

Relatório de Projeto Integrador
Análise de Padrões de Produtividade

2022/2023

Licenciatura em Engenharia Informática e Computação

Diogo Alexandre da Costa Melo Moreira da Fonte - up202004175@edu.fe.up.pt

Índice

1. Introdução	2
2. Plano de Trabalho	3
3. Desenvolvimento e Produto	4
4. Conclusões	15
5. Referências	16
6. Anexos	17

1. Introdução

O meu projeto integrador (PI) foi realizado na modalidade de estágio curricular e decorreu em acordo com a Armis - Sistemas de Informação, Lda. O horário de estágio principal foi sexta-feira entre as 11h30 e as 19h30, resultando em 8 horas de trabalho presencial nas instalações do Porto da empresa. A restante hora e meia prevista foi concluída de forma remota. O Professor Ademar Aguiar foi o meu orientador por parte da FEUP e esteve sempre disponível para todas as questões/problemas que tive. Durante este estágio fui inserido na área de Data do departamento Enterprise Solutions da Armis e fui acompanhado pelo Engenheiro Milton Nunes. Contudo também existiu muito apoio ao longo de todas as atividades por parte do Engenheiro Sávio Castro e do Engenheiro Márcio Ribeiro.

O objetivo principal deste trabalho é analisar dados de histórico de vários projetos e identificar padrões. Estes padrões devem permitir:

- identificar desvios ou comportamentos negativos na produtividade dos colaboradores, para assim ser possível definir estratégias para os corrigir ou antecipar;
- identificar lacunas nas ferramentas internas que devam ser implementadas para melhorar o controlo e a antecipação de problemas.

O resultado esperado é a apresentação de um dashboard que possibilite a análise e interpretação dos dados fornecidos com informações de vários projetos realizados, no âmbito da produtividade. A ideia é apresentar vários elementos visuais para a análise de vários parâmetros em termos de produtividade, tais como avanço técnico dos projetos, margem horária (horas previstas vs horas realizadas) e classificações de produtividade, tendo em conta intervalos (desta forma não são emitidas opiniões e apenas são apresentados factos). O previsto será ter uma página global para os projetos e uma para as tarefas, tal como páginas mais específicas, para ser possível analisar com mais pormenor um certo projeto e as suas tarefas.

Criei um repositório no GitHub para poder ter disponível todos os ficheiros e anexos referentes a este projeto integrador. O link do repositório é o seguinte: <https://github.com/diogofonte/feup-pi>.

2. Plano de Trabalho

Para o desenvolvimento do projeto, as tarefas previstas foram distribuídas em 6 sprints.

1. Sprint 1

- Pesquisa sobre as tecnologias a utilizar, nomeadamente Azure Sql Database, Azure Data Factory, Azure Databricks e PowerBI;
- Análise da base de dados original.

2. Sprint 2

- Criação da base de dados Staging e da base de dados DW;
- Definição do modelo de dados da DW.

3. Sprint 3

- Criar um Data Factory Pipeline para data ingestion.

4. Sprint 4

- Criar um Python Notebook para a transformação dos dados;
- Carregar os dados para a base de dados DW.

5. Sprint 5

- Criar um relatório em Power BI para apresentar os resultados em forma de dashboard.

6. Sprint 6

- Testes da Data Ingestion e da transformação dos dados;
- Testes do report criado em Power BI.

Em todos os sprints existiu uma tarefa extra direcionada à escrita deste relatório, para o fazer de forma regular e incremental. Desta forma consegui documentar as tarefas que fui executando ao longo do estágio com maior pormenor, devido ao intervalo entre a data de execução e a data de escrita ser curto.

Todas as semanas existiram duas reuniões, sendo uma às 17h30 de quarta-feira e a outra às 11h30 de sexta-feira (dia de estágio na empresa). De forma a controlar o desenvolvimento das tarefas e o avanço dos sprints, utilizamos o Azure DevOps, onde foram inseridos os sprints e as respetivas tarefas a executar.

3. Desenvolvimento e Produto

Sprint 1 - Pesquisa sobre as tecnologias e análise da base de dados

Neste primeiro sprint, comecei por receber uma abordagem high-level do que ia abordar neste projeto, tal como a ordem dos processos necessários para chegar ao resultado pretendido. Com o Azure DevOps foi possível acompanhar o progresso e realizar alguns comentários através de tarefas inseridas em sprints. A divisão por sprints foi feita como referi anteriormente.

Em termos da aprendizagem das tecnologias necessárias comecei por aprender sobre Azure Data Factory para depois saber como criar o pipeline necessário para copiar os dados de umas tabelas para outras. Também pesquisei sobre Azure Databricks, que é a ferramenta que permitiu transformar os dados originais para suportarem o esquema final. Sobre Power BI, pesquisei sobre a forma como se apresenta os dados e vi alguns exemplos de relatórios para ter uma ideia do que é possível apresentar.

Quando me disponibilizaram a base de dados original (dbo schema) num ficheiro bacpac, criei uma Azure SQL Database no meu Azure SQL Server. Estas tabelas ficaram guardadas no schema dbo, pois utilizei apenas uma base de dados e coloquei nesta os 3 schemas que necessitei (sbo, stg e dw). Desta forma foi possível poupar recursos em base de dados. DW significa Data Warehouse que é um sistema

empresarial utilizado para a análise e geração de relatórios de dados estruturados e semiestruturados de várias fontes, ou seja, o destino final e onde os dados são armazenados para posterior análise.

Após análise da base de dados e discussão com os meus orientadores, referenciamos 5 tabelas importantes que me facultam os dados necessários para o meu projeto:

- dbo.EXT_TBL_PROJETOS - Lista de projetos;
- dbo.EXT_TBL_ORCAMENTO - Lista de tarefas orçamentadas pelo departamento de recursos humanos;
- dbo.EXT_TBL_HORASPREVISTAS - Lista das tarefas planeadas para cada projeto atribuídas a funcionários;
- dbo.EXT_TBL_IMPUTACAO_DETALHE - Lista de imputações, cada uma associada a uma certa tarefa planeada;
- dbo.EXT_TBL_HISTORICO_AVANCOS - Lista de avanços técnicos de cada projeto.

Com os dados disponíveis, abordei o tema produtividade incidindo mais no parâmetro das horas, já que uma das melhores formas de medir a produtividade é a verificação das horas que foram necessárias para realizar uma certa tarefa. Analisar a evolução do avanço do projeto ao longo dos meses e atribuir classificações/intervalos de produtividade (em percentagem) são outras verificações que executei para dar uma análise mais high-level do que aconteceu em cada tarefa e em cada projeto.

Sprint 2 - Base de dados Staging, base de dados DW e modelo DW

Este sprint foi dividido em duas tarefas principais, tal como mencionado anteriormente. Para a execução da primeira tarefa criei dois schemas dentro da minha base de dados SQL do Azure. Criei um script SQL que cria as tabelas selecionadas no sprint anterior para o schema staging, que está disponível no link: https://github.com/diogofonte/feup-pi/blob/main/stg/create_stg_tables_script.sql.

Neste schema staging tenho a informação original, mas tenho a possibilidade de fazer alterações consoante as minhas necessidades.

Para conseguir criar o modelo dw, que me dá suporte aos dados que são inseridos no Power BI, tive de pensar na estrutura que necessitava. Após muitos rascunhos e trocas de ideias, cheguei a um modelo final. Fiz um diagrama para representar o modelo numa forma conceptual e posteriormente criei o script SQL. Na figura 1 é possível visualizar o modelo com as tabelas e os atributos necessários para suportar o relatório Power BI:

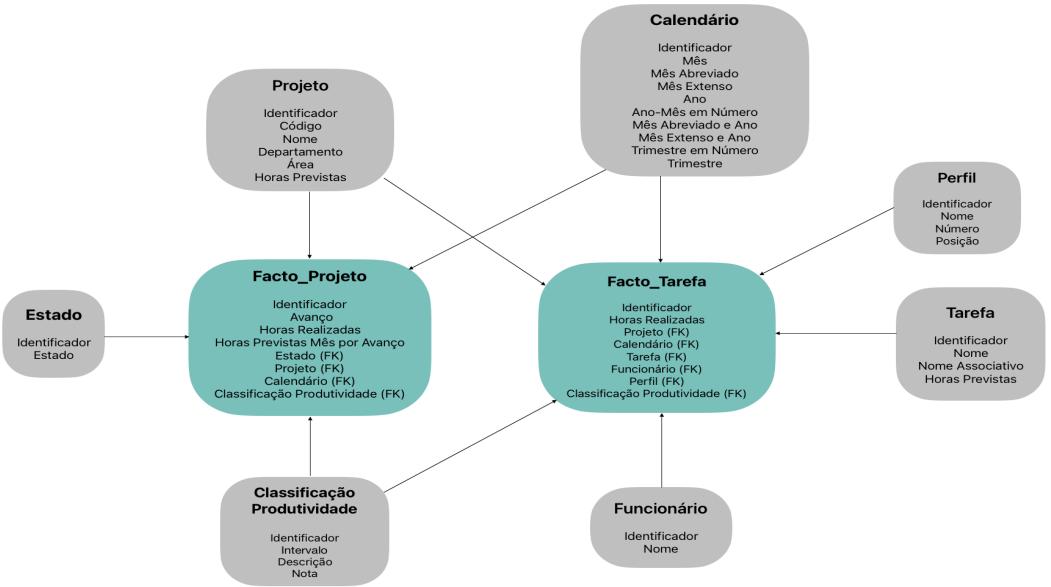


Figura 1 - diagrama do modelo de dados da dw

Para melhor visualização, este diagrama encontra-se no seguinte link: https://github.com/diogofonte/feup-pi/blob/main/dw/modelo_dw.pdf. O script SQL de criação das tabelas referidas no diagrama acima apresentado está no link: https://github.com/diogofonte/feup-pi/blob/main/dw/create_dw_tables_script.sql.

Primeiramente, tenho as tabelas que representam dimensões a cinzento e as tabelas com os dados para analisar (factos) a verde-claro. As tabelas facto servem para suportar os dados que pretendo analisar e as tabelas dimensão guardam informações que identificam elementos das tabelas facto. Por este motivo, nas tabelas facto possuo ID's para as tabelas dimensão necessárias e os restantes campos de cálculo / informação. Nas tabelas dimensão tenho vários atributos que me ajudam a identificar os dados.

Um projeto é composto por várias tarefas, que são anteriormente estipuladas e orçamentadas. Decidi ter duas tabelas distintas para poder ter a informação global e final do projeto como um todo, mas também possuir informação individualizada de cada uma das tarefas do projeto. Desta forma é possível verificar se o projeto foi produtivo e/ou qual ou quais foram as tarefas mais ou menos produtivas.

A menor granularidade das tabelas facto, em termos temporais, é o mês, pois como se pode ver pela tabela da dimensão Calendário, apenas estou a fazer uma análise mensal em vez de diária.

Falando um pouco sobre a classificação de produtividade que atribuí às tarefas e aos projetos, esta é calculada de forma global, ou seja, verifica-se as horas que foram necessárias para a conclusão da tarefa/projeto e compara-se com as horas que eram previstas. A fórmula utilizada para ter o resultado em percentagem foi a seguinte: $HorasRealizadas / HorasPrevistas * 100$

Os intervalos que utilizei para atribuir uma nota informativa sobre a classificação de produtividade calculada foram os seguintes:

- [0%, 100%] - Menos do Previsto - Nota 5
- [100%, 100%] - Como Previsto - Nota 4
-]100%, 125%] - Até 25% Horas Extra Consumidas - Nota 3
-]125%, 150%] - Até 50% Horas Extra Consumidas - Nota 2
-]150%, 175%] - Até 75% Horas Extra Consumidas - Nota 1
-]175%, +infinito] - Mais de 75% Horas Extra Consumidas - Nota 0

Sprint 3 - Data Ingestion

Posteriormente à criação das tabelas no schema Staging (stg), utilizei as funcionalidades do Azure Data Factory para copiar os dados presentes nas tabelas referidas da base de dados original (dbo) para estas que foram criadas.

Criei um pipeline para copiar os dados do schema dbo (as *is*, ou seja, sem nenhuma alteração) para o schema stg. Utilizei uma função de pesquisa (*lookup*) que usa uma consulta para obter os nomes das tabelas que referi anteriormente como importantes. Com o output deste lookup, defini uma função *for each* que executa uma atividade *copy data*, para copiar os dados de cada tabela do schema dbo para a correspondente no schema stg. A query que introduzi para selecionar as tabelas que pretendia copiar foi a seguinte:

```
SELECT TABLE_SCHEMA, TABLE_NAME FROM information_schema.tables WHERE TABLE_TYPE =  
'BASE TABLE' AND TABLE_SCHEMA = 'dbo' AND TABLE_NAME in  
('TBL_PROJETOS', 'TBL_HORASPREVISTAS', 'TBL_ORCAMENTO', 'TBL_IMPUTACAO_DETALHE', 'TBL_H  
ISTORICO_AVANCOS')
```

Nas figuras 2, 3 e 4 é possível visualizar alguns dos componentes necessários para a migração dos dados do schema dbo para o schema stg.

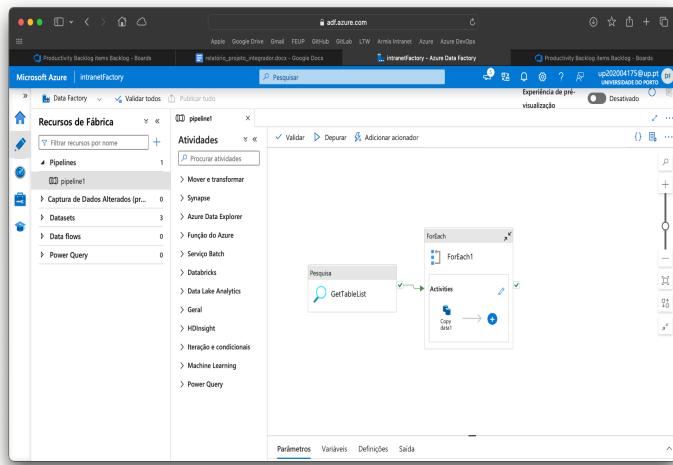


Figura 2 - Data Factory Pipeline

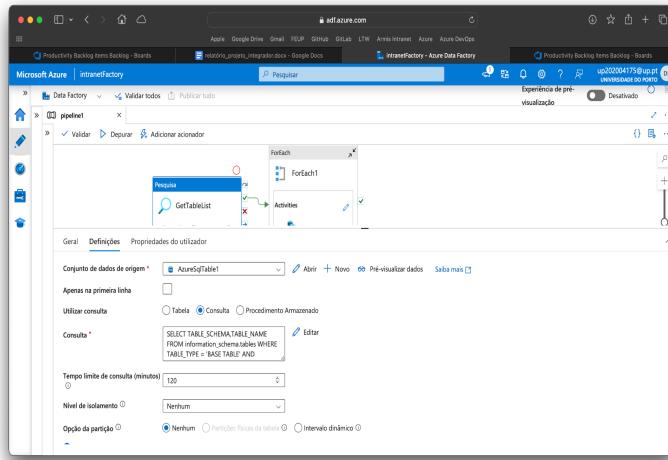


Figura 3 - Lookup function

Figura 4 - For each activity

Estas imagens também podem ser encontradas no repositório git, na pasta:
https://github.com/diogofonte/feup-pi/tree/main/screenshots/data_factory_pipeline

Sprint 4 - Transformação de Dados e Carregamento dos Dados para a DW

O objetivo deste sprint é transformar os dados que estão presentes nas tabelas do schema stg e introduzi-los nas tabelas de dimensão e de facto do schema dw. Numa primeira abordagem, utilizei ciclos for para conseguir percorrer as dataframes (que continham os resultados das queries executadas) e fazer todas as alterações nos campos que pretendia. Ao carregar os dados, percebi que os tempos de execução não eram suportáveis. Em conversa com o meu orientador, percebi que não é uma boa abordagem utilizar ciclos em transformação de dados, pois são processos bastante lentos, mesmo numa base de dados com pouca carga. Esta primeira abordagem está presente neste notebook: https://github.com/diogofonte/feup-pi/blob/main/data_transformation/data_transformation_notebook_with_fors.ipynb.

Na segunda abordagem, que é a forma optimal de se proceder quando se trabalha com dados, utilizei apenas operações das próprias spark dataframes, tais como, join, withColumn e select, por exemplo.

Após a conexão à base de dados, comecei a transformação das tabelas dimensão. Iniciei pela tabela que representa o Calendário e esta vai ter como menos granularidade o mês, por isso inseri todos os meses necessários entre a data mais antiga e a mais recente (estas datas foram verificadas através de querys). Decidi ter as seguintes colunas na tabela:

- Mes (ex.: 1)
- Mes_Abrev (ex.: Jan)
- Mes_Extenso (ex.: Janeiro)
- Ano (ex.: 2020)
- Ano_Mes_Num (ex.: 2020-01)
- Mes_Abrev_E_Ano (ex.: Jan 2020)
- Mex_Extenso_E_Ano (ex.: Janeiro 2020)
- Trimestre_Num (ex.: 1)
- Trimestre (ex.: 2020 Trimestre 1)

Quanto às tabelas de Estado (do projeto), Perfil, Funcionário e Tarefa, apenas tive de executar queries simples para ir buscar os dados necessários às tabelas do schema stg em que estão disponíveis e fazer algumas formatações para preencher alguns campos extra. Estas quatro tabelas têm o seguinte formato:

- **Estado** - Nome
- **Perfil** - Nome, Número e Posição
- **Funcionário** - Nome
- **Tarefa** - Nome, Número de Horas e Descritivo Pormenorizado

Na tabela da Classificação de Produtividade apenas tive de inserir os tuplos que referi na secção anterior, com os respetivos atributos (Intervalo, Descrição e Nota).

Para a dimensão Projeto, foi necessário fazer um join entre as tabelas TBL_PROJETOS e TBL_HORASPREVISTAS para conseguir ter acesso aos detalhes e às horas totais previstas de cada um dos projetos em questão. Esta tabela ficou com os seguintes atributos: Código Projeto, Departamento, Área e Horas Previstas.

Na transformação dos dados para as tabelas facto (Projeto e Tarefa), executei duas queries iniciais em ambos e estas tinham como origem a tabela TBL_IMPUTACOES, já que é nesta que consigo recolher as imputações de cada funcionário em cada tarefa por mês. Para a facto Tarefa fiz um *group by* pelas colunas CódigoProjeto, Username, Ano, Mes e FK_TarefaID, para conseguir associar o projeto, o funcionário e por sua vez o seu perfil, o mês e ano em que foram realizadas as horas e a que tarefa pertencem. Já para a facto Projeto, retirei as três últimas colunas do *group by*, para desta forma ter a junção das horas realizadas no projeto e mês em questão. Em termos de processo, utilizei operações join e withColumn (tal como nas tabelas dimensão) com parâmetros when que verificam os valores para associar aos ID's corretos das tabelas que representam dimensões. Quando algum valor era inválido, o ID colocado apontava para o tuplo inválido da respetiva dimensão.

Como o python notebook é bastante extenso decidi não o colocar no relatório, mas está presente neste link: https://github.com/diogofonte/feup-pi/blob/main/data_transformation/data_transformation.ipynb.

Sprint 5 - Relatório em Power BI

Durante a elaboração do relatório em Power BI, detectei dois erros na transformação de dados. O primeiro foi no cálculo da classificação de produtividade, pois estava a utilizar o número de horas realizadas mensalmente, em vez de utilizar o número de horas totais, porque a classificação é um parâmetro global e final. Este erro provocava mais do que uma classificação por tarefa, já que as horas realizadas variam entre meses. O segundo erro estava presente na ligação das imputações ao perfil de funcionário correspondente, pois estava a associar o perfil a partir do código do projeto, do nome da tarefa e do nome do funcionário. A forma correta é utilizar o id presente na tabela TBL_HORASPREVISTAS, que corresponde ao id da tarefa e com o atributo OrcamentoID é possível aceder à tabela TBL_ORCAMENTO para recolher o perfil associado à tarefa.

Em termos de inicialização, comecei por conectar à base de dados para poder carregar as tabelas do *schema dw*. Esta parte foi executada na secção de transformação do Power BI Desktop (que é a ferramenta necessária para a criação deste tipo de relatórios). Criei 3 pastas, que são Parameters, Facts e Dimensions. A primeira (Parameters) serviu para guardar o nome do Azure SQL Server e o nome da Azure SQL Database na forma de parâmetros para poderem ser utilizados nos carregamentos das tabelas necessárias. As duas pastas restantes (Facts e Dimensions) são *self-explanatory*. Existe um ponto que necessita de atenção e este tem a ver com as queries de seleção. Para carregar as tabelas foi preciso especificar as colunas que pretendia, porque, num contexto de trabalho diferente, poderiam ser adicionadas colunas às tabelas iniciais e com um select * todas iriam ser carregadas para o *report*. Desta forma é garantido que não são carregadas colunas desnecessárias.

O passo seguinte é a organização do modelo de dados e a certificação de que todas as relações estão bem identificadas. Na secção “Vista do Modelo” conseguimos ter acesso às tabelas e às relações existentes na forma de um diagrama. Após organização e colocação de algumas relações, o resultado obtido foi o presente na figura 5.

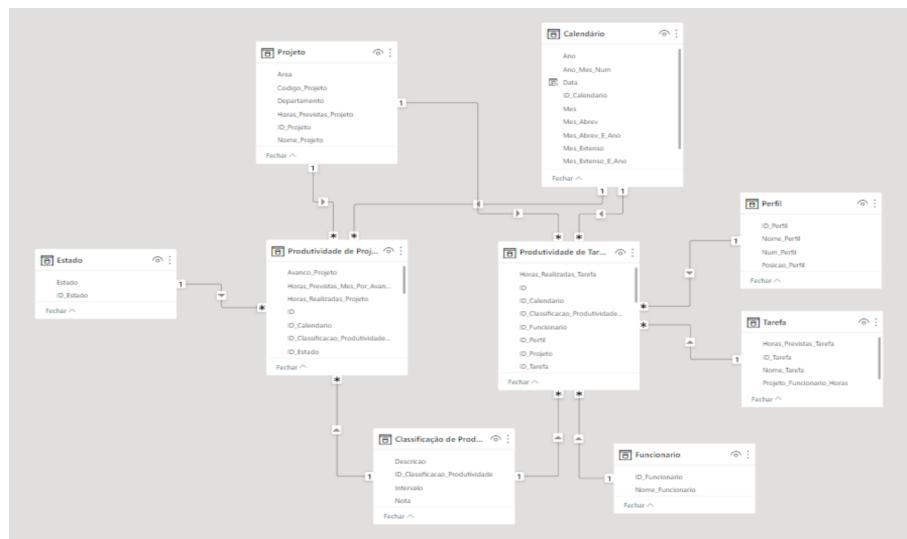


Figura 5 - Power BI Data Model

Após a verificação do modelo de dados, avancei para a remoção de somatórios e outras funções automáticas que o Power BI coloca automaticamente em campos numéricos. Também alterei alguns nomes para serem mais identificativos, como por exemplo, a tabela FACTO_PROJETO ficou com o nome Produtividade de Projetos.

A etapa seguinte foi decidir as páginas que pretendia apresentar e optei pela seguinte estrutura:

- Perspetiva Global de Projetos
- Relação de Projetos
- Análise por Projeto
- Perspetiva Global de Tarefas
- Relação de Tarefas
- Análise por Tarefa
- Imputações por Tarefa e Funcionário

Na primeira página temos acesso a dados globais dos projetos e podemos filtrar estes resultados com seleção de departamentos, áreas, estados de projeto e classificações de produtividade. Podemos fazer um drill-through (botão direito do rato no elemento, selecionar “drill-through” ou “pormenorizar” e escolher a página) em algum cartão ou elemento de um gráfico e somos encaminhados para a página Relação de Projetos, onde é possível visualizar uma tabela com as informações individualizadas dos projetos que estavam inseridos na seleção. Dentro desta página Relação de Projetos, podemos selecionar um projeto e fazer drill-through para a página Análise por Projeto onde temos a possibilidade de analisar as características do projeto ao longo da sua duração. Ainda na Análise por Projeto temos presente uma tabela com as tarefas associadas ao projeto e é possível fazer drill-through para a Análise por Tarefa.

A página Perspetiva Global de Tarefas tem o mesmo conceito da Perspetiva Global de Projetos, mas não possui o filtro do Estado do Projeto, pois este não está presente na tabela FACTO_TAREFA. Tal como para os projetos, é possível fazer um drill-through em algum elemento visual para ser redirecionado para a Relação de Tarefas, em que são apresentadas informações individuais de cada tarefa. Com um drill-through para a Análise por Tarefa numa tarefa, conseguimos analisar as suas características, tendo ainda a possibilidade de selecionar funcionários (que trabalharam na tarefa) e executar um drill-through para a página Imputações por Tarefa e Funcionário, onde conseguimos verificar em que meses e quantas horas por mês foram imputadas por cada funcionário.

Para elementos visuais com informações mais detalhadas, tive de criar métricas, que consistem em cálculos e/ou ligações de tabelas para conseguir ter acesso aos dados que pretendo. Para isto, tive de utilizar DAX (Data Analysis Expressions), que é uma biblioteca de funções e operadores que podem ser combinados para criar expressões e fórmulas de cálculo. Um exemplo de uma das métricas é a Média das Classificações de Produtividade de Projetos. Para esta foi necessário o seguinte código presente na figura 6.

```

Média das Classificações (Projeto) =
VAR _table = SUMMARIZE(
    CALCULATETABLE('Produtividade de Projetos',Projeto[ID_Projeto] <> -1, 'Classificação de
Produtividade'[ID_Classificacao_Produtividade] <> -1),
    'Calendário'[Ano_Mes_Num],
    Projeto[ID_Projeto],
    "c1",CALCULATE(MAX('Classificação de Produtividade'[Nota]),RELATEDTABLE('Produtividade de
Projetos'))
)

VAR _AVG = DIVIDE(
    SUMX( _table, [c1]),
    COUNTROWS(_table)
)
RETURN _AVG

```

Figura 6 – Exemplo Métrica

Coloquei todas as métricas de cada Facto numa pasta, para ter tudo organizado e conseguir distinguir claramente métricas de colunas da tabela. A título de exemplo é possível visualizar a primeira página na figura 7, que é a responsável por permitir uma análise geral de todos os projetos. Em anexo, encontram-se todas as páginas do relatório em maior ponto, contudo cada uma possui um link que permite aceder à imagem no repositório. O relatório Power BI final encontra-se no seguinte link: <https://github.com/diogofonte/feup-pi/blob/main/power%20bi/power%20bi%20report.pbix>



Figura 7 – Primeira Página do Relatório Power BI

Sprint 6 - Testes

Tal como referi no início da secção anterior, detetei dois erros na transformação de dados, que foram logo corrigidos.

Para testar a data ingestion, criei um Python Notebook no Azure Databricks, onde, para cada tabela, criava duas dataframes, uma com a tabela do schema dbo e outra com a tabela do schema stg. Ao comparar as duas dataframes consigo verificar se o conteúdo é o mesmo, tal como pretendido. O notebook está presente neste link: https://github.com/diogofonte/feup-pi/blob/main/tests/data%20ingestion/data_ingestion_test.ipynb

No caso da data transformation, foquei-me num único projeto, para conseguir perceber se as transformações foram executadas como pretendido. O projeto que selecionei foi o AGEAS.2017.190. Comparei os dados presentes nas tabelas originais e os presentes nas tabelas dimensão e facto. No repositório tenho presente vários screenshots para ser possível comparar as várias tabelas de forma mais visual, contudo as queries foram inseridas num Python Notebook onde também comparo as tabelas. Na figura 8 é possível visualizar a tabela FACTO_PROJETO à esquerda e a junção das tabelas do schema stg necessárias para obter a informação pretendida. O Notebook e os screenshots dos testes encontram-se na seguinte pasta do repositório: https://github.com/diogofonte/feup-pi/tree/main/tests/data_transformation

The screenshot displays two side-by-side SSMS windows. The left window shows the original data from the 'intranetupserver.database.windows.net.intranet14' database, specifically the 'FACTO_PROJETO' table, with a result set of 17 rows. The right window shows the transformed data from the 'intranet14' database, also from the 'FACTO_PROJETO' table, with a result set of 17 rows. Both windows include their respective execution plans at the top.

Código_Projeto	ID_Caixa	Hora_Ano	Avançado	Horas_Produção	Horas_Realizadas
1	AGEAS.2017.190	201711	96,00	NULL	2872,00
2	AGEAS.2017.190	201712	52,00	4,00	82,000000
3	AGEAS.2017.190	201801	46,00	8,00	165,760000
4	AGEAS.2017.190	201802	312,00	25,00	518,000000
5	AGEAS.2017.190	201803	212,00	25,00	518,000000
6	AGEAS.2017.190	201804	226,00	40,00	626,000000
7	AGEAS.2017.190	201805	237,00	50,00	1836,000000
8	AGEAS.2017.190	201806	386,00	70,00	1450,000000
9	AGEAS.2017.190	201807	273,00	65,00	1761,000000
10	AGEAS.2017.190	201808	85,00	69,00	1644,000000
11	AGEAS.2017.190	201809	22,00	89,00	1844,000000
12	AGEAS.2017.190	201810	64,00	89,00	1844,000000
13	AGEAS.2017.190	201811	68,00	89,00	1844,000000
14	AGEAS.2017.190	201812	81,00	89,00	1844,000000
15	AGEAS.2017.190	201901	24,00	89,00	1844,000000
16	AGEAS.2017.190	201903	3,00	89,00	1844,000000
17	AGEAS.2017.190	201904	2,00	89,00	1844,000000

Código_Projeto	Ano	Mes	Hora_Ano	Avançado	Estado
1	AGEAS.2017.190	2017	11	96,00	NULL
2	AGEAS.2017.190	2017	12	52,00	4,00
3	AGEAS.2017.190	2018	1	46,00	8,00
4	AGEAS.2017.190	2018	2	312,00	25,00
5	AGEAS.2017.190	2018	3	212,00	NULL
6	AGEAS.2017.190	2018	4	226,00	40,00
7	AGEAS.2017.190	2018	5	237,00	50,00
8	AGEAS.2017.190	2018	6	306,00	70,00
9	AGEAS.2017.190	2018	7	273,00	65,00
10	AGEAS.2017.190	2018	8	85,00	69,00
11	AGEAS.2017.190	2018	9	22,00	89,00
12	AGEAS.2017.190	2018	10	64,00	89,00
13	AGEAS.2017.190	2018	11	68,00	NULL
14	AGEAS.2017.190	2018	12	81,00	NULL
15	AGEAS.2017.190	2019	1	24,00	89,00
16	AGEAS.2017.190	2019	3	3,00	NULL
17	AGEAS.2017.190	2019	4	2,00	NULL

(Figura 8 - Comparação de informações do projeto AGEAS.2017.190)

Em termos de testes do relatório em Power BI, analisei o relatório para conseguir perceber se todas as métricas faziam sentido no âmbito e análise da produtividade e se possuía indicadores para distinguir a produtividade de cada tarefa e projeto. Com este pensamento, analisei o modelo de dados e detetei que criei bastantes métricas que ajudam na percepção da variação da produtividade, quer seja no contexto de um projeto ou de uma tarefa.

Criei métricas que mostram a variação *Month Over Month* (MoM) para conseguirmos detetar perdas/ganhos de produtividade ou de outros valores. Outra métrica interessante é a de Horas Realizadas Year To Date (YTD) que me fornece as horas realizadas até ao mês em questão, mas dentro do mesmo ano. Também tenho métricas que contam o número de projetos, tarefas, funcionários e perfis, para conseguir verificar quantos elementos de uma certa dimensão estão associados a cada classificação de produtividade, por exemplo.

4. Conclusões

De forma resumida, este projeto consistiu em criar um dashboard (Power BI Report) para análise dos dados presentes numa base de dados fornecida sem nenhum tratamento e/ou verificação de dados. Foi necessário utilizar o Azure Data Factory para a *data ingestion*, que me permitiu passar os dados da base de dados original para uma base de dados stg, onde posso executar possíveis alterações. No meu caso utilizei schemas diferentes, em vez de base de dados distintas, conseguindo minimizar os custos no Azure.

De seguida, criei o modelo dw, que consiste na representação da base de dados que me dá suporte ao relatório Power BI. Com o modelo em mente, executei a transformação de dados através de operações com dataframes num Python Notebook presente no Azure Databricks. Trabalhar unicamente com operações entre dataframes resultou numa eficiência indicada para trabalho com dados, pois a introdução de ciclos neste tipo de processos é impensável pelo tempo de execução.

Com os dados tratados e com a estrutura que pretendia, avancei para a execução do dashboard (Power BI). Primeiramente tratei de verificar o modelo de dados e a forma como estava a carregar as tabelas para o relatório. Depois, comecei a explorar e experimentar algumas possibilidades para a estrutura. Criei métricas que me permitem calcular campos de dados, utilizando a linguagem DAX que permite criar fórmulas e expressões no Power BI. Após muitas versões e alterações, cheguei a uma versão final, em que é possível analisar a produtividade dos projetos e das tarefas de cada um deles. O último passo foi a execução de testes na data ingestion, data transformation e no relatório Power BI, para verificar se estava tudo em conformidade com o previsto.

Uma das melhorias a executar no futuro para este projeto, seria a alteração da dimensão calendário para possuir informação diária em vez de mensal. Desta forma seria possível analisar semanas, por exemplo. Algumas lacunas que poderiam melhorar a análise dos dados são as seguintes:

- atualizar o avanço dos projetos de forma mais regular, para ser possível melhorar o acompanhamento do progresso, comparando as horas realizadas até ao momento com as horas previstas em função do avanço atual;

- falta da informação do avanço de cada tarefa, à semelhança do que acontece com os projetos. É possível um acompanhamento mais pormenorizado no caso do projeto devido à existência deste campo.

Consegui aprender bastante (apesar de não ser um trabalho muito aprofundado na área) sobre os processos necessários e especificações das áreas de Engenharia de Dados e Análise de Dados. Não foram áreas tratadas nas unidades

curriculares da licenciatura, e por isso, este projeto integrador foi bastante útil para a minha formação como engenheiro informático. Na minha opinião, devemos ter, pelo menos, uma noção de como são os processos em cada campo da engenharia informática e ciência de computadores, para desta forma sermos melhores engenheiros.

5. Referências

Azure Documentation:

<https://learn.microsoft.com/en-us/azure/?product=popular>

Azure Databricks Documentation:

https://docs.databricks.com/?_gl=1*1csvqda*_gcl_au*NTA3MzE1NTA5LjE2ODU2MjE4Njc.&_ga=2.152145785.1901359658.1685621868-2001652000.1685621868

Power BI Documentation:

<https://learn.microsoft.com/pt-pt/power-bi/fundamentals>

DAX Documentation:

<https://learn.microsoft.com/pt-pt/dax/>

6. Anexos

Perspetiva Global de Projetos



(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/perspetiva_global_projetos.PNG)

Relação de Projetos

The dashboard displays the following key metrics:

- Projetos:** 2.346
- Média de Produtividade:** 3,38
- Horas Realizadas:** 2.653.557,60

Table of Project Details:

Código Projeto	Departamento	Área	Estado	Total de Horas Realizadas	Classificação de Produtividade	Descrição
Não Especificado	Não Especificado	Não Especificada	Não Definido	56,00	-1 Inválida	
BDIL.2022.429	Admin & Finance	AI & ML	Iniciado	2.176,00	2 Até 50% Horas Extra Consumidas	
SGPS.2021.035	Admin & Finance	Não Especificada	Iniciado	6.459,50	3 Até 25% Horas Extra Consumidas	
SGPS.2022.110	Admin & Finance	Não Especificada	Iniciado	8.066,00	5 Menos do Previsto	
SGPS.2023.120	Admin & Finance	Não Especificada	Iniciado	066,00	5 Menos do Previsto	
SGPS.2021.238	Administration	Chairman + CEO	Iniciado	3.440,00	4 Como Previsto	
ARMBENELUX.2023.143	Administration	Não Especificada	Iniciado	42,00	5 Menos do Previsto	
SGPS.2022.206	Administration	Não Especificada	Iniciado	3.632,00	3 Até 25% Horas Extra Consumidas	
BDIL.2023.104	Artificial Intelligence	Não Especificada	Iniciado	128,00	5 Menos do Previsto	
ARMIS.2022.458	Artificial Intelligence & RA	Não Especificada	Iniciado	2.040,00	2 Até 50% Horas Extra Consumidas	
UNICRE.2022.449	Backend & Middleware	Não Especificada	Iniciado	2.304,00	5 Menos do Previsto	
UNICRE.2022.404	Backend & Middleware	Não Especificada	Iniciado	1.228,00	5 Menos do Previsto	
UNICRE.2023.226	Backend & Middleware	Não Especificada	Iniciado	296,00	5 Menos do Previsto	
ARMIS.2023.030	Business Development	Business Development	Iniciado	91,00	5 Menos do Previsto	
OZONO.2023.024	Business Development	Business Development	Iniciado	143,00	5 Menos do Previsto	
OZONO.2023.026	Business Development	Business Development	Iniciado	37,00	5 Menos do Previsto	
OZONO.2023.028	Business Development	Business Development	Iniciado	24,00	5 Menos do Previsto	
OZONO.2023.029	Business Development	Business Development	Iniciado	116,00	5 Menos do Previsto	
OZONO.2023.259	Business Development	Business Development	Iniciado	56,00	5 Menos do Previsto	
OZONO.2023.237	Business Development	Não Especificada	Iniciado	12,00	4 Como Previsto	
ARMBENELUX.2022.209	Business Development for BENELUX	Não Especificada	Iniciado	3,00	5 Menos do Previsto	
ADS.2021.307	Business Development for DS	Business Development	Iniciado	871,00	3 Até 25% Horas Extra Consumidas	
ADS.2022.268	Business Development for DS	Business DEvlopment	Iniciado	1.734,00	0 Mais de 75% Horas Extra Consumidas	
Total				2.653.557,60	5 Menos do Previsto	

(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/relacao_projetos.PNG)

Análise por Projeto



(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/analise_projeto.PNG)

Perspetiva Global de Tarefas



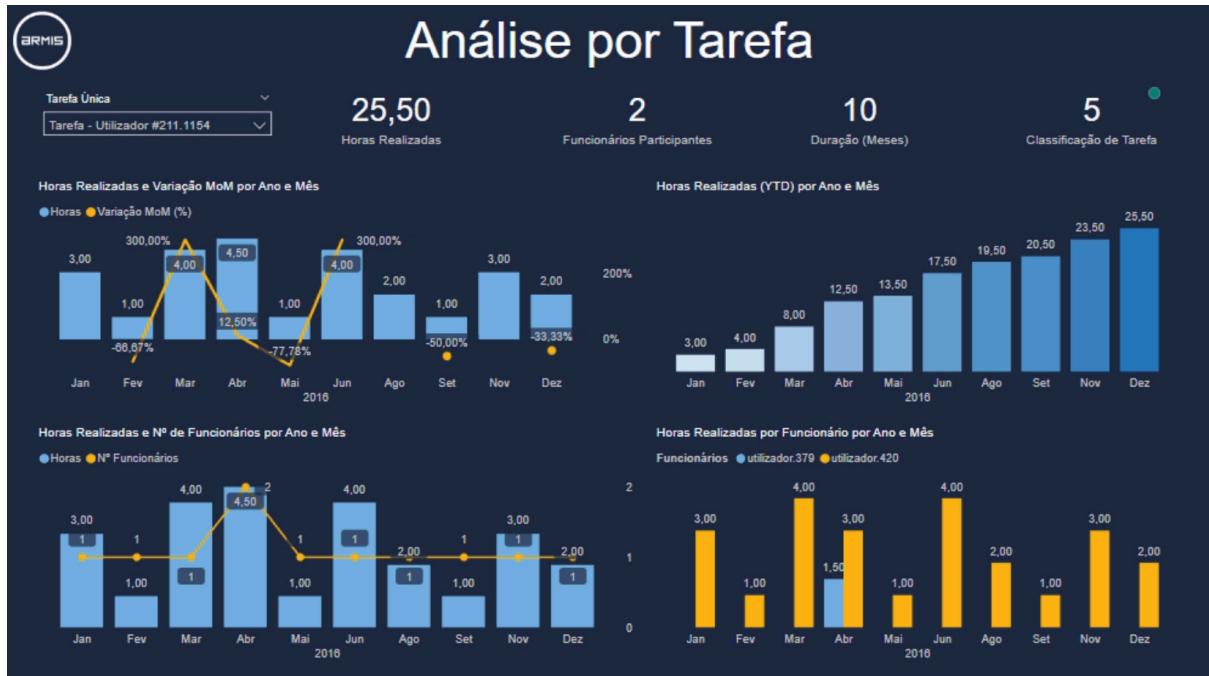
(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/perspetiva_global_tarefas.PNG)

Relação de Tarefas



(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/relacao_tarefas.PNG)

Análise por Tarefa



(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/analise_tarefa.PNG)

Imputações por Tarefa e Funcionário

Imputações por Tarefa e Funcionário

Tarefa Única	Ano	Nome_Funcionario	t	2010			2011												2012				
				Out	Nov	Dez	Total	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Total	Jan		
Não Definido.-1		utilizador:221																					
		utilizador:223																					
		utilizador:225																					
		utilizador:227																					
		utilizador:228		48,00	56,00	160,00	264,00	158,00	160,00	176,00	152,00	176,00	152,00	168,00	176,00	176,00	160,00	144,00	160,00	1.968,00	176,00		
		utilizador:229																					
		utilizador:231																					
		utilizador:233																					
		utilizador:235																					
		utilizador:237																					
		utilizador:238																					
		utilizador:239																					
		utilizador:240																					
		utilizador:241																					
		utilizador:242																					
		utilizador:243		77,00	163,00	157,00	397,00	96,00	96,00	123,00	34,00			40,00						24,00	413,00	178,00	
		utilizador:244																					
		utilizador:245																					
		utilizador:246																					
		utilizador:247																					
		utilizador:248																					
		utilizador:250																					
		utilizador:251																					
		utilizador:252		2,00	160,00	168,00	160,00	620,00	16,00	120,00	176,00	152,00	184,00	152,00	168,00	176,00	176,00	160,00	168,00	136,00	1.784,00	176,00	
		utilizador:254																					
		utilizador:255																					
		utilizador:256																					
		utilizador:257																					
		utilizador:258																					
		utilizador:259																					
		utilizador:260																					
				88,00			88,00	22,00			8,00			24,00	40,00	1.150,00	160,00	240,00	40,00	168,00	160,00	2.012,00	176,00

(https://github.com/diogofonte/feup-pi/blob/main/screenshots/power_bi/imputacoes_tarefa_funcionario.PNG)