



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA (ISEL)

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE
TELECOMUNICAÇÕES E COMPUTADORES (DEETC)

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA

UNIDADE CURRICULAR DE PROJETO

Ferramenta de visualização de dados

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Alice Fernandes 45741

Orientador(es)

Professor Paulo Trigo (ISEL)

Professor Artur (ISEL)

Professor Helder Fanha (ISCAL)

Junho, 2025

Resumo

Este relatório apresenta o desenvolvimento de uma ferramenta de visualização de dados, desenhada para apoiar os alunos que utilizam a plataforma *Marketplace Simulations - International Corporate Management*, durante o semestre. O principal objetivo da aplicação é simplificar a análise de dados, permitindo aos alunos carregar ficheiros Excel Spreadsheet Format (XLSX) exportados da plataforma de simulação, processar e normalizar a informação carregada e mostrar gráficos interativos e dinâmicos.

A solução proposta introduz um modelo de dados baseado nos períodos de simulação da plataforma, em que os utilizadores podem gerir os dados. Cada ficheiro carregado é analisado e convertido em ficheiros Comma-Separated Values (CSV) estruturados através de uma pipeline de normalização, construída utilizando *Python* e *Pandas*. Os conjuntos de dados resultantes são visualizados através de gráficos dinâmicos com recurso ao Plotly incorporados na interface gráfica *web*.

O projeto segue um design modular e centrado no utilizador, utilizando o *Django* como a biblioteca base da plataforma. As principais contribuições incluem a criação de um modelo de organização de ficheiros, automatismos no processamento de dados e uma interface gráfica intuitiva que vai de encontro às necessidades dos utilizadores. O sistema foi desenvolvido de forma incremental e está preparado para ser implementado num ambiente de produção, suportando vários utilizadores com isolamento de dados, controlo de versões.

Abstract

This report presents the development of a data visualization tool designed to support students using the Marketplace Simulations - International Corporate Management platform during the semester. The main aim of the application is to simplify data analysis by allowing users to upload XLSX files exported from the simulation platform, automatically process and normalize the data and create interactive and dynamic graphs.

The proposed solution introduces a data model based on the platform's simulation periods, in which users can manage the data. Each uploaded file is analysed and converted into structured CSV files through a normalization pipeline built using *Python* and *Pandas*. The resulting data sets are visualized through dynamic graphs using Plotly incorporated into the *web* graphical interface.

The project follows a modular, user-centered design, using *Django* as the platform's base library. The main contributions include the creation of a file organization model, automated data processing and an intuitive graphical interface that meets users' needs. The system was developed incrementally and is ready to be implemented in a production environment, supporting multiple users with data isolation and version control.

Agradecimentos

Escrever aqui eventuais agradecimentos ...

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	vii
Lista de Tabelas	ix
Lista de Figuras	xi
Lista de Acrónimos	xiii
1 Introdução	3
1.1 Motivação	3
1.2 Objetivos	4
2 Trabalho Relacionado	5
3 Modelo Proposto	7
3.1 Requisitos	7
3.1.1 Casos de Utilização	8
3.2 Fundamentos	9
3.3 Abordagem	9
3.3.1 Web Scraping	10
3.3.2 Exportação e carregamento manual de ficheiros	11

4	Implementação do Modelo	15
4.1	Arquitetura Tecnológica	15
4.2	Análise dos dados	15
4.2.1	Análise inicial da informação	15
4.2.2	Classificação da informação	15
4.2.3	Visualização escolhidas para os vários grupos	16
4.3	Processamento dos dados	16
4.3.1	<i>Pipeline</i> de transformação dos dados	16
4.3.2	Conversão e Exportação para CSV	16
4.4	Desenvolvimento da Aplicação Web	17
4.4.1	Arquitetura da aplicação Django	17
4.4.2	Endpoints e API interna	17
4.4.3	Sistema de processamento assíncrono e normalização	17
4.5	Interface Gráfica e Experiência do Utilizador (Frontend)	18
4.5.1	Design System e Flowbite	18
4.5.2	WebComponents e gráficos interativos	18
4.5.3	Gestão de Quarters e Uploads	18
5	Validação e Testes	19
6	Conclusões e Trabalho Futuro	21
A	Sistemas de Apoio à Decisão	23
B	Casos de Utilização - UML	25
C	Tabela de Requisitos funcionais	27
D	Tabela de Requisitos não funcionais	29
E	Classificação das folhas XLSX	31
	Bibliografia	33

Lista de Tabelas

5.1	Uma tabela	19
-----	----------------------	----

Lista de Figuras

5.1	Uma figura	19
-----	----------------------	----

Lista de Acrónimos

CSV Comma-Separated Values. i, iii, viii, 1, 7, 11, 12, 16

ETL Extract, Transform, Load. 13

ISCAL Instituto Superior de Contabilidade e Administração de Lisboa. 3, 5, 9

SAD Sistemas de Apoio à Decisão. viii, 5, 23

XLSX Excel Spreadsheet Format. i, iii, viii, 4, 7, 11, 12, 15, 16, 31, 32

No sistema, o CSV pe

Capítulo 1

Introdução

No contexto do ensino superior, a integração de ferramentas tecnológicas que potencializem o processo de aprendizagem é cada vez mais valorizada. Particularmente, em ambientes que simulam situações empresariais reais, como é o caso do simulador utilizado pelos alunos do Instituto Superior de Contabilidade e Administração de Lisboa (ISCAL). A análise eficiente de dados torna-se essencial para a tomada de decisões estratégicas e tem impacto na avaliação final dos alunos, no entanto, a complexidade e a falta de visuais que ajudem a perceber as informações apresentadas pela plataforma original podem representar um desafio significativo para os estudantes.

1.1 Motivação

O presente projeto surgiu da necessidade entre os alunos do ISCAL que utilizam o simulador empresarial *Marketplace Simulations - International Corporate Management* que é um simulador de negócios internacionais, onde os alunos se agrupam em empresas e simulam a criação de um negócio num mercado internacional. Embora a plataforma forneça toda a informação necessária à tomada de decisões na simulação, esses dados estão dispersos em múltiplas secções e apresentados em tabelas, sem funcionalidades de visualização gráfica ou filtragem. Esta limitação obriga os alunos a alternar entre páginas, copiar dados manualmente e criar folhas de cálculo externas, comprometendo tanto a eficiência quanto a qualidade da análise.

1.2 Objetivos

A aplicação proposta neste relatório pretende ajudar nesse sentido, oferecendo uma interface funcional que permite aos utilizadores carregar dados retirados da plataforma de simulação e tornar esses ficheiros em visualizações que podem ser consultadas e manipuladas. A aplicação permitirá aos utilizadores:

- Criar uma conta na plataforma que permita persistir a informação carregada.
- Carregar ficheiros XLSX exportados da plataforma original;
- Visualizar os dados em gráficos interativos;

Do ponto de vista técnico, queremos que projeto adote uma arquitetura fácil de manter e que vá de encontro à utilização da plataforma, dando ênfase aos seguintes itens:

- A normalização e transformação automática de dados provenientes de fontes externas;
- A facilidade na gestão de utilizadores e ficheiros, com o objetivo de oferecer uma experiência intuitiva para os utilizadores finais.
- Organizar a informação por utilizador, garantindo que o utilizador apenas consegue consultar a informação carregada.
- Adotar um modelo de organização semelhante à plataforma de simulação, de modo a tornar a experiência de utilização mais intuitiva e garantido que a nossa aplicação tenha fronteiras claras de utilização.

Ao longo deste relatório, serão detalhadamente apresentadas as decisões tomadas, bem como os fundamentos que orientaram o desenvolvimento da aplicação proposta.

Capítulo 2

Trabalho Relacionado

O presente projeto insere-se num contexto mais vasto de ferramentas pedagógicas e Sistemas de Apoio à Decisão (SAD), ainda que neste caso concreto, em ambientes simulados, no ensino superior. No âmbito do ISCAL, é recorrente a utilização de plataformas como a que iremos descrever nos seguintes capítulos *Marketplace Simulations*, que permitem aos estudantes desenvolver competências práticas em ambientes virtuais de negócios, replicando o funcionamento de mercados reais. Esta necessidade de suporte digital à análise e simulação motivou o desenvolvimento de outras ferramentas auxiliares, com destaque para um projeto também realizado em parceria com o ISCAL, focado em simulações parciais de modelos económicos.

Esse projeto, embora partilhe uma motivação semelhante (criar micro-simulações) apresenta uma abordagem distinta. Em particular, a aplicação permite simular cenários específicos com base em inputs manuais, o que pode ser útil para quando se procura prever resultados com foco muito concreto (por exemplo, simular o impacto de uma única variável nos resultados). Esta plataforma foca-se numa outra vertente, em que pretende ser uma ferramenta para auxiliar o ensino de forma prática.

O projeto atual distingue-se, tanto pelo seu foco como pela componente técnica. Ao contrário do sistema anterior, que se baseava em simulações parametrizadas modelos estáticos, esta nova aplicação aposta numa abordagem orientada a dados, onde a informação real dos jogos de simulação é carregada diretamente pelo utilizador. Essa escolha trouxe consigo a necessidade de resolver questões de normalização, estruturação e tratamento de dados, algo que não era pedido no projeto relacionado.

Este trabalho assume, portanto como pressuposto, o uso de dados extraídos diretamente da plataforma de simulação, e a valorização da experiência do utilização na apresentação de dados. Em conjunto, estes elementos definem uma solução mais abrangente e alinhada com os desafios técnicos reais da análise de dados em contexto educativo, indo além das simulações reduzidas ou controladas que caracterizavam os trabalhos relacionados anteriormente desenvolvidos.

Capítulo 3

Modelo Proposto

(falta introduzir o modelo proposto)

3.1 Requisitos

Aqui o essencial (e se aplicável) dos requisitos funcionais, não funcionais e modelo de casos de utilização. Aqui deve também apresentar matriz para decisão sobre prioridade dos casos de utilização (se aplicável) ...

Deve apresentar de forma “moderada” o resultado da fase avaliação de requisitos. A informação de maior detalhe (e.g., diagramas UML demasiado detalhados) deve ser colocada em apêndice.

Requisitos funcionais

Os requisitos funcionais descrevem as funcionalidades específicas que a aplicação deve oferecer para atender às necessidades dos utilizadores. No contexto deste projeto, definem as ações que o sistema deve ser capaz de executar. No nosso projeto, identificamos os seguintes requisitos:

- **Visualizações interativas:** a aplicação deve conseguir mostrar gráficos com base nos dados carregados, com suporte a alteração de parâmetros em tempo real (por exemplo, mudar de quarter ou selecionar um país específico).
- **Upload e processamento de ficheiros XLSX:** os utilizadores devem conseguir carregar ficheiros *XLSX* com múltiplas folhas; esses ficheiros são automaticamente convertidos para CSV e normalizados.

- **Autenticação e gestão de utilizadores:** cada utilizador tem uma conta e pode gerir os seus próprios dados. Apenas os ficheiros mais recentes serão considerados para as visualizações.
- **Gestão de quarters:** os utilizadores podem criar períodos temporais (quarters), cada um identificado por um número, que funcionam como *bucket* lógicos para organizar os ficheiros carregados.

Requisitos não funcionais

Alguns requisitos não funcionais foram igualmente críticos para garantir a robustez e usabilidade do sistema (TODO):

- **Normalização da informação:** A *pipeline* de processamento é isolada e modular, facilitando a manutenção e futura extensão do sistema.
- **Normalização dos dados:** foram aplicadas rotinas automáticas para garantir coerência nos nomes de colunas, remoção de quebras de linha, e eliminação de colunas irrelevantes.
- **Isolamento por utilizador:** cada utilizador só pode aceder aos seus dados, e visualizar a informação que carregou.
- **Experiência de utilizador:** A plataforma tem de ser capaz de oferecer uma boa experiência de utilização.

(Colocar as actual tabelas no apendice)

3.1.1 Casos de Utilização

(TODO)

- Colocar UML como apendice - Referenciar a matriz de prioridade dos casos de utilização Com os casos de utilização estabelecidos, foram identificados então os requisitos da plataforma, que podemos separar em requisitos funcionais e não funcionais

Falta a matriz de prioridade dos casos de utilização

3.2 Fundamentos

A Marketplace Simulations é uma empresa que desenvolve plataformas de simulação para fins educativos, ou seja, ferramentas que colocam os estudantes numa espécie de "jogo" onde cada equipa gere a sua própria empresa e compete com os colegas em cenários de mercado realistas. Isto acaba por funcionar como uma espécie de laboratório virtual, usado em faculdades de gestão e economia, como no ISCAL, onde os alunos aplicam os conceitos aprendidos numa experiência em contexto educativo.

No caso concreto do nosso projeto, o módulo em questão chama-se *International Corporate Management* (referida doravante como plataforma de simulação), e é utilizado tipicamente no último semestre, na cadeira Projeto de Simulação em Negócios Internacionais da Licenciatura de Comércio e Negócios Internacionais. Aqui, cada grupo de alunos representa uma empresa que tem de atuar num mercado competitivo — tomando decisões sobre posicionamento de produto, investimento, preços, distribuição, contratação de equipas e por aí fora. Essas decisões são processadas pela plataforma, que simula o comportamento do mercado com base num algoritmo interno.

No final de cada ronda (ou trimestre), os dados são disponibilizados na plataforma nas várias secções disponíveis numa visualização tabular, o que, na prática, faz com que os alunos saltem entre secções, fazer gráficos à mão e tirar conclusões com base em tabelas.

Em termos de estrutura temporal, a simulação decorre ao longo de vários períodos, normalmente designados por quarters. Cada um destes quarters representa uma fase do ciclo de vida da empresa simulada — incluindo decisões estratégicas, execução e análise de resultados. Ou seja, no final de cada trimestre, os alunos recebem os dados com os resultados das decisões anteriores, o que obriga a uma análise comparativa constante entre períodos. É precisamente esta lógica iterativa — decidir, analisar, ajustar, repetir — que dá ritmo à simulação e aproxima o exercício de uma situação real de gestão empresarial.

3.3 Abordagem

O projeto proposto pretende então ajudar nesse sentido, de conseguir transformar dados desta plataforma em algo mais fácil e rápido de analisar. Tento

o contexto da plataforma, iremos então descrever duas abordagens que considerámos.

3.3.1 Web Scraping

A primeira abordagem que consideramos foi a hipótese de automatizar a extração dos dados diretamente da plataforma do Marketplace Simulations, através de técnicas de web scraping. A ideia parecia interessante numa fase conceptual, já que permitiria reduzir a dependência do utilizador no processo de exportação manual dos dados. No entanto, rapidamente percebemos que esta abordagem trazia vários desafios que, na prática, a tornavam pouco viável, ou mesmo arriscada.

Primeiro, cada conta na plataforma está associada a um grupo de alunos, ou seja, é uma conta ativa e personalizada, usada diretamente durante a simulação. Isto significa que qualquer processo automático que iniciasse sessão, mesmo que fosse só para leitura, poderia inadvertidamente interagir com a interface e acabar por alterar alguma opção crítica — o que seria desastroso num contexto académico em que cada decisão tem impacto na avaliação. Além disso, como o acesso à plataforma é feito por licenças pagas, não existe qualquer possibilidade de criar uma service account ou utilizador apenas para leitura com permissões de administração. Ou seja, qualquer tentativa de scraping teria de reutilizar credenciais reais, o que levanta não só questões de segurança, mas também (possivelmente) legais.

Outro fator que pesou na decisão foi o próprio risco técnico do scraping: plataformas deste tipo estão muitas vezes protegidas com mecanismos anti-automação (por exemplo: desafios CAPTCHA), e não conhecendo em detalhe a aplicação, poderíamos facilmente encontrar barreiras inesperadas, ou mesmo cair em práticas que fossem contra os termos de uso do serviço.

Por todos estes motivos, optámos por não seguir esta via. Em vez disso, definimos como parte do fluxo normal da aplicação que os próprios alunos devem exportar dados a partir da plataforma de simulação e, de seguida, carregar para a nossa aplicação. Esta solução, embora mais manual, garante segurança, respeita a integridade das contas dos utilizadores, e evita problemas legais ou técnicos com a aplicação de simulação.

3.3.2 Exportação e carregamento manual de ficheiros

A abordagem que acabamos por usar foi os alunos exportam manualmente os dados diretamente da plataforma de simulação e carregarem esses ficheiros na nossa plataforma. A partir daí, o sistema processa automaticamente os dados, normaliza os conteúdos e cria visualizações interativas. Esta abordagem, apesar de requerer uma ação manual inicial por parte do utilizador, é mais segura e prática quando comparada com a alternativa de web scraping, que, como vimos acima, apresentava vários desafios que poderiam ir contra os termos do serviço. Deste modo, garantimos um equilíbrio entre usabilidade, segurança e fiabilidade do sistema. Acabamos então por criar um *sad* (cf., capítulo A) que permita os estudantes tomarem melhores decisões.

Para isso, procurámos que a nossa aplicação refletisse a estrutura da própria simulação. Como tal, os dados são organizados por *quarters*, como acontece na plataforma de simulação e permitir o carregamento dos dados exportados. Estes ficheiros são depois processados o que nos permite trabalhar com dados mais consistentes.

Esta estrutura base implica a existência de três entidades principais na nossa aplicação (que iremos descrever a seguir) e cuja interação define o funcionamento base do sistema.

Quarters

Os *quarters* funcionam como *buckets* lógicos para organizar os ficheiros carregados pelos utilizadores. Cada utilizador pode criar múltiplos *quarters*, identificados de forma única por um número. Este número serve também de identificador, sendo que não é possível ter dois *quarters* identificados como 1 para cada utilizador.

A nível de implementação, cada *quarter* está associado unicamente a um utilizador e é identificado por um UUID (gerado automaticamente pelo *Django*), que garante que o *quarter* seja único.

Ficheiros

Os ficheiros são inicialmente carregados no formato XLSX, contendo uma ou várias folhas de cálculo. Cada folha é tratada como uma entidade individual e transformada para CSV. O ficheiro XLSX é guardado como referência, mas

não é diretamente utilizado para visualização.

Este processo de conversão para CSV é acompanhado por uma *pipeline* de normalização de dados, que limpa os dados (como por exemplo, remover quebras de linha, colunas sem representação, nomes inconsistentes) e aplica regras para que os gráficos possam ser gerados de forma consistente. Cada ficheiro CSV gerado é associado ao seu ficheiro XLSX de origem, ao quarter correspondente, e o seu nome será baseado no nome da folha de onde foi extraído. Aos ficheiros CSV é também associado uma *slug*, baseado também no nome, que identifica a informação que o CSV representa.

A plataforma garante que só existe uma versão ativa de cada ficheiro por tipo — caso o utilizador carregue novamente um ficheiro com o mesmo nome lógico, o anterior será marcado como não ativo, evitando duplicações e garantindo que os gráficos usam apenas dados mais recentes.

Utilizadores

A plataforma foi desenhada para funcionar com utilizadores. Cada utilizador tem a sua conta, e pode criar quarters, carregar ou alterar ficheiros, e ver aos gráficos gerados a partir desses ficheiros.

Cada utilizador tem acesso apenas aos seus próprios dados, garantindo o isolamento da informação. Esta separação é feita a nível da base de dados, através da associação de cada entidade ao utilizador que criou.

Apesar da plataforma não suportar explicitamente equipas ou grupos, assume-se que alunos do mesmo grupo podem carregar ficheiros semelhantes, mas o sistema trata-os como ficheiros diferentes. Assim, evita-se a complexidade adicional de gerir permissões ou partilha de dados entre contas. Também se assume que as contas podem ser criadas ao nível do grupo, pelo que para a plataforma, é indiferente se a conta é individual ou partilhada entre membros desse grupo.

No futuro, pode ser considerada a funcionalidade de desativação automática de contas (por exemplo, após o final do semestre), mas para já o modelo é simples e robusto: conta individual, dados isolados, e controlo completo sobre os próprios uploads.

Pipeline de Normalização de Dados

Outro contributo diferenciador foi o desenvolvimento de uma *pipeline* modular de normalização de dados. O objetivo é garantir que os ficheiros carregados, que muitas vezes contêm nomes de colunas inconsistentes, quebras de linha, espaços em excesso ou colunas irrelevantes, sejam adaptados para serem visualizados.

Como podemos receber muitos ficheiros, a variabilidade entre os dados recebidos é muito alta, pelo que alguns dados passam por mais do que uma fase de normalização. Esta decisão foi tomada com base numa análise manual, em que identificámos possíveis fontes de dados que precisam de mais do que uma fase de normalização. As várias fases de normalização alteram os dados de modo a facilitar a representação visual dos mesmos e é um passo essencial no projeto, porque garante que a aplicação trabalha com formatos e regras conhecidas, e remove a variabilidade dos ficheiros importados.

As fases de normalização irão ser descritas em mais detalhe nos capítulos seguintes, uma vez que a implementação destas *pipeline* estão relacionadas à tecnologia escolhida, mas o desenvolvimento desta *pipeline* é um fator diferenciador deste projeto, uma vez que tem de lidar com dados que não estão estruturados de forma a facilitar representações visuais.

Este processo de normalização é semelhante aos processos Extract, Transform, Load (ETL) ainda que neste projeto tenha sido desenvolvido com uma escala menor.

Capítulo 4

Implementação do Modelo

O desenvolvimento da aplicação assentou em fundamentos sólidos tanto ao nível tecnológico como ao nível das boas práticas de engenharia de software e tratamento de dados. A escolha da arquitetura, das tecnologias utilizadas e dos modelos de organização dos dados foi orientada por critérios de fiabilidade, modularidade, simplicidade de manutenção e escalabilidade.

4.1 Arquitetura Tecnológica

- Falar sobre *stack* usada no projeto, alternativas e prós e contras

4.2 Análise dos dados

Nesta secção iremos descrever como analisamos a informação recebida da plataforma de simulação e as decisões tomadas sobre essa informação.

4.2.1 Análise inicial da informação

Após termos recebido os dados, fizemos uma primeira análise com o objetivo de perceber a informação recebida e os próximos passos a tomar. No total, recebemos 39 ficheiros XLSX que foram exportados da plataforma,

4.2.2 Classificação da informação

- Falar sobre como classificamos os 39 ficheiros Excell em "buckets" e cada bucket ficou com um "template" gráfico pré-determinado - Falar sobre cada

grupo, o que significa e o que todos os gráficos partilham em comum em cada grupo.

4.2.3 Visualização escolhidas para os vários grupos

- Falar sobre os vários

4.3 Processamento dos dados

- Iniciar o tema de processamento de informação, converter informação desconhecida para um formato normalizado.

4.3.1 *Pipeline* de transformação dos dados

A primeira fase de normalização segue os seguintes passos:

- Extrai o nome do gráfico a partir da primeira linha de cada folha.
- Usa a segunda linha como cabeçalhos e normaliza os nomes (ex: substitui espaços por *underscores*, remove quebras de linha).
- Remove colunas irrelevantes com base numa lista de columnas que não tem representação.
- Normaliza dados nas células (como por exemplo: remove quebras de linha, espaços duplos).

A segunda fase vai depender do ficheiro e do gráfico que queremos apresentar, mas no geral,

(...) completar isto

- Falar como foi desenvolvida, e as várias fases de transformação A desenvolvida em *Python* com recurso ao Pandas.

4.3.2 Conversão e Exportação para CSV

- Falar sobre a transformação do pandas- Falar da exportação do CSV- Falar da gestão de ficheiros CSV e XLSX, e metodologias para não haver conflito entre ficheiros carregados

4.4 Desenvolvimento da Aplicação Web

Esta secção foca-se na estrutura e funcionamento da aplicação web — tanto o backend em Django como a interface construída com HTML, WebComponents e Flowbite.

4.4.1 Arquitetura da aplicação Django

- Descrever como está organizada a aplicação: os modelos principais (Quarter, ExcelFile, CSVFile), as relações entre eles e como são usados para garantir o isolamento por utilizador.
- Explicar o sistema de autenticação (Django Auth) e como se garantem as permissões e o acesso aos dados por utilizador.
- Referir também o uso do sistema de media (uploads) com UUIDs por quarter, e a criação automática das pastas.

4.4.2 Endpoints e API interna

Detalhar os principais endpoints utilizados: uploads, visualização de gráficos, listagem de ficheiros, etc.

- Mencionar a estrutura REST dos endpoints e como a interface os consome (por exemplo, o `/api/charts/chart_id > / < quarter > /`).
-
- Referir como se gere o estado do sistema com propriedades como *is_current*, *eoqueacontecequando*

4.4.3 Sistema de processamento assíncrono e normalização

- Falar sobre o `run_pipeline_for_sheet(...)` do `data_processing.py` e como isso se encaixa com o modelo Excel
-

- *Comentar o sistema de marcação dos ficheiros como processados e o uso de flags como processed.*

4.5 Interface Gráfica e Experiência do Utilizador (Frontend)

Aqui explicas como a interface foi pensada, as ferramentas utilizadas e como os gráficos são construídos com base nos dados enviados pelo backend.

4.5.1 Design System e Flowbite

Explicar por que escolheste Flowbite e como ele ajudou a construir uma UI consistente e reutilizável.

Mostrar exemplos de componentes reutilizados, como modais, botões, tabs, etc.

4.5.2 WebComponents e gráficos interativos

Falar sobre como encapsulaste a lógica dos gráficos usando WebComponents, para evitar conflitos de JS e garantir modularidade.

Descrever a comunicação entre os WebComponents e o backend via fetch, passando parâmetros (por exemplo, o quarter atual, ou filtros).

4.5.3 Gestão de Quarters e Uploads

Mostrar como funciona a criação de quarters e a navegação entre diferentes trimestres (com os botões e setas).

Explicar o fluxo de upload de ficheiros e como a interface valida o tipo de ficheiro, evita duplicações, e atualiza a visualização após carregamento.

Capítulo 5

Validação e Testes

Validação e testes aqui ...; pode precisar de referir o capítulo 3 ou alguma das suas secções, e.g., a secção 3.2 ...

Pode precisar de apresentar tabelas. Por exemplo, a tabela 5.1 apresenta os dados obtidos na experiência ...

c_1	c_2	c_3	$\sum_{i=1} c_i$
1	2	3	6
1.1	2.2	3.3	6.6

Tabela 5.1: Uma tabela

Para além de tabelas pode também precisar de apresentar figuras. Por exemplo, a figura 5.1 descreve ...

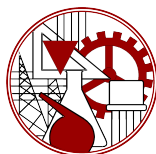


Figura 5.1: Uma figura

Atenção. Todas as tabelas e figuras, e.g., diagramas, imagens ilustrativas da aplicação em funcionamento, têm que ser devidamente enquadradas no texto antes de serem apresentadas e esse enquadramento inclui uma explicação da imagem apresentada e eventuais conclusões (interpretações) a tirar dessa imagem.

Capítulo 6

Conclusões e Trabalho Futuro

Conclusões e trabalho futuro aqui . . .

Quais as principais mensagens a transmitir ao leitor deste trabalho? O leitor está certamente interessado nos temas aqui abordados. Em geral procurará, neste projeto, pistas para algum outro objetivo. Assim, é muito importante que o leitor perceba rapidamente a relação entre este trabalho e o seu próprio (do leitor) objetivo.

Aqui é o local próprio para condensar a experiência adquirida neste projeto e apresentá-la a outros (futuros leitores).

O pressuposto é o de que este projeto é um “elemento vivo” que recorreu a outros elementos (cf., capítulo 2) para ser construído e que poderá servir de suporte à construção de futuros projetos.

Apêndice A

Sistemas de Apoio à Decisão

Um SAD é uma aplicação ou conjunto de ferramentas desenhadas para ajudar os utilizadores a tomar decisões mais informadas e fundamentadas, geralmente com base na análise de dados. Na prática, um SAD recolhe, organiza e processa dados, muitas vezes em tempo real ou a partir de ficheiros carregados pelo utilizador, e apresenta esses dados de forma visual (como gráficos ou dashboards), permitindo aplicar filtros e explorar cenários. Ou seja, não toma decisões por si só, mas fornece os elementos certos para que o utilizador possa decidir melhor, especialmente em contextos complexos ou com grande volume de informação.

Apêndice B

Casos de Utilização - UML

Apêndice C

Tabela de Requisitos funcionais

Apêndice D

Tabela de Requisitos não funcionais

Apêndice E

Classificação das folhas XLSX

Bibliografia