 **INSTITUTO POLITÉCNICO DE BEJA **

**Escola Superior de Tecnologia e Gestão**

**Licenciatura em Engenharia Informática**

**Sistemas de Informação**

**Trabalho Prático nº 1**

Elaborado por:

José Francisco Fernandes

Tierri Ferreira

Docentes:

Isabel Brito

13/11/2023

Índice

[1. Introdução 3](#_Toc137715407)

[2. Metodologia de trabalho 4](#_Toc137715408)

[3. Calendarização 5](#_Toc137715409)

[4. Desenvolvimento do projeto 7](#_Toc137715410)

[4.1. Recolha de informação 7](#_Toc137715411)

[U-Bike 7](#_Toc137715412)

[U-Bike IPLeiria 13](#_Toc137715413)

[U-Bike UPorto 14](#_Toc137715414)

[Lime 15](#_Toc137715415)

[5. Requisitos funcionais e não funcionais 17](#_Toc137715416)

[5.1. Requisitos funcionais 17](#_Toc137715417)

[5.2. Requisitos não funcionais 18](#_Toc137715418)

[6. Funcionalidades 19](#_Toc137715419)

[6.1. U-Bike 19](#_Toc137715420)

[6.2. U-Bike Management 20](#_Toc137715421)

[7. Utilizadores 21](#_Toc137715422)

[8. Diagrama de Casos de Uso 22](#_Toc137715423)

[8.1. U-Bike 22](#_Toc137715424)

[8.2. U-Bike Management 23](#_Toc137715425)

[9. Diagrama de Classes 24](#_Toc137715426)

[10. Diagramas de Sequência 25](#_Toc137715427)

[11. Conclusões e considerações finais 34](#_Toc137715428)

Figuras

Figura 1: Quadro de análise do projeto. 8

Figura 2: Quadro de desenho do projeto. 8

Figura 3: Quadro de gestão do projeto. 9

Figura 4: Ecrã de login – U-Bike. 10

Figura 5: Ecrã principal – U-Bike. 11

Figura 6: Ecrã de código QR U-Bike. . 12

Figura 7: Ecrã de nº da bicicleta. 12

Figura 8: Ecrã de menu - U-Bike. 12

Figura 9: Ecrã “Minhas Viagens” - U-Bike. 13

Figura 10 – Ecrã de “Métodos de Pagamento” – U-Bike. 14

Figura 11 – Ecrã das Definições - U-Bike. 14

Figura 12: Ecrã de Perfil – U-Bike. 15

Figura 13: Ecrã principal – U-Bike IPLeiria. 16

Figura 14: Tela de viagem U-Bike IPLeiria. 16

Figura 15: Ecrã Principal U-Bike UPorto. 17

Figura 16: Ecrã Principal – Lime. 18

Figura 17: Diagrama de casos de uso – U-Bike. 24

Figura 18: Diagrama de casos de uso – U-Bike Management. 25

Figura 19: Diagrama de classes da U-Bike. 26

Figura 20: Diagrama de sequência “LoginSequence”. 27

Figura 21: Diagrama de sequência “RentElectricBike”. 28

Figura 22: Diagrama de sequência "RentNonElectricBike”. 28

Figura 23: Diagrama de sequência " FindOwnedBikeSequence”. 29

Figura 24: Diagrama de sequência " ViewTripsSequence”. 30

Figura 25: Diagrama de sequência " SavingsSequence”. 30

Figura 26: Diagrama de sequência "TripStarterSequence”. 30

Figura 27: Diagrama de sequência "FinishTripSequence”. 31

Figura 28: Diagrama de sequência "ProfileEditSequence”. 31

Figura 29: Diagrama de sequência "ViewAllBikesSequence”. 32

Figura 30: Diagrama de sequência "ListAllInstitutionBikesSequence”. 32

Figura 31: Diagrama de sequência "InstitutionGranterSequence”. 32

Figura 32: Diagrama de sequência "BikeInformationSequence”. 33

Figura 33: Diagrama de sequência "RegisterBikeSequence”. 33

Figura 34: Diagrama de sequência "CreateManagerSequence”. 34

Figura 35: Diagrama de sequência "ListInstitutionsSequence”. 34

Figura 36: Diagrama de sequência "ViewAppAsAnotherRoleSequence”. 35

Figura 37: Diagrama de sequência "ViewInstitutionStatsSequence”. 35

Figura 38: Diagrama de sequência "ControlBikeSequence”. 36

# Introdução

Neste trabalho temos como objetivo extrair, transformar e carregar dados sobre um tema escolhido por nós para fazer uma análise OLAP aos mesmos. Será também feito um modelo de data warehouse para estes dados.

O tema escolhido está relacionado com dados de vídeos nas tendências do YouTube, filtrados com base nos canais com mais subscritores.

Para fazer o ETL, foi feito um *script* em Python para filtrar os dados com base num dataset dos canais mais subscritos no YouTube para reduzir a quantidade de informação e manter os dados estatísticos nos canais maiores, para evitar excesso de dados que apenas existem devido a uma certa tendência do momento.

O presente relatório encontra-se organizado na seguinte forma:

# Datasets obtidos

Os datasets foram obtidos a partir do Kaggle [1] [2] e estão relacionados com dados de vídeos nas tendências no YouTube e os canais mais subscritos do YouTube, respetivamente.

Ambos estes datasets estão formatados em CSV, mas o primeiro dataset (com os vídeos nas tendências) também tem um ficheiro JSON de categorias à parte com os dados das categorias dos vídeos com o seu nome, em vez de um ID numérico. Este ficheiro foi também usado para fazer a análise OLAP.

Adicionalmente, o dataset de vídeos nas tendências traz diversos ficheiros CSV com países diferentes, uma vez que as tendências variam consoante a localização geográfica. Neste trabalho, escolhemos os EUA.

O segundo dataset (canais mais subscritos no YouTube) foi usado para manter a análise dos dados mais abrangente, ou seja, uma vez que estamos a trabalhar ao nível estratégico, não queremos que tendências atuais influenciem dados. Por exemplo, se houver uma tendência que dura um mês e acontece num canal pouco famoso, este(s) vídeo(s) influenciarão os resultados da análise OLAP e os dados poderão ser menos realistas.

Para tornar esta filtragem possível, usámos um *script* Python que filtrava os vídeos nas tendências pelos canais mais subscritos. Não foi possível fazer esta filtragem sem este dataset devido ao facto de dados de subscritores apenas existirem em datasets de canais, e não em datasets de vídeos. Na secção 3 será possível visualizar este processo.

# Processo ETL

Para tornar o processo ETL possível, foram usadas duas tecnologias principais. Inicialmente, para o primeiro *extract*, foi usado o Python com a biblioteca *csv*. Na Figura 1 é possível observar este *script* simples.

A screen shot of a computer program

Description automatically generated

Figura 1 – *Script* em Python para o primeiro *extract* e *transform*.

Na função definida na linha 4, *parseCsvToArray*, é feita a leitura do ficheiro CSV e já é feita uma transformação muito simples, que é a eliminação de dados corruptos/incorretos na sintaxe CSV. Isto foi observável no maior dataset, constituído pelos vídeos nas tendências.

Nas linhas 26 a 31, apenas foi feita uma verificação do espaço de tempo dos dados, para que se verificasse se este dataset tinha um intervalo de tempo suficientemente grande.

Seguidamente é feita a filtragem a partir do segundo dataset, que é lido da mesma forma que o primeiro. Estes dados são iterados (os canais) e faz-se assim uma filtragem dos vídeos, de modo que apenas vídeos do canal iterado sejam adicionados ao array final.

Este array final, por fim, é enviado para um ficheiro JSON de saída, que depois será interpretado pelo PowerBI.

Para a segunda e última parte do *extract* e *transform*, foi usado o PowerBI para alterações mais simples.

A primeira coisa que foi feita foi a abertura de um novo projeto no PowerBI e o carregamento do ficheiro JSON *out.json*. Isto abrirá o Power Query, como se observa na Figura 2.

A screenshot of a computer

Description automatically generated

Figura 2 – Power Query, aberto ao carregar o ficheiro out.json no PowerBI, com os vídeos filtrados no *script* Python.

Por agora, não será necessário editar nada. Simplesmente fecharemos e carregaremos estes dados para o PowerBI.

Uma vez que os dados são pouco legíveis devido aos seus nomes, mudaremos os mesmos, como se vê nas Figuras 3 e 4.

A screenshot of a computer screen

Description automatically generated

Figura 3 – Nomes anteriores nos dados recentemente carregados.

A screenshot of a computer screen

Description automatically generated

Figura 4 – Nomes atualizados.

Agora, uma vez que na análise vamos precisar de verificar horas de publicação, extraímos as horas de “Published At” para uma nova coluna. Na Figura 5 observamos o código em M para fazer esta coluna.



Figura 5 – Criação da coluna “Published Hour”.

Foi necessária a criação desta coluna uma vez que o PowerBI, por alguma razão que não foi possível verificar, não permite a filtragem de conteúdo por hora, apenas até ao dia, mesmo que o tipo de dados seja *datetime*.

De seguida, para tornar a leitura mais fácil, vamos alterar tipos booleanos em texto para que seja mais fácil de interpretar a informação analisada. Para fazê-lo, abrimos o Power Query e alteramos os tipos das colunas “Comments Disabled" e “Ratings Disabled”. O tipo é alterado conforme se apresenta na Figura 6.

A screenshot of a computer

Description automatically generated

Figura 6 – Alteração do tipo de dados de colunas booleanas.

Após esta alteração, vamos substituir todos os valores “true” e “false” em “sim” e “não”, respetivamente. Na Figura 7 podemos observar como o mesmo se faz.

A screenshot of a computer

Description automatically generated

Figura 7 – Substituição de valores booleanos em texto português.

Após esta substituição, podemos novamente carregar os dados do Power Query para o PowerBI.

Agora que terminámos as alterações nos vídeos, vamos carregar as categorias, que se encontram em JSON. Isto novamente abrirá o Power Query.

Desta vez, iremos já fazer a transformação destes dados. Iremos eliminar colunas/atributos redundantes e desnecessárias do ficheiro, uma vez que apenas queremos, principalmente, os nomes das categorias, que não se encontram no dataset principal, para depois fazer a análise com categorias legíveis.

Eliminaremos as seguintes colunas/atributos:

* kind (atributo)
* etag (atributo)
* items.kind (coluna)
* items.snippet.assignable (coluna)
* items.snippet.channelId (coluna)
* items.etag (coluna)

Os únicos dados que restam desta transformação são os seguintes (Figura 8):

A screenshot of a computer

Description automatically generated

Figura 8 – Categorias de vídeos.

Os nomes das colunas e do ficheiro foram alterados, tal como no anterior. Na Figura 9 observam-se os novos nomes.

A screenshot of a survey

Description automatically generated

Figura 9 – Novos nomes das colunas de Category.

Por fim, foi feita uma relação entre Trending e Category para que o ID das categorias fosse reconhecido como chave estrangeira e o nome das categorias seja diretamente associado durante a fase de análise.

Na Figura 10 podemos observar esta relação.

A screenshot of a computer

Description automatically generated

Figura 10 – Relação entre Trending e Category.

Para criar esta relação, simplesmente arrasta-se o atributo “Category ID” em Trending para “ID” em Category e o PowerBI automaticamente cria a relação.

# Conclusões e considerações finais

Podemos concluir que este projeto foi muito enriquecedor para os nossos conhecimentos profissionais e académicos, neste caso na área de Engenharia de Software.

Este projeto foi trabalhoso, com múltiplas fases por desenvolver, e muita pesquisa a fazer. No entanto, acreditamos que o projeto ficou bem planeado e está pronto a implementar, agora que as fases de análise e de desenho estão prontas.

Algumas dificuldades que sentimos foi encontrar informação para o projeto, uma vez que apesar de a U-Bike estar implementada em três instituições de ensino superior, não está funcional em nenhuma delas (ou pelo menos sem acesso público, como na U-Bike U.Porto). No entanto, após alguma pesquisa intensiva na matéria, ficámos a perceber como teríamos de proceder com este projeto na fase de análise.

# Bibliografia

|  |  |
| --- | --- |
| [1] | “YouTube Trending Video Dataset (updated daily),” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset?select=US\_youtube\_trending\_data.csv%29. [Acedido em 23 10 2023]. |
| [2] | “Most Subscribed YouTube Channels,” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/surajjha101/top-youtube-channels-data. [Acedido em 7 11 2023]. |