

Integração API Nubank

&

Construção Dashboard

Autor: Rafael Moreira

Orientador: Anderson Nascimento

Pontifícia Universidade Católica do Rio de Janeiro

Curso de Pós Graduação Business Intelligence Master

SUMÁRIO

1	INTRODUÇÃO	3
1.1	OBJETIVOS	4
1.2	ARQUITETURA	4
2	METODOLOGIA	5
2.1	PYTHON	5
2.1.1	API	5
2.1.2	ETL	6
2.1.3	Conexão SQL	6
2.2	SQL	6
2.3	Power BI	7
2.3.1	Power Query	7
2.3.2	Modelagem de Dados	7
2.3.3	Calulator / DAX	8
3	RESULTADOS FINAIS	8
4	CONCLUSÃO	9
5	REFERÊNCIAS	9
6	ANEXOS	10
6.1	Jupyter Notebook (.html)	10
6.2	Documentação <i>pynubank</i>	11
6.3	Power BI (.pbix)	12

1) INTRODUÇÃO

O conceito de *Business Intelligence* compõe-se de um conjunto de metodologias de gestão implementadas através de ferramentas de software, cuja função é proporcionar ganhos nos processos decisórios gerenciais e de alta administração nas organizações, baseada na capacidade analítica das ferramentas que integram um só lugar todas as informações necessárias ao processo decisório. Reforça-se que o objetivo do *Business Intelligence* é transformar dados em conhecimento suportando o processo decisório, no intuito de gerar vantagens competitivas. (ANGELONI E REIS, 2006)

O principal objetivo dos sistemas de *BI* é disponibilizar acesso interativo aos dados, permitindo sua manipulação disponibilizando aos gestores condições para efetuarem análises apropriadas. Analisando dados históricos e correntes, o *BI* maximiza as chances de a decisão ser efetuada de forma mais correta em relação ao atual estado do negócio. (PENNA, 2013)

A **Figura 1** demonstra um esquema que exemplifica os componentes do *BI*. O primeiro passo para sua implementação é possuir uma ou mais fontes de dados. Os dados são a matéria prima do projeto de *BI*. Em seguida um processo de *ETL* é realizado, sendo responsável pela preparação dos dados a serem armazenados em um *Data Warehouse*. Essa é a fase mais crítica, pois envolve a movimentação dos dados.

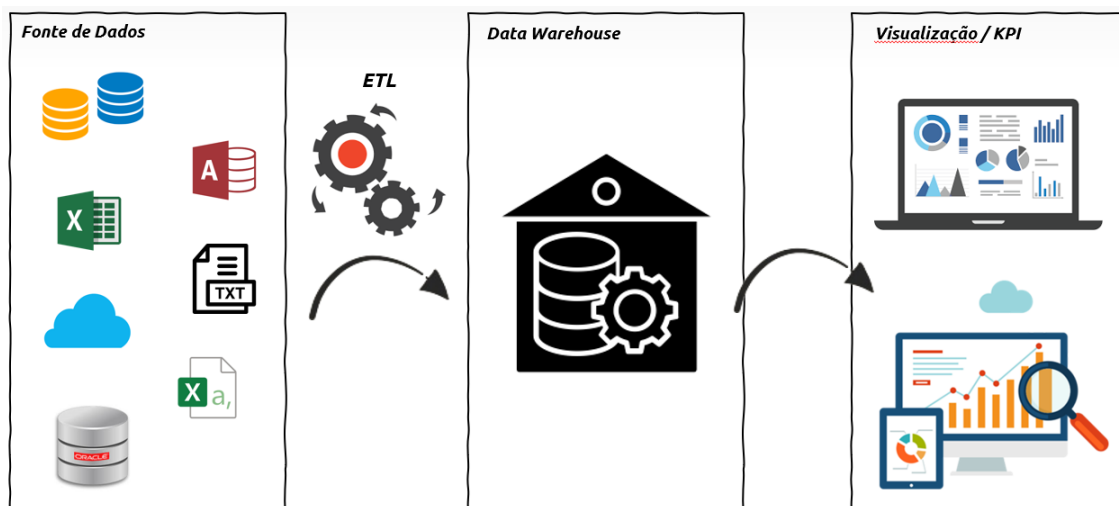


Figura 1: Componentes de um ambiente de *BI*

Os dados são consolidados e armazenados em um *Data Warehouse* de uma forma mais eficiente, possibilitando assim uma futura consulta mais eficaz e rápida. Em seguida é realizada uma rica análise em cima dos dados (*Data Mining*), para traçarem novas oportunidades e estratégias para o futuro. Por fim, são necessárias ferramentas

que possibilitem a visualização destas informações, de forma amigável e gerencial, auxiliando os usuários a tomarem suas decisões.

1.1 Objetivo

Implementar um projeto de *BI* em cima das minhas movimentações financeiras do banco digital *Nubank*. A maior motivação para esse projeto, é criar algo bastante relevante para o meu dia a dia, usando ferramentas tecnológicas de linha (*Python*, *SQL* e *Power BI*).

1.2 Arquitetura

A **Figura 2** demonstra um esquema que exemplifica a arquitetura desenvolvida no projeto.

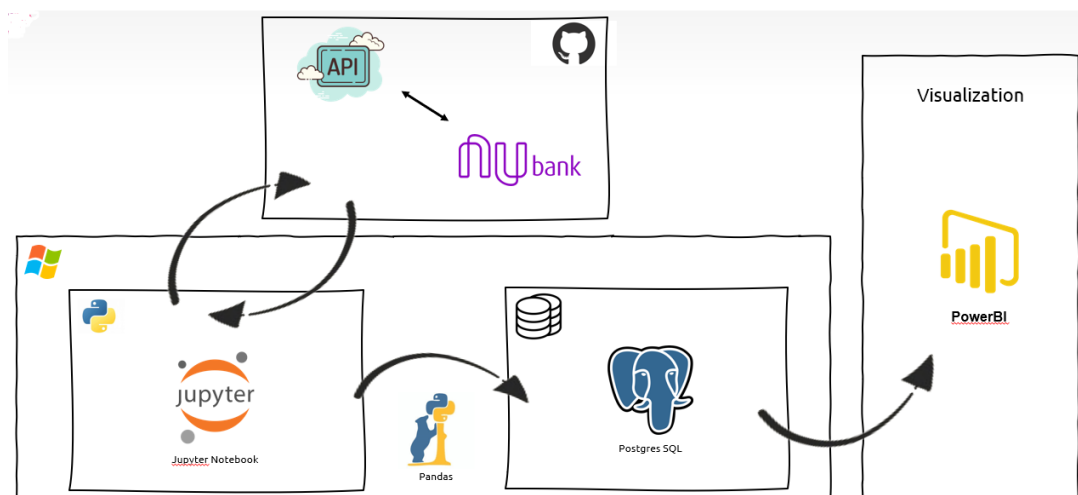


Figura 2: Arquitetura desenvolvida

Em uma máquina *Windows*, instalou-se o *Python* e o *Jupyter Notebook* como ambiente de execução da linguagem. Os programas *PostgreSQL* e o *Power BI* também foram instalados.

A fonte de dados para esse projeto é disponibilizada pelo *Nubank* por chamadas de API. O próprio banco desenvolveu uma biblioteca *Python*, encapsulou e disponibilizou no *Github* (*pynubank*). Os dados são recebidos no formato *json* e tratados com a biblioteca *pandas*, antes de serem inseridos no banco de dados (conexão direta via *Python*). Por fim, conectou-se o *Power BI* ao banco de dados e um dashboard interativo foi desenvolvido.

2) METODOLOGIA

2.1 Python

Todo desenvolvimento desta linguagem foi realizado no *Jupyter Notebook*, e o *script* encontra-se em Anexo (**Anexo 1**).

2.1.1 API

Para comunicação com o banco e extração dos dados, diversas chamadas APIs são utilizadas. Os próprios desenvolvedores do Nubank criaram uma biblioteca encapsulada com diversas funções para facilitar essas chamadas de API. A biblioteca é chamada de *pynubank* e está encapsulada no *Github*.

Cinco chamadas de API são realizadas:

- *get_qr_code*: chamada para garantir o TOKEN
- *get_account_balance*: chamada para informação do saldo da conta
- *get_bills*: lista de dicionários contendo todas as faturas do seu cartão de crédito
- *get_card_statements*: lista de dicionários contendo todas as transações de seu cartão de crédito
- *get_account_statements*: lista de dicionários contendo todas as transações de débito

Para maior segurança do processo e dos dados do cliente, após realizar a chamada *get_qr_code*, um QR Code é gerado sendo necessário escaneá-lo pelo aplicativo do celular, conforme demonstrado pela **Figura 3**.



Figura 3: Autenticação do cliente

2.1.2 ETL

Utiliza-se a biblioteca *Pandas*, do *Python*, para realizar todos os tratamentos dos objetos recebidos pela API.

Duas principais funções foram criadas para esse procedimento:

- *json_transactions*: tratamento do objeto *Transaction*
- *json_statements*: tratamento do objeto *Statements*

2.1.3 Conexão SQL

Para importação dos dados no banco de dados (*Postgres*), chamamos uma biblioteca *Python* (*sqlalchemy*), para se conectar com o banco, e criamos uma função chamada *sql_conn*, conforme demonstrado pela **Figura 4**.

```
# SQL connection
def sql_conn():
    # Definindo Variáveis de Conexão
    hostname = 'localhost'
    username = 'postgres'
    password = 'postgres'
    database = 'db_nubank'
    port = '5432'

    # Criando a Engine
    engine = create_engine('postgresql+psycopg2://' + str(username) + ':' + str(password)
                          + '@' + str(hostname) + ':' + str(port) + '/' + str(database))

    return hostname, username, password, database, port, engine
```

Figura 4: Função responsável pela conexão com o banco de dados *Postgres*

2.2 SQL

Conforme mencionado anteriormente, utilizou-se o banco de dados *Postgres*, e 5 tabelas foram criadas:

- *tb_cc_transaction*
- *tb_order*
- *tb_order_aux*
- *tb_bills*
- *tb_saldo*

2.3 Power BI

Utilizou-se o Microsoft Power BI como ferramenta de visualização de dados, e o dashboard encontra-se em Anexo (**Anexo 2**).

2.3.1 Power Query

O Power Query foi utilizado para fazer algumas etapas do ETL, e para criar a dim-calendario, conforme demonstrado pela **Figura 5**.

dim_calendario

Opções de Exibição

```
let
    Fonte = List.Dates(dt_inicio, 365*qtdd_anos, intervalo),
    #Tipo Alterado = Table.FromList(Fonte, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #Ano Inserido = Table.AddColumn(#Tipo Alterado, "Ano", each Date.Year([Column1]), Int64.Type),
    #Mês Inserido = Table.AddColumn(#Ano Inserido, "Mês", each Date.Month([Column1]), Int64.Type),
    #Nome do Mês Inserido = Table.AddColumn(#Mês Inserido, "Nome do Mês", each Date.MonthName([Column1]), type text),
    #Trimestre Inserido = Table.AddColumn(#Nome do Mês Inserido, "Trimestre", each Date.QuarterOfYear([Column1]), Int64.Type),
    #Semana do Ano Inserida = Table.AddColumn(#Trimestre Inserido, "Semana do Ano", each Date.WeekOfYear([Column1]), Int64.Type),
    #Dia Inserido = Table.AddColumn(#Semana do Ano Inserida, "Dia", each Date.Day([Column1]), Int64.Type),
    #Tipo Alterado1 = Table.TransformColumnTypes(#Dia Inserido,{{"Column1", type date}}),
    #Colunas Renomeadas = Table.RenameColumns(#Tipo Alterado1,{{"Column1", "data"}, {"Ano", "ano"}, {"Mês", "mes"}, {"Nome do Mês", "nome_mes"}, {"Trimestre", "trimestre"}, {"Semana do Ano", "semana"}, {"Dia", "dia"}}),
    #Coluna Duplicada = Table.DuplicateColumn(#Colunas Renomeadas, "data", "g_data")
in
    #Coluna Duplicada
```

Figura 5: Script em M para geração da dim-calendario

2.3.2 Modelagem de dados

A **Figura 6** ilustra a modelagem de dados aplicada ao problema, utilizou-se conceito de *star schema*.

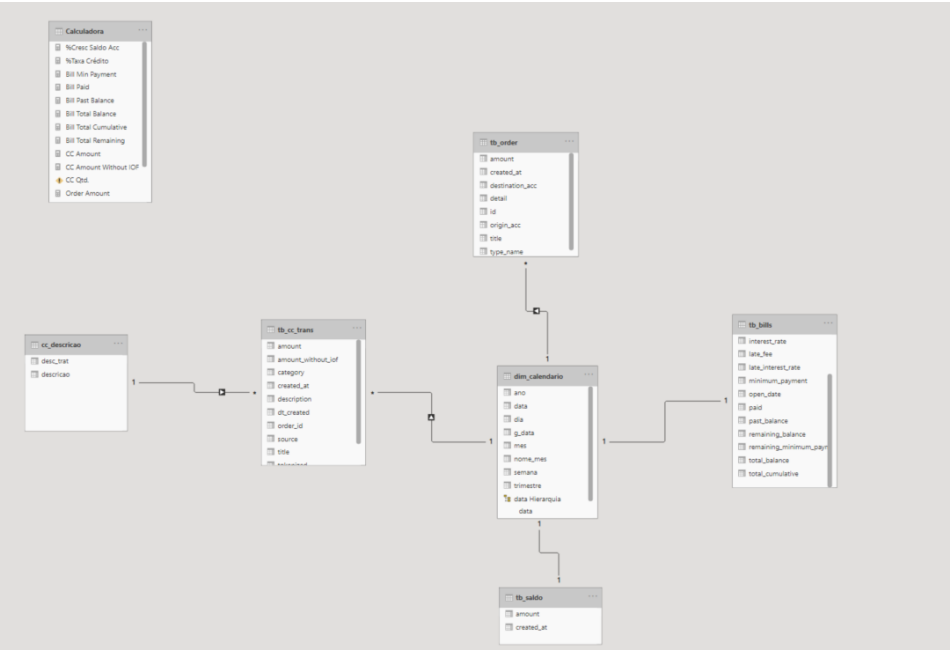


Figura 6: Modelagem de dados

2.3.3 Calculator / DAX

Uma tabela foi criada em branca, e transformada em Calculadora, com o intuito de deixar o PBI mais organizado, separando tudo que é Medida, de Campos.

Algumas Medidas foram criadas em DAX, e em algumas utiliza-se o *CALCULATE*, uma das fórmulas mais importantes do *Power BI* (**Figura 7**)

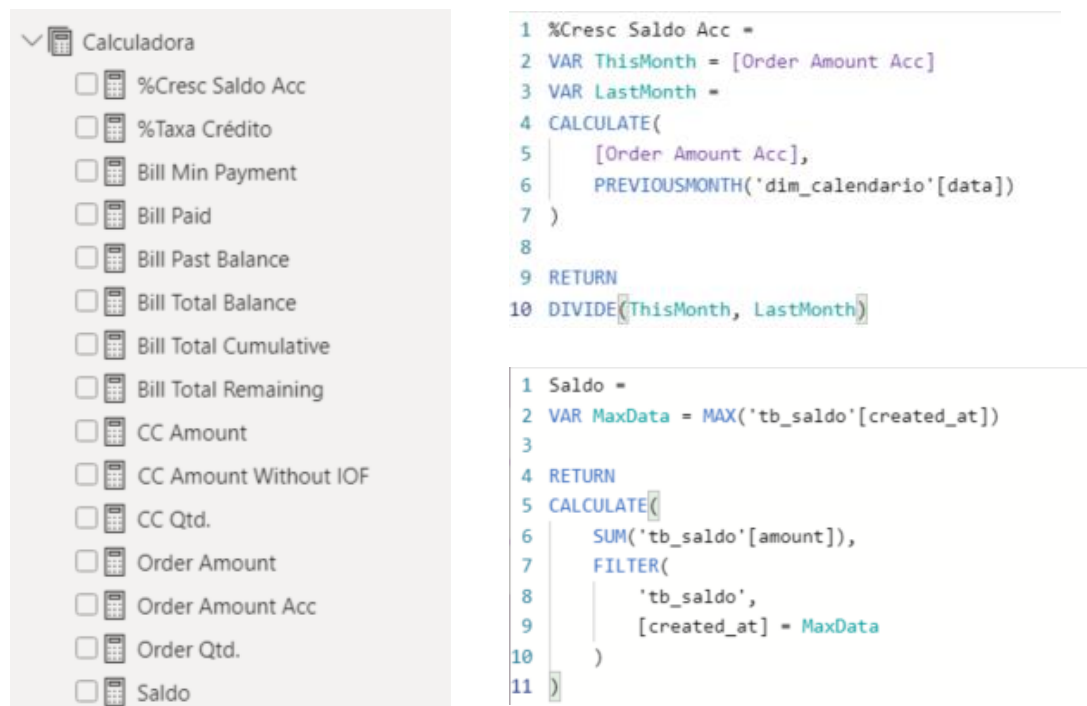


Figura 7: Calculator e algumas medidas DAX

3) RESULTADOS FINAIS

O **Anexo 3** ilustra o Resultado Final do Projeto, que é o dashboard desenvolvido para acompanhar as minhas movimentações financeiras do banco digital *Nubank*. O relatório possui algumas abas, com diferentes informações e bastante detalhe.

Vale ressaltar que o **Anexo 3** possui algumas informações sensíveis, por isso valores e nomes de pessoas foram ocultados.

4) CONCLUSÃO

O projeto foi desenvolvido com base da necessidade do meu dia a dia, e utilização de ferramentas tecnológicas de linha (*Python, SQL, Power BI*). Com esse dashboard (resultado do projeto), tenho a visibilidade de todos meus gastos (passados e futuros). Com certeza minhas tomadas de decisão vão ser influenciadas positivamente, pelo fato de ter um histórico consolidado para analisar os acontecimentos.

5) REFERÊNCIAS

ANTONELLI, Ricardo. **Conhecendo o Business Intelligence (BI)**. Revista TECAP – Número 03 – Ano 3 – Volume 3 – 2009 anual

SAITO, Evandro. HORITA, Ricardo. **BUSINESS INTELLIGENCE COMO UMA FERRAMENTA DE GESTÃO**. V Encontro Científico e Simpósio de Educação Unisalesiano 2015