

Case TON - Analista de Dados/Analytics  
na  
Área de Relacionamento Ativo



Candidato: Raphael da Silva Barradas

Prazo de entrega: 19/11/2020

Pessoal, esse documento foi elaborado com objetivo de informar as tecnologias/ferramentas utilizados na resolução do desafio proposto.

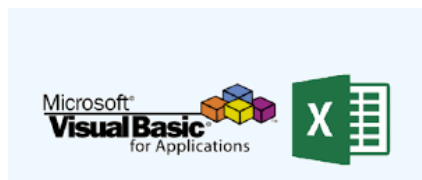
- **Tecnologias e ferramentas**



SQL Server como banco de dados, para armazenar as bases enviadas.



.NET (C#) para criação de script de limpeza das base e carga no SQL Server.



Excel + VBA para consumir os dados armazenados do SQL Server e criar as análises/visualizações.



Git para versionar o projeto  
GitHub para ser o repositório

- **Limpeza e carga das bases no SQL Server**

Uma dificuldade inicial do projeto foi entender as bases e os campos enviados para o desafio, pois existiam campos repetidos, linhas em branco e texto com caracteres codificados, conforme será detalhado abaixo.

Tabela 'cases': Chamados recebidos pelo time de Relacionamento com o Cliente.

Coluna	Descrição	Observação
accountid	Chave primária da tabela	Esse campo foi utilizado como chave estrangeira. (Creds)
date_ref	Data do chamado	
channelid	Canal do chamado (chat = 1; telefone = 2; e-mail = 3)	Foi criada uma tabela de dimensão para armazenar a descrição do canal.
waitingtime	Tempo de espera para atendimento (segundos)	
missed	Flag de chamado atendido	
pesquisa_de_satisfa_o_c	Flag de pesquisa de satisfação preenchida	
assunto	Assunto do chamado	
Id	Chave da tabela	Conteúdo igual ao campo accountid.

Na base Cases.csv, foram realizados os tratamentos abaixo:

Foram removidos registros sem conteúdo:

	A
1	,accountid,date_ref,channelid,waitingtime,missed,pesquisa_de_satisfa_o_c,assunto,Id
2	0,0,0,0,0,0,0,0
3	1,0,0,0,0,0,0,0
4	2,0,0,0,0,0,0,0
5	3,0,0,0,0,0,0,0
6	4,0013j00002z0CeEAAU,2020-07-31,2,15,FALSE,,Aplicativo:DÃvidas funcionalidades App:Primeiro Acesso,0013j00002z0CeEAAU
7	5,0013j00002z0CeEAAU,2020-07-31,2,15,FALSE,,Produto:S920:AtivaÃsÃo,0013j00002z0CeEAAU
8	6,0013j00002z0CeEAAU,2020-09-23,2,1,FALSE,,Produto:CartÃo prÃ- pago:DÃvidas,0013j00002z0CeEAAU
9	7,0013j00002zQgldAAC,2020-08-29,2,6,FALSE,,LogÃstica:Envio de CartÃo:Tracking,0013j00002zQgldAAC
10	8,0013j00002zQgldAAC,2020-08-29,2,6,FALSE,,Pedido:Cancelamento:Outros,0013j00002zQgldAAC
11	9,0013j00002zQgldAAC,2020-08-29,2,6,FALSE,,Produto:CartÃo prÃ- pago:DÃvidas,0013j00002zQgldAAC
12	10,0013j00002zQgldAAC,2020-09-05,2,10,FALSE,,Aplicativo:DÃvidas funcionalidades App:RedefiniÃo de senha,0013j00002zQgldAAC

Utilizei o número da linha como conteúdo do campo Id, para deixar a tabela com uma chave primária.

## Tratamento dos caracteres do campo “assunto”:

Tabela 'creds': Clientes credenciados.

Coluna	Descrição	Observação
cred_date	Data de credenciamento	
shipping_address_city	Endereço de entrega – cidade	
shipping_address_state	Endereço de entrega – estado	
max_machine	Produto adquirido	
accountid	Chave secundária da tabela	Utilizado como chave primária da tabela

Na base Creds.csv, foram realizados os tratamentos abaixo:

Foram removidos registros sem conteúdo na chave:

	cred_date	shipping_address_city	shipping_address_state	max_machine	accountid
1	0,2020-04-18,	Feira de Santana,BA,	T1,		
2	1,2020-10-16,	Bacuri,MA,	T1,		
3	2,2020-09-01,	Bernardo Sayão,TO,	T1,		
4	3,2020-08-29,	Rio de Janeiro,RJ,	T3,		
5	4,2020-07-28,	São Gonçalo,RJ,	T3,0013j00002z0CeEAAU		
6	5,2020-07-28,	São Gonçalo,RJ,	T3,0013j00002z0CeEAAU		
7	6,2020-07-28,	São Gonçalo,RJ,	T3,0013j00002z0CeEAAU		

Foram removidos registros com chave duplicada:

1	,cred_date,shipping_address_city,shipping_address_state,max_machine,accountid	
2	0,2020-04-18,Feira de Santana,BA,T1,	
3	1,2020-10-16,Bacuri,MA,T1,	
4	2,2020-09-01,Bernardo Sayão,TO,T1,	
5	3,2020-08-29,Rio de Janeiro,RJ,T3,	
6	4,2020-07-28,São Gonçalo,RJ,T3,0013j00002z0CeEAAU	
7	5,2020-07-28,São Gonçalo,RJ,T3,0013j00002z0CeEAAU	
8	6,2020-07-28,São Gonçalo,RJ,T3,0013j00002z0CeEAAU	

- **Scripts de criação das tabelas no SQL Server**

OBS: Para melhor forma de manipular os dados, separei o campo assunto em CATEGORIA\_1, CATEGORIA\_2 e CATEGORIA\_3.

/\*\*\*\*\* Script de criação da tabela TB\_CASES \*\*\*\*\*/

```
CREATE TABLE Raphael.dbo.TB_CASES (
    ID INT NOT NULL,
    ACCOUNTID VARCHAR(50) NOT NULL,
    DATE_REF DATE NOT NULL,
    CHANNELID INT NOT NULL,
    WAITINGTIME INT NOT NULL,
    MISSED BIT NOT NULL,
    FEEDBACK VARCHAR(50),
    CATEGORY_1 VARCHAR(200),
    CATEGORY_2 VARCHAR(200),
    CATEGORY_3 VARCHAR(200)

    CONSTRAINT PK_ID PRIMARY KEY (ID),
    CONSTRAINT FK_ACCOUNTID FOREIGN KEY (ACCOUNTID)
        REFERENCES Raphael.dbo.TB_CREDS (ACCOUNTID),
    CONSTRAINT FK_CHANNELID FOREIGN KEY (CHANNELID)
        REFERENCES Raphael.dbo.DM_CHANNEL (CHANNELID)
);
```

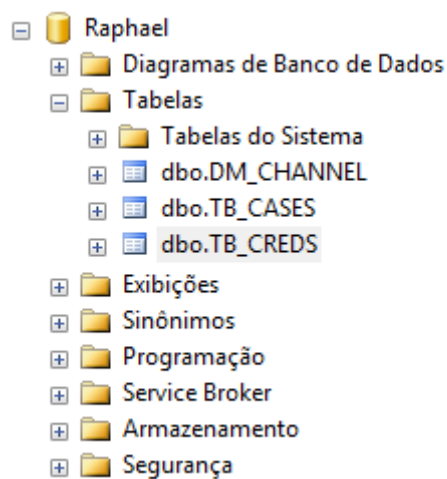
/\*\*\*\*\* Script de criação da tabela TB\_CREDS \*\*\*\*\*/

```
CREATE TABLE Raphael.dbo.TB_CREDS (  
    ACCOUNTID VARCHAR(50) NOT NULL,  
    CRED_DATE DATE NOT NULL,  
    SHIPPING_ADDRESS_CITY VARCHAR(500) NOT NULL,  
    SHIPPING_ADDRESS_STATE VARCHAR(2) NOT NULL,  
    MAX_MACHINE VARCHAR(50) NOT NULL  
  
    CONSTRAINT PK_ACCOUNTID PRIMARY KEY (ACCOUNTID)  
);
```

/\*\*\*\*\* Script de criação da tabela de dimensão DM\_CHANNEL \*\*\*\*\*/

```
CREATE TABLE Raphael.dbo.DM_CHANNEL (  
    CHANNELID INT NOT NULL,  
    DESC_CHANNEL VARCHAR(50) NOT NULL  
  
    CONSTRAINT PK_CHANNELID PRIMARY KEY (CHANNELID)  
);
```

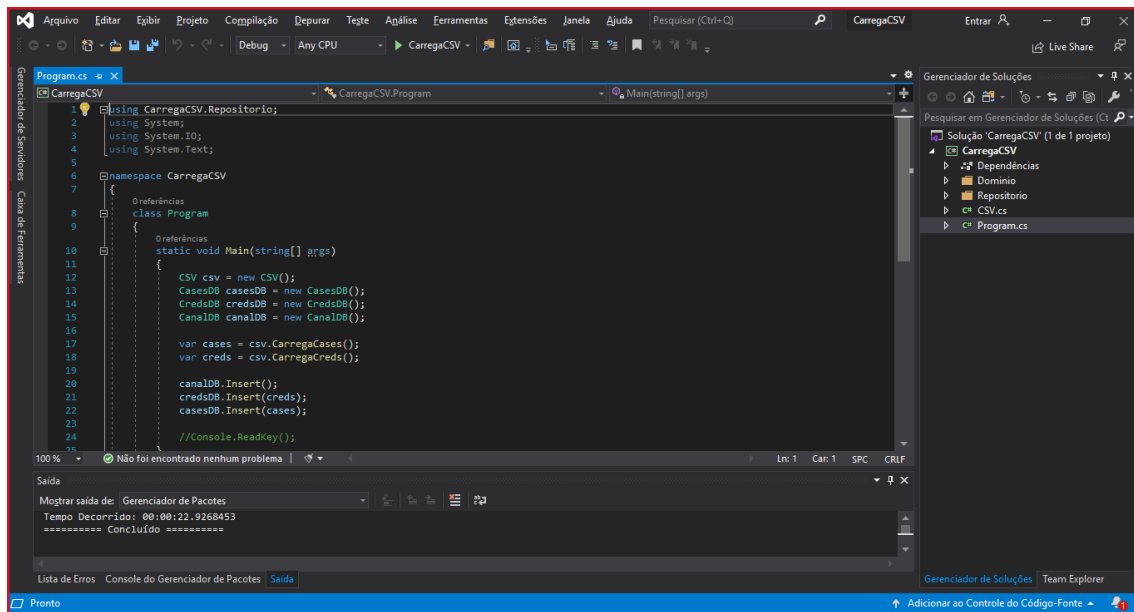
Tabelas criadas:



- **.NET (C#) – Programa de carga**

Foi desenvolvido um programa console application (CarregaCSV) para realizar a carga no SQL Server.

Para não ficar um documento muito grande, o código utilizado estará no GitHub, junto com o restante do desafio.



The screenshot displays the Visual Studio IDE with the 'CarregaCSV' project open. The main window shows the source code for 'Program.cs' in the 'CarregaCSV' namespace. The code defines a 'Program' class with a 'Main' method that initializes a 'CSV' object and three database objects ('CasesDB', 'CredsDB', 'CanalDB'), then calls methods to load data from CSV files into these databases. The output window at the bottom shows the command prompt output: 'Mostrar saída de: Gerenciador de Pacotes', 'Tempo Decorrido: 00:00:22.9268453', and '==== Concluído =====', indicating the program ran successfully. The right-hand side of the IDE shows the 'Gerenciador de Soluções' (Solution Explorer) with the project structure.

```
1 using CarregaCSV.Repositorio;
2 using System;
3 using System.IO;
4 using System.Text;
5
6 namespace CarregaCSV
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             CSV csv = new CSV();
13             CasesDB casesDB = new CasesDB();
14             CredsDB credsDB = new CredsDB();
15             CanalDB canalDB = new CanalDB();
16
17             var cases = csv.CarregaCases();
18             var creds = csv.CarregaCreds();
19
20             canalDB.Insert();
21             credsDB.Insert(creds);
22             casesDB.Insert(cases);
23
24             //Console.ReadKey();
25         }
26     }
27 }
```

Saída

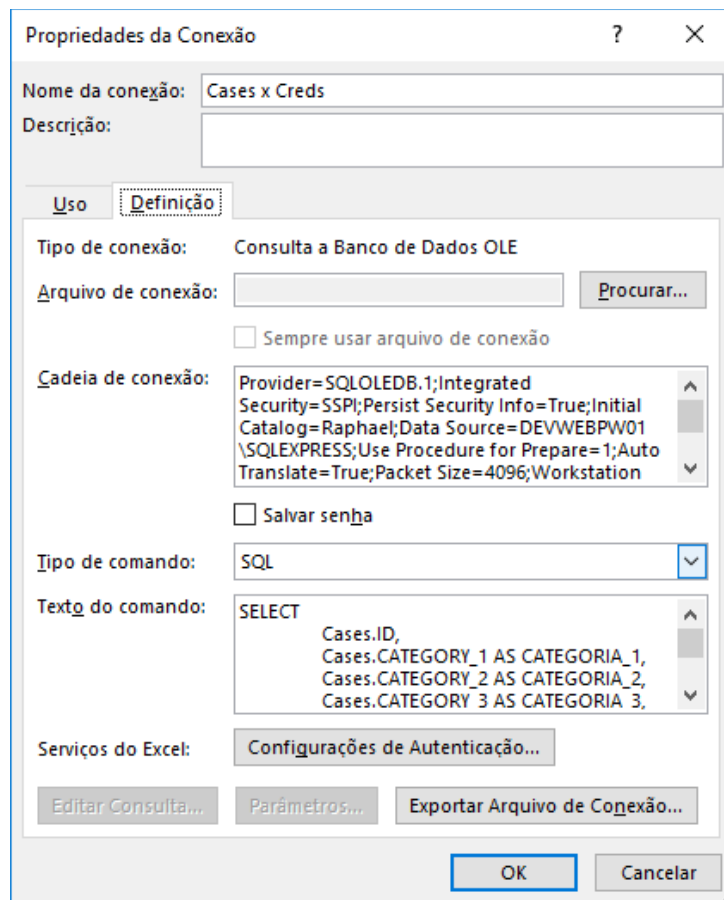
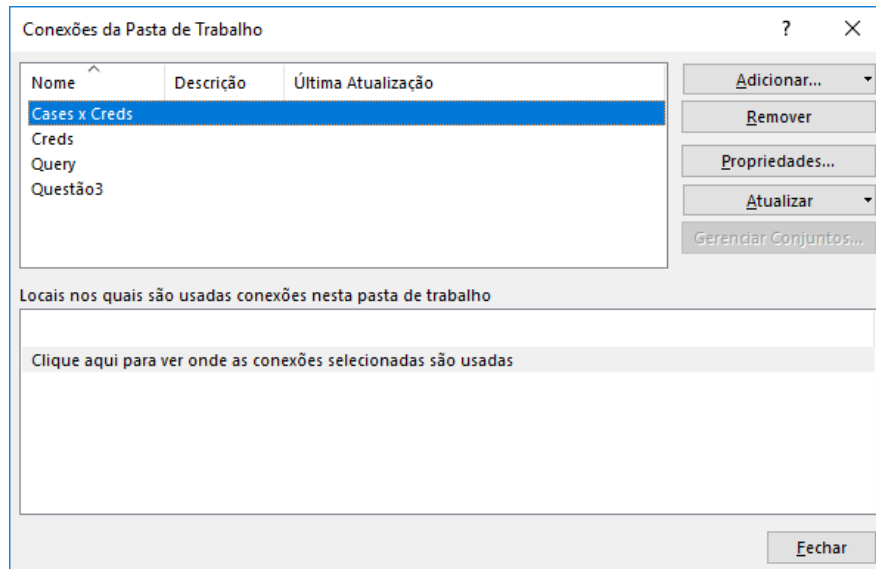
Mostrar saída de: Gerenciador de Pacotes

Tempo Decorrido: 00:00:22.9268453

==== Concluído =====

- **Excel – Conexão com o SQL Server**

Para consumir os dados do SQL Server, foram criadas algumas conexões de dados no excel, conforme abaixo.



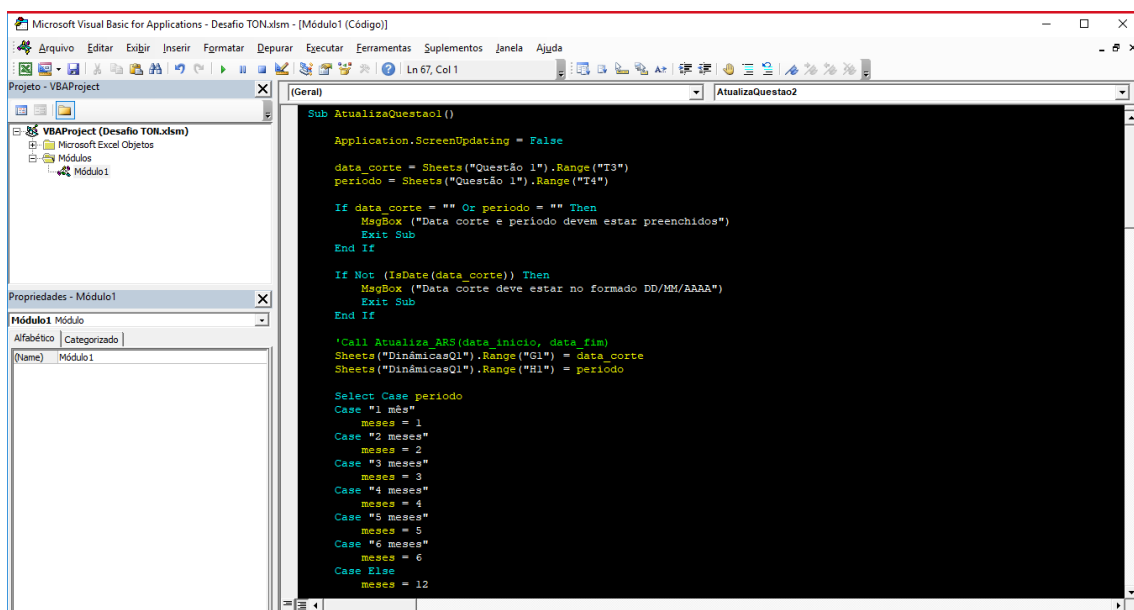


String de conexão:

Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=True;Initial Catalog=Raphael;Data Source=DEVWEBPW01\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=LAPCORPRJ097175;Use Encryption for Data=False;Tag with column collation when possible=False

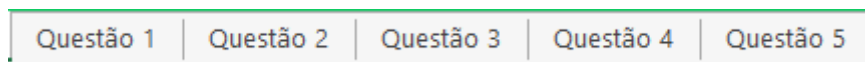
- **VBA para auxiliar a conexão com o Sql Server, validações de campos e filtros dinâmicos.**

Os scripts realizados no vba podem ser acessados diretamente na planilha “Desafio TON.xlsm”, pressionando Alt + F11.



- Excel para visualização dos dados e análises

Cada questão foi desenvolvida em uma “Aba” separada, para maior organização:

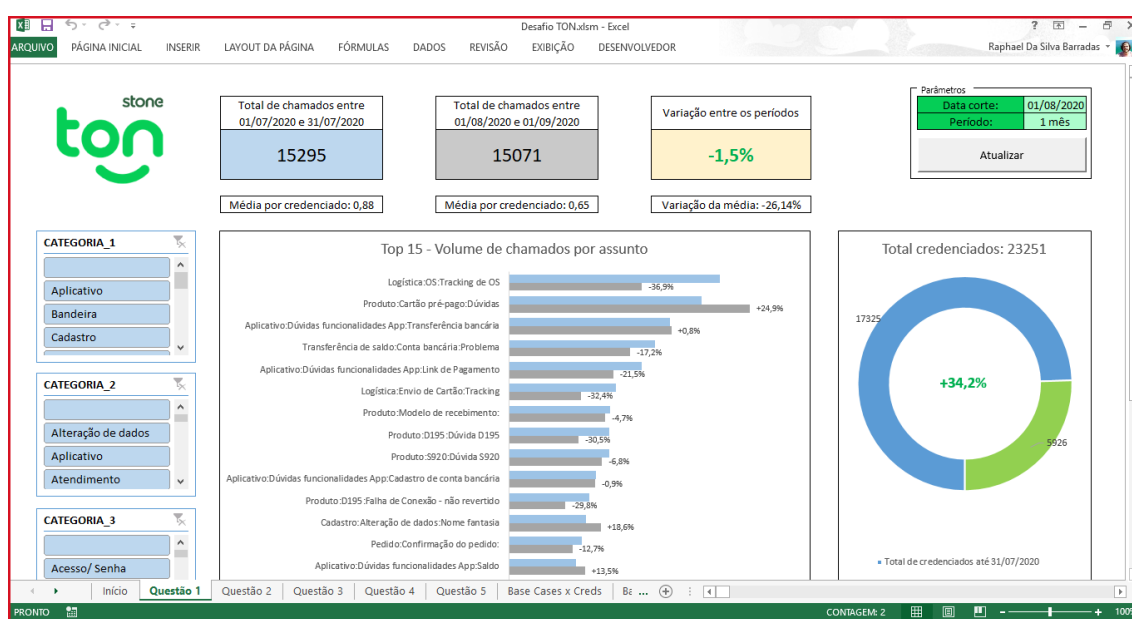


Aqui neste documento, colocarei apenas um print de cada questão, como exemplo.

Acesse a planilha para utilizar os filtros e visualizar os resultados de forma dinâmica.

### Questão 1:

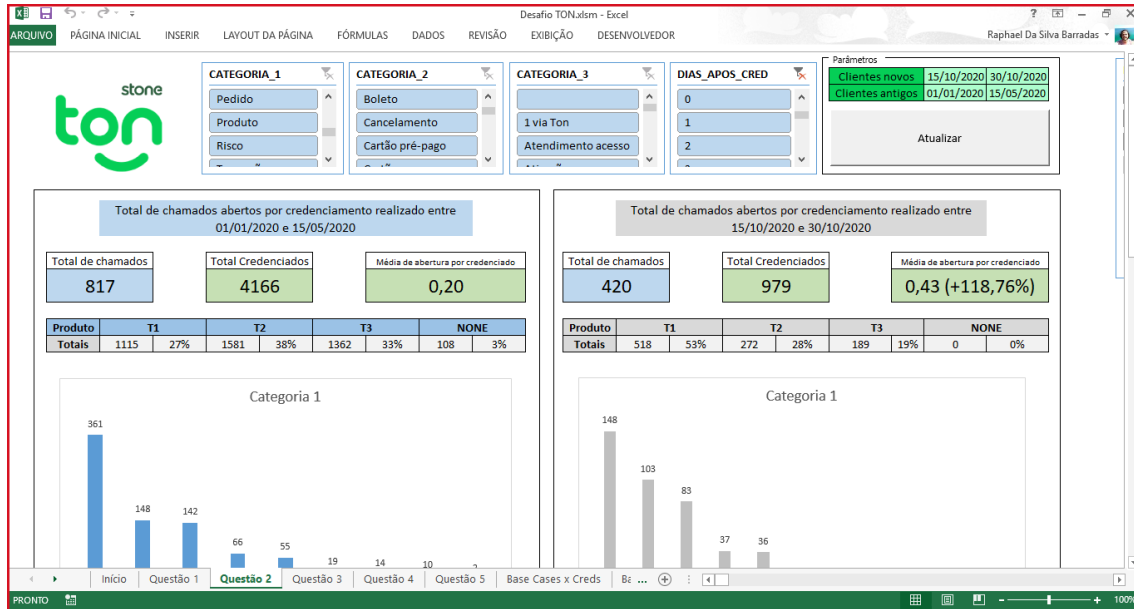
Obs: No lugar de colocar uma análise fixa com a data 01/08/2020, coloquei de forma parametrizada, para que seja possível realizar outras análises.



A minha análise está logo abaixo do gráfico “Top 15 – Volume de chamados por assunto”.

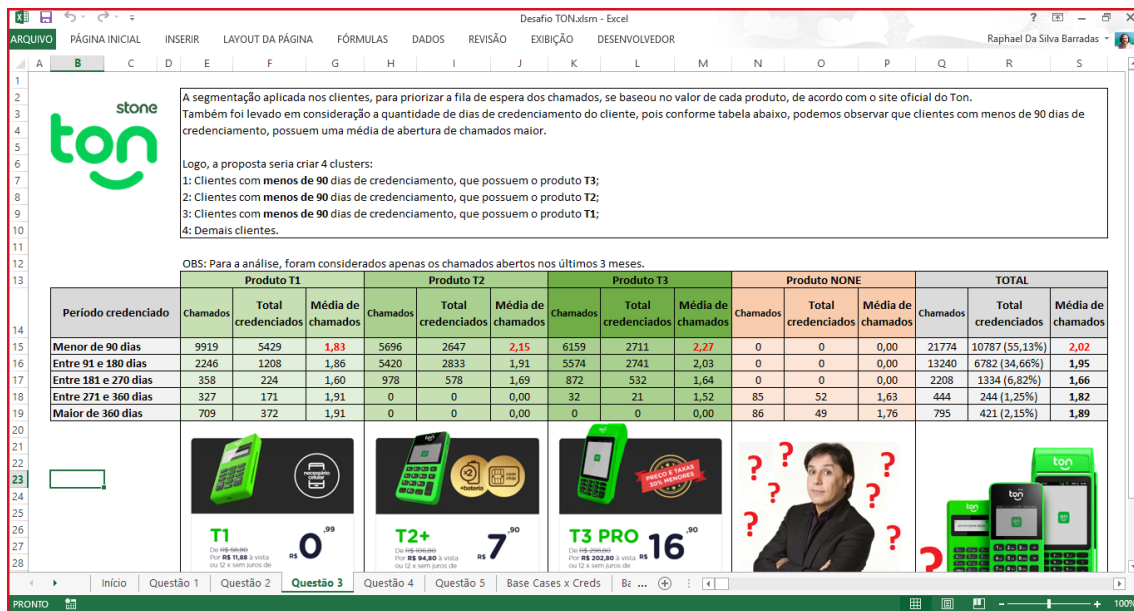
## Questão 2:

Obs: A questão utilizava termos como “Clientes mais recentes” e “Logo após o seu credenciamento”. Como essa definição não é exata, coloquei de forma parametrizada.



A minha análise está logo abaixo do gráfico “Categoria 3”.

## Questão 3:



## Questão 4:

As queries são exibidas de acordo com os botões.

Query T3 < 90 dias

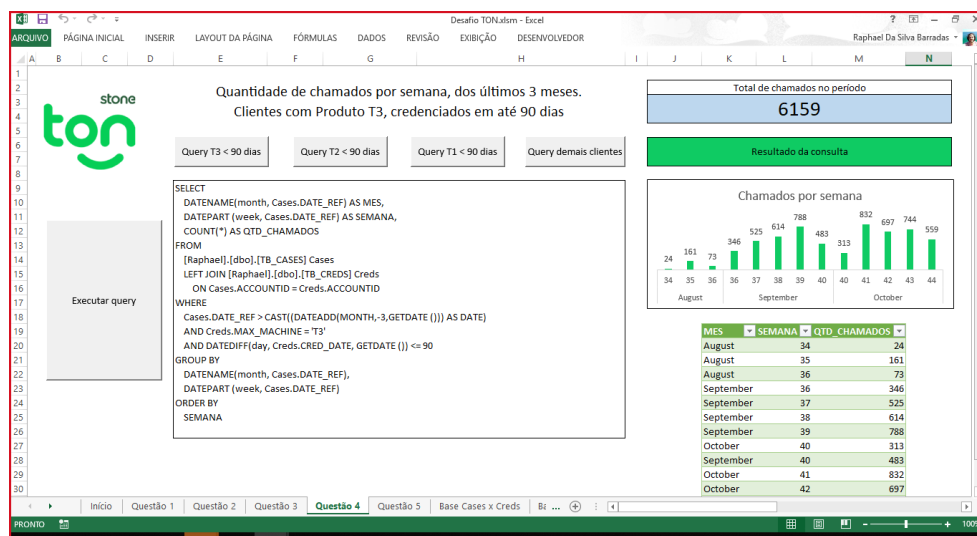
Query T2 < 90 dias

Query T1 < 90 dias

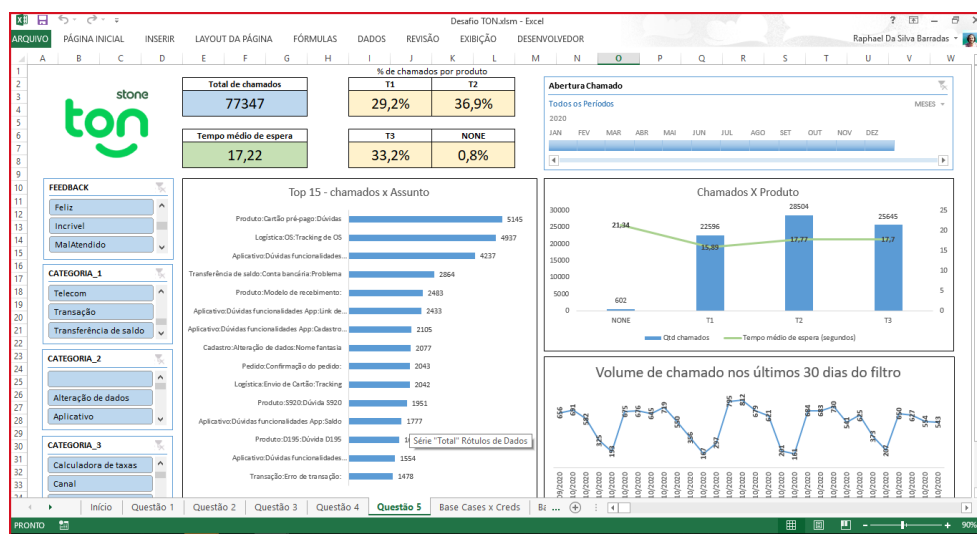
Query demais clientes

Obs: O botão “Executar query”, envia a consulta para o SQL Server e retorna os resultados, atualizando as informações dessa tela. Porém, como o banco está local, não será possível executar essa consulta de outra máquina.

Vídeo demonstrativo: [https://youtu.be/NIVZ0wv\\_iws](https://youtu.be/NIVZ0wv_iws)



## Questão 5:



- **GitHub**

Link do repositório => <https://github.com/raphaeldsb/CaseTON>

- **Observações**

Um dos maiores desafios que tive para desenvolver e entregar este projeto, foi a questão “Tempo”, devido ao meu emprego atual consumir grande parte do meu dia... Utilizei o restante das horas livres do dia e boa parte da madrugada... rs

Sobre a questão 3, pensei em utilizar alguns algoritmos de clusterização de dados, como por exemplo o K-means, mas como nunca tinha trabalhado com esse tipo de algoritmo e após perder algumas horas estudando sobre, achei melhor dar sequência com outra abordagem, para conseguir concluir o projeto a tempo.

Aproveito para agradecer a oportunidade de mostrar um pouco do meu conhecimento e reafirmo a minha vontade de entrar para o time do Ton.

Obrigado!!!!!!!!!!!!!!!

Ahhh,  
Já ia esquecendo..

**Se liga no Ton! Se liga no Ton!!** 🎵 🎵

Rio de Janeiro, 19 de novembro de 2020

*Raphael da Silva Barradas*