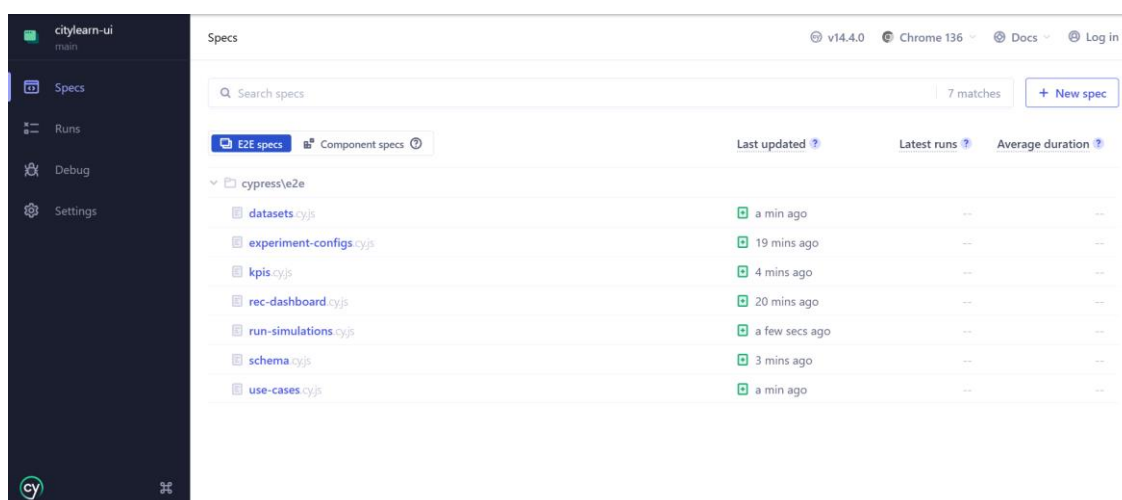


Figura 77 - Lista de testes por página

Cada ficheiro da lista corresponde aos testes feitos dentro da página com o nome correspondente. Ao clicar num dos ficheiros, todos os testes serão executados aparecendo do lado esquerdo todos os passos para cada teste e do lado direito aparece as ações no browser respetivas como se fosse o próprio utilizador a fazê-las.

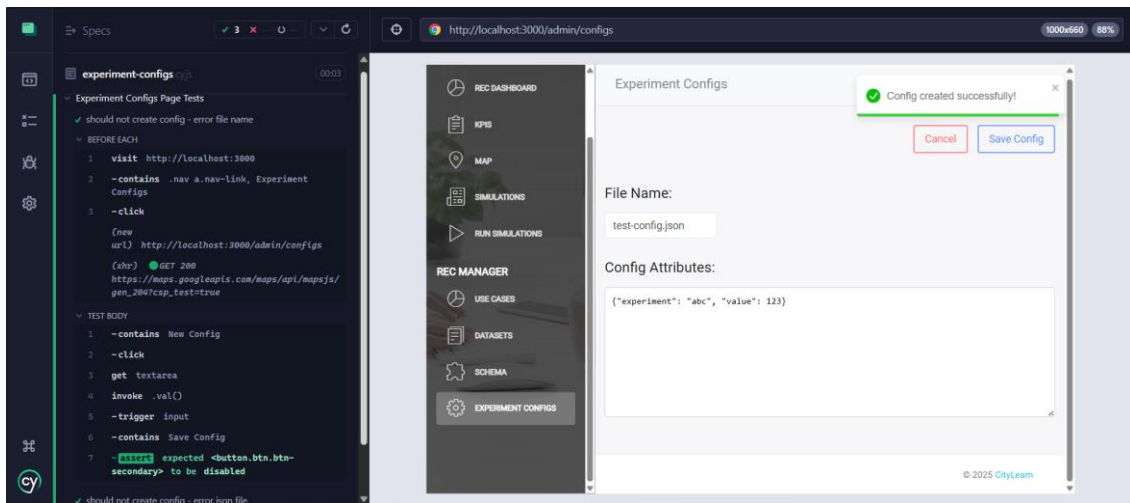


Figura 78 - Estrutura da execução dos testes

Com base nos testes realizados, chegou-se à conclusão de que a interface se encontra bem estruturada e preparada para uso futuro pelos utilizadores do CityLearn, tendo sido testados os cenários tanto de sucesso como de erro de forma a garantir que os diferentes fluxos de atividade possíveis de realizar na interface são praticáveis garantindo assim a integridade da interface.



## 6 Considerações Finais e Trabalho Futuro

Concluindo, esta dissertação tem como objetivo a criação de uma interface com o intuito de melhorar significativamente a experiência dos utilizadores do CityLearn, tornando a aplicação mais intuitiva, acessível e eficiente. Apesar do projeto focar-se maioritariamente na melhoria da aplicação, este pretende também servir como um teste no qual é possível avaliar o efeito e os benefícios que a adoção de uma interface bem concebida pode oferecer a aplicações baseadas em simulações, podendo assim estender o seu impacto a muitos outros projetos existentes na área.

Tendo o planeamento devidamente estruturado, com as tarefas e o fluxo de trabalho delineado e uma análise prévia dos possíveis riscos que podem impactar o trabalho, o estudante acredita ter tido todas as ferramentas necessárias para garantir o sucesso do trabalho ao qual se propôs realizar e também aproveitar o mesmo para melhorar as suas competências pessoais e profissionais com o grande objetivo final de evoluir na carreira profissional.

### 6.1 Objetivos Concretizados

Através do desenvolvimento das diferentes páginas, o estudante conseguiu implementar uma interface que aprimora bastante a experiência dos utilizadores do CityLearn para a visualização e análise de dados, cumprindo assim com os objetivos O2 e O3. Algumas das páginas como a página do Schema oferece uma estrutura que permite a criação de esquemas de uma forma bem mais intuitiva e simples, simplificando aquilo que previamente era uma tarefa árdua e confusa para os utilizadores.

A adição das ações para criação de ficheiros de configuração aprimorou também a experiência do utilizador para definir configurações para as simulações, simplificando o processo.

Relativamente à análise feita dos requisitos funcionais e não funcionais propostos, cumprindo assim com o objetivo O1, todos estes estão representados na interface seja através das páginas

desenvolvidas, da estrutura da interface e da forma como a navegação é feita ou de como os componentes interagem entre si.

A realização da análise de métricas e testes para garantir a qualidade e o bom funcionamento da interface garante também o cumprimento do objetivo O4.

## 6.2 Trabalho Futuro

Relativamente ao progresso do projeto, já foi feito algum trabalho preliminar para a continuação do desenvolvimento da UI relativamente à ideia de implementar uma vista em mapa na qual é possível ver o mapa com alguns pontos indicados com diferentes ícones, que representam casas e veículos elétricos, tal como demonstra a Figura 79.

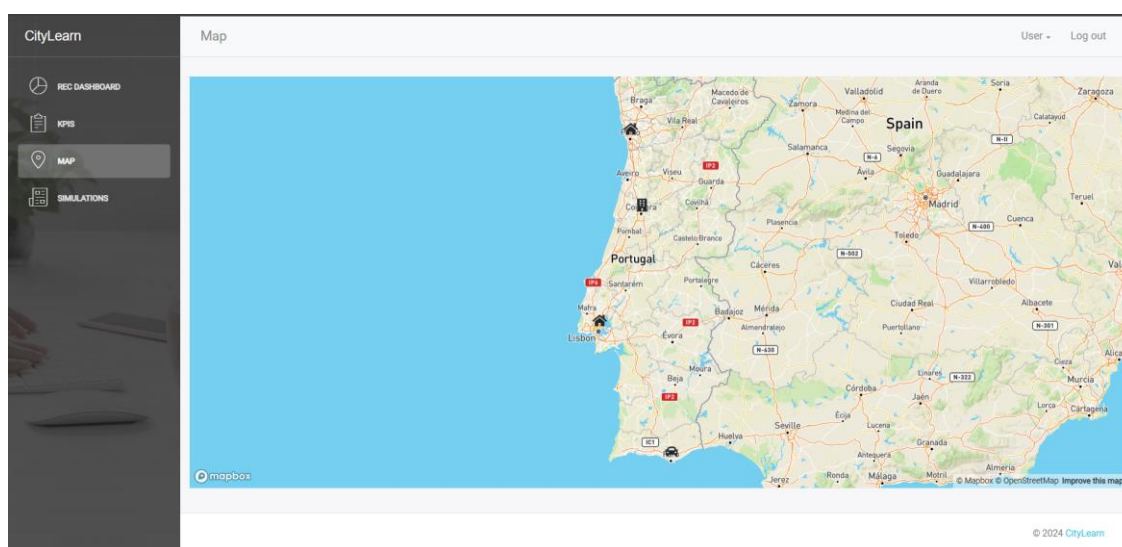


Figura 79 - Página da vista em mapa

No entanto, é de notar que foi uma ideia que se decidiu ser implementada numa fase futura do projeto devido à sua complexidade.

Para além disto, é necessário manter um constante cuidado na estrutura das páginas previamente feitas e tentar sempre otimizar as mesmas e continuar a procurar formas cada vez mais intuitivas e benéficas para os utilizadores de representar e analisar os dados, sendo este projeto uma base sólida e motivação para a contínua evolução da interface desenvolvida e a procura de cada vez mais melhorar a mesma, de forma a garantir a qualidade desejada pelos utilizadores do CityLearn.

# Referências

- Akram, M., Ali, G. A., Sulaiman, A., & ul Hassan, M. (2023). Accessibility evaluation of Arabic University websites for compliance with success criteria of WCAG 1.0 and WCAG 2.0. In *Universal Access in the Information Society* (Vol. 22, Issue 4, pp. 1199–1214). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1007/s10209-022-00921-8>
- Al-Ajily, M. (2022). *Automated Testing for React Web Application with Cypress*.
- Alsaeedi, A. (2020). Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: Proposed frameworks. *Information (Switzerland)*, 11(1). <https://doi.org/10.3390/info11010040>
- Cardoso, Alberto., Alves, G. R. ., & Restivo, Teresa. (2020). *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON) : date and venue, 27-30 April, 2020, Porto, Portugal*. IEEE.
- Chatziemmanouil, T., & Katsanos, C. (2024). Accessibility Academy: Interactive Learning of the WCAG 2.1 Web Accessibility Guidelines. *IEEE Global Engineering Education Conference, EDUCON*. <https://doi.org/10.1109/EDUCON60312.2024.10578915>
- Chen, S., Thaduri, U. R., & Ballamudi, V. K. R. (2019). Front-End Development in React: An Overview. *Engineering International*, 7(2), 117–126. <https://doi.org/10.18034/ei.v7i2.662>
- Comparative analysis of angularjs and reactjs. (2016). *International Journal of Latest Trends in Engineering and Technology*, 7(4). <https://doi.org/10.21172/1.74.030>
- Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., Witte, M. J., & Glazer, J. (n.d.). *EnergyPlus: creating a new-generation building energy simulation program*.
- Diário da República, 2.ª série PARTE E Artigo 2.º*. (n.d.).
- Diniz, R. N. V., Figueiredoy, C. C. L., De S.Russo, G., Bahiense, M. R. G., Arbex, M. V. L., Dos Santos, L. M., Da Rocha, R. F., Bezerra, R. R., & Giuntini, F. T. (2022). Evaluating the performance of web rendering technologies based on JavaScript: Angular, React, and Vue. *Proceedings - 2022 48th Latin American Computing Conference, CLEI 2022*. <https://doi.org/10.1109/CLEI56649.2022.9959901>
- Enhancing Readability, Accessibility, and Shareability of Robot Framework Test Results*. (2023).
- Fatimatuzzahra, N., Priyadi, Y., & Nurtantyana, R. (2025). Functional and Non-Functional Requirement (FR and NFR) Formation Based on Requirement Elicitation Using Text

- Semantics in IdVar4CL Artifact. *2025 10th International Conference on Signal Processing and Communication (ICSC)*, 421–426.  
<https://doi.org/10.1109/ICSC64553.2025.10968323>
- Fonseca, T., Ferreira, L., Cabral, B., Severino, R., & Praça, I. (2024). *EnergAlze: Multi Agent Deep Deterministic Policy Gradient for Vehicle-to-Grid Energy Management*.
- Garmendia, A. I., Morri, F., Cappart, Q., & Le Cadre, H. (n.d.). *Winning the 2023 CityLearn Challenge: a Community-based Hierarchical Energy Systems Coordination Algorithm*. <https://hal.science/hal-04685791v1>
- Gotterbarn, D., Miller, K., & Rogerson, S. (n.d.). *Code Ethics*.
- Jake Spurlock. (2013). *Bootstrap: Responsive Web Development*. O'Reilly Media, Inc.
- Joaquim Pessôa Filho. (2024). *Programação orientada a objetos com C#* (Senac São Paulo, Ed.).
- Jokste, L., & Kampars, J. (2024). Comprehensible software architecture visualization with C4 model for WRM ecosystem members. *ITMS - International Scientific Conference on Information Technology and Management Science of Riga Technical University, 2024*. <https://doi.org/10.1109/ITMS64072.2024.10741937>
- Kumpulainen, T. (2021). *Web application development with Vue.js*.
- Miraz, M. H., Ali, M., & Excell, P. S. (2021). Adaptive user interfaces and universal usability through plasticity of user interface design. In *Computer Science Review* (Vol. 40). Elsevier Ireland Ltd. <https://doi.org/10.1016/j.cosrev.2021.100363>
- Nagy, Z., Vázquez-Canteli, J. R., Dey, S., & Henze, G. (2021). The citylearn challenge 2021. *BuildSys 2021 - Proceedings of the 2021 ACM International Conference on Systems for Energy-Efficient Built Environments*, 218–219.  
<https://doi.org/10.1145/3486611.3492226>
- Orfanoudakis, S., Diaz-Londono, C., Yilmaz, Y. E., Palensky, P., & Vergara, P. P. (2024). *EV2Gym: A Flexible V2G Simulator for EV Smart Charging Research and Benchmarking*. <http://arxiv.org/abs/2404.01849>
- Pratama, M. A. T., & Cahyadi, A. T. (2020). Effect of User Interface and User Experience on Application Sales. *IOP Conference Series: Materials Science and Engineering*, 879(1). <https://doi.org/10.1088/1757-899X/879/1/012133>
- Suman Aryal *BOOTSTRAP-A Front-End Framework For Responsive Web Design*. (n.d.).
- Tiun, S., Mokhtar, U. A., Bakar, S. H., & Saad, S. (2020). Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. *Journal of Physics: Conference Series*, 1529(4). <https://doi.org/10.1088/1742-6596/1529/4/042077>

Vázquez-Canteli, J. R., Dey, S., Henze, G., & Nagy, Z. (2020). The CityLearn Challenge 2020. *BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 320–321. <https://doi.org/10.1145/3408308.3431122>

Vázquez-Canteli, J. R., Kämpf, J., Henze, G., & Nagy, Z. (2019). CityLearn v1.0: An OpenAI gym environment for demand response with deep reinforcement learning. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 356–357. <https://doi.org/10.1145/3360322.3360998>

*Web application for questionnaire creation and monitoring in React.js.* (n.d.).

Yeh, C., Li, V., Datta, R., Arroyo, J., Christianson, N., Zhang, C., Chen, Y., Hosseini, M., Golmohammadi, A., Shi, Y., Yue, Y., & Wierman, A. (n.d.). *SustainGym: Reinforcement Learning Environments for Sustainable Energy Systems*. <https://github.com/chrisyeh96/sustaingym/>

Zhou, Y. (n.d.). *Exploring the benefits of animation in user interface design Faculty of Design and Creative Technologies Colab: Creative Technologies 2.*

Ably, 2023. *How to use Next.js and Recharts to build an information dashboard.* [Online] Available at: <https://ably.com/blog/informational-dashboard-with-nextjs-and-recharts> [Acedido em 21 12 2024].

Kingsley Nweye, A. W. H. P. Y. A. Z. N., s.d. *CityLearn: A Tutorial on Reinforcement Learning Control for Grid-Interactive Efficient Buildings and Communities.* [Online] Available at: [https://colab.research.google.com/github/climatechange-ai-tutorials/citylearn/blob/main/citylearn\\_ccai\\_tutorial.ipynb#scrollTo=WieiE91Fx0xr](https://colab.research.google.com/github/climatechange-ai-tutorials/citylearn/blob/main/citylearn_ccai_tutorial.ipynb#scrollTo=WieiE91Fx0xr) [Acedido em 21 12 2024].

OPEVA, s.d. *OPEVA.* [Online] Available at: <https://opeva.eu/>. [Acedido em 12 2024].

Schmidl, C., 2020. *How To Read Scientific Papers.* [Online] Available at: <https://towardsdatascience.com/how-to-read-scientific-papers-df3afd454179> [Acedido em 12 2024].

SoftCPS, s.d. *SoftCPS – Software Technologies for Cyber-Physical Systems.* [Online] Available at: <https://www2.isep.ipp.pt/softcps/> [Acedido em 21 12 2024].

United Nations, Department of Economic and Social Affairs, s.d. *The 17 Goals | Sustainable Development*. [Online]

Available at: <https://sdgs.un.org/goals>

[Acedido em 12 2024].