

TOTI Diversidade
Curso: Análise de Dados

Projeto Final
Apresentação técnica

Grupo 1:

Ambar Pereira Urrieta

Luz Santiago Altahona

Wendell Rojas Camejo

Xavier Mata Ibarra

Junho, 2022

Projeto Final

Turma 14 Grupo 1

Levantamento de requerimentos

A proposta do trabalho é desenvolver um dashboard do Power BI sobre o estado atual de um hospital fictício. Fiquem à vontade para escolher o nome do seu hospital.

Instruções

1. Siga os códigos presentes em: [SQL Exercises/The Hospital - Wikibooks, open books for an open world](#) para criar as tabelas e inserir os dados.
2. Modele e organize os dados utilizando SQL.
3. Exporte seus dados para um arquivo .csv.
4. Importe o arquivo no colab e trate os dados utilizando Python.
5. Comente os passos do seu código.
6. Crie um relatório para apresentar os resultados utilizando o Power BI.

Analise de dados

- Entender o modelo lógico oferecido no site
- Quantidade de tabelas
- Tipos de dados
- Tipo de relacionamento entre as tabelas

A estrutura do modelo representa o armazenamento dos dados em um hospital, especificamente nos departamentos de medicina geral, cirurgia e psiquiatria. Conta com 4 enfermeiras e 9 médicos. Realiza 7 procedimentos cirúrgicos, consultas de medicina geral e psiquiátrica. O hospital conta com 4 andares, cada andar com 3 blocos e em cada andar temos 9 habitações distribuídas entre os blocos. O total de habitações é 36.

O número de pacientes é crescente, mas na base de dados só tem 4 registrados.

Infraestrutura do Hospital	Ações	Pessoal
1- Departamento (medicina geral, Cirurgia) 2- Sala (Habitações) 3- Bloco (Bloco e andar)	4- Ficar (stay) = Ingresso 5-Passar_Com (UNDERGOES) = Intervenção 6- On_call (chamada de enfermeira ou turno) 7- Prescreve 8- Medicamentos 9-Compromisso (appointment) = avaliação ou consulta clínica 10- Procedimento (tipo de cirurgia)	11- Paciente 12- Enfermeira 13- Médico 14- Treinado_en (certificação medica) 15- Afiliado_com (Medico e departamento)

Dividimos as tabelas em três tópicos para um melhor entendimento da dinâmica, são 15 tabelas.

Os tipos de dados na estrutura do modelo lógico em sua maioria são do tipo interger, datetime, booleano e texto.

Os relacionamentos entre elas são do tipo “um para muitos”.

Durante a análise percebemos que algumas tabelas poderiam ser consideradas tabelas fatos, algumas delas são do grupo ações (tabelas Prescreve, Passar_Com (Intervenção), Avaliação (Appointment) e Stay (fica)). Sendo elas as que sempre vão a ter atualizações constantes, usando chaves estrangeiras para usar as informações das tabelas dimensões que contém dados fixos (tabelas Sala, bloco, departamento, Médico, Enfermeira, procedimento e medicamentos, on_Call, Treinado_em, Afiliado_com).

Projeto

Nome do Hospital: Hospital do Sul Humberto Fernández Morán.

O nome é do médico e cientista Venezuelano Humberto Fernandez-Morán, que em 1967 recebeu o prêmio Vovain por sua invenção, o bisturi com ponta de diamante, foi nomeado ao prêmio Nobel pela mesma invenção.

Objetivos do projeto

Objetivo principal

Realizar relatório do funcionamento e desempenho no segundo trimestre de 2021 do Hospital do Sul Humberto Fernández-Morán nos departamentos de medicina geral, cirurgia e psiquiatria.

Objetivos secundários

Avaliar:

- Capacidade do Hospital em infraestrutura
- Rendimento dos funcionários
- Total de pacientes ingresados (hospitalização)
- Total de procedimentos cirúrgicos realizados
- Quantidade de pacientes atendidos por mês (hospitalizados e consultas)
- Medicamentos prescritos por mês
- Faturamento e Lucro por procedimentos cirúrgicos realizados
- Comissão para médicos e enfermeiras por procedimento atendido

Dados novos e criação das tabelas

Usamos os códigos do link indicado para realizar o projeto. Deixamos as tabelas preexistentes, analisamos seus relacionamentos, dinâmica e fluxo de dados.

Dentro das tabelas modificamos os nomes dos medicamentos e nome dos procedimentos.

Inserimos um total de 150 pacientes, 50 medicamentos e mais 3 enfermeiras.

Dentro das tabelas Appointment (avaliações), Prescrive, Undergoes (Intervenção) e Stay (Hospitalização) inserimos dados distribuídos com datas referentes ao segundo trimestre do ano 2008.

Para cumprir nossos objetivos decidimos criar a tabela Salary (salários) para fazer o cálculo de quanto ganha um médico e uma enfermeira por procedimento atendido, e saber quanto fica de lucro para o Hospital.

Códigos SQL

Ferramenta usada: Microsoft SQL server Management Studio 2018

Criamos nossa base de dado:

CREATE DATABASE Hospital_HFM;

Usamos os códigos do link indicado para realizar o projeto, para criar as tabelas e inserir os dados iniciais. Depois inserimos os dados novos.

Dados modificados:

```
-- MEDICAMENTOS
```

```
SELECT * FROM Medication;
```

```
INSERT INTO Medication VALUES(1,'Amoxicilina','Anvisa','Antibiótico');
INSERT INTO Medication VALUES(2,'Rivotril','Foo Labs','Ansiolítico');
INSERT INTO Medication VALUES(3,'Alpram','Bar Laboratories','Ansiolítico');
INSERT INTO Medication VALUES(4,'Ciprofloxacina','Baz Industries','Antibiótico');
INSERT INTO Medication VALUES(5,'Dipirona','Snafu Pharmaceuticals','Anti-inflamatório');
INSERT INTO Medication VALUES(6,'Aripiprazol','Abilify®','Anti-psicótico');
INSERT INTO Medication VALUES(7,'Risperidona','Zyprexa®','Anti-psicótico');
```

Results Messages

Code	Name	Brand	Description
1	Amoxicilina	Anvisa	Antibiótico
2	Rivotril	Foo Labs	Ansiolítico
3	Alpram	Bar Laboratories	Ansiolítico
4	Ciprofloxacina	Baz Industries	Antibiótico
5	Dipirona	Snafu Pharmaceuticals	Anti-inflamatório
6	Aripiprazol	Abilify®	Anti-psicótico
7	Risperidona	Zyprexa®	Anti-psicótico
8	Quetiapina	Seroquel®	Anti-psicótico
9	Olanzapina	Zyprexa®	Anti-psicótico
10	Ziprasidona	Geodon®	Anti-psicótico

```
-- PROCEDIMENTOS
```

```
SELECT * FROM Procedures;
```

```
INSERT INTO Procedures VALUES(1,'Apendicectomia',3000);
INSERT INTO Procedures VALUES(2,'Colostomia',5200);
INSERT INTO Procedures VALUES(3,'Blefaroplastia',6500);
INSERT INTO Procedures VALUES(4,'Rinoplastia reversa',12000);
INSERT INTO Procedures VALUES(5,'Gastrotomia',4500);
INSERT INTO Procedures VALUES(6,'Pancreatectomia',6500);
INSERT INTO Procedures VALUES(7,'Biopsia cirurgica',1500);
```

Results Messages

Code	Name	Cost
1	Apendicectomia	3000
2	Colostomia	5200
3	Blefaroplastia	6500
4	Rinoplastia reversa	12000
5	Gastrotomia	4500
6	Pancreatectomia	6500
7	Biopsia cirurgica	1500

No SQL server foi criada a nova tabela Salary:

```
CREATE TABLE Salary (
    CodeProcedure INT NOT NULL,
    SalaryPhysician REAL NOT NULL,
    SalaryNurse REAL NOT NULL,
    FOREIGN KEY("CodeProcedure") REFERENCES "Procedures"("Code")
);
```

A chave primária é CodeProcedure que faz referência à tabela Procedimentos, onde está a descrição de cada intervenção cirúrgica.

Inserimos os dados na tabela Salary:

```
INSERT INTO Salary VALUES(1,600,300);
INSERT INTO Salary VALUES(2,1040,520);
```

```
INSERT INTO Salary VALUES(3,1300,650);
INSERT INTO Salary VALUES(4,2400,1200);
INSERT INTO Salary VALUES(5,900,450);
INSERT INTO Salary VALUES(6,1300,650);
INSERT INTO Salary VALUES(7,300,150);
```

```
SELECT * FROM Salary;
```

SAIDA:

	CodeProcedure	SalaryPhysician	SalaryNurse
1	1	600	300
2	2	1040	520
3	3	1300	650
4	4	2400	1200
5	5	900	450
6	6	1300	650
7	7	300	150

Algumas tabelas que tinham colunas com datas no formato Ano- Mês- Dia, apresentaram problemas logo de ser inseridos. SQL server não reconhece esse formato, então invertia os dias e os meses. Antes de inserir os dados devíamos usar o código “`SET DATEFORMAT ymd;`” para manter o formato original. Consideramos deixar assim para depois tratar em Python o formato.

```
-- CONSULTAS ----
SELECT * from Appointment
order by Starto;

SET DATEFORMAT ymd;
INSERT INTO Appointment VALUES(13216584,100000001,101,1,'2008-04-24 10:00','2008-04-24 11:00','A');
INSERT INTO Appointment VALUES(26548913,100000002,101,2,'2008-04-24 10:00','2008-04-24 11:00','B');
INSERT INTO Appointment VALUES(36549879,100000001,102,1,'2008-04-25 10:00','2008-04-25 11:00','A');
INSERT INTO Appointment VALUES(46846589,100000004,103,4,'2008-04-25 10:00','2008-04-25 11:00','B');
INSERT INTO Appointment VALUES(59871321,100000004,103,4,'2008-04-26 10:00','2008-04-26 11:00','A');
```

AppointmentID	Patient	PrepNurse	Physician	Starto	Endo	ExaminationRoom
1313	100000034	103	2	2008-06-05 09:00:00.000	2008-06-05 10:00:00.000	C
2517	100000013	NULL	3	2008-06-05 12:00:00.000	2008-06-05 13:00:00.000	C
3241	100000013	NULL	3	2008-04-05 12:00:00.000	2008-04-05 13:00:00.000	C
5689	100000013	NULL	3	2008-05-05 12:00:00.000	2008-05-05 13:00:00.000	C
44321	100000010	102	1	2008-06-16 10:00:00.000	2008-06-16 11:00:00.000	B
68945	100000033	101	3	2008-06-15 09:00:00.000	2008-06-15 10:00:00.000	A
69288	100000038	101	4	2008-04-05 10:00:00.000	2008-04-05 11:00:00.000	C
344578	100000010	102	1	2008-05-16 10:00:00.000	2008-05-16 11:00:00.000	B
345678	100000005	101	3	2008-05-03 09:00:00.000	2008-05-11 10:00:00.000	A
569654	100000026	102	9	2008-06-19 12:00:00.000	2008-06-19 13:00:00.000	B
569854	100000006	103	9	2008-06-21 14:00:00.000	2008-06-21 15:00:00.000	A

Consultas em SQL server

```

]SELECT
pro.name AS 'Procedimento Cirúrgico', --Nesta Busqueda começamos analisando a tabela
Sum(pro.Cost) AS 'Valor Total 2do trim.'
FROM Undergoes AS un --procedimentos para conhecer os valores gerais.
JOIN Procedures AS pro ON pro.Code = un.Procedures --Logo realizei um JOIN para unir as tabelas
GROUP BY pro.Name
order by 'Valor Total 2do trim.' desc; --todas os procedimentos aplicados no trimestre

```

Saída:

Procedimento Cirúrgico	Valor Total 2do trim.
Rinoplastia reversa	168000
Gastrotomia	67500
Pancreatectomia	58500
Blefaroplastia	58500
Colostomia	46800
Apendicectomia	30000
Biopsia cirurgica	10500

```

/*Quantidade de Habitacoes*/
select count (Room.RoomNumber)from Room;

/*Quantidade de Habitaciones indisponibles para el uso*/

]select count (Room.RoomNumber)
from Room
where Unavailable = 1;

/*Descrição de Habitaciones indisponibles para el uso*/

]select Room.BlockFloor as 'Andar',Room.RoomNumber as 'Numero', Room.RoomType as 'Tipo'
from Room
where Unavailable = 1

/*Ocupação da area de internação por mes*/

]select count(Stay.StayID) as 'Pacientes Internados em abril'
from Stay
where MONTH(Stay.StayStart)= 4 ;

/*Ocupação da area de internação por mes*/

]select count(Stay.StayID) as 'Pacientes Internados em abril'
from Stay
where MONTH(Stay.StayStart)= 4 ;

]select count(Stay.StayID) as 'Pacientes Internados em Maio'
from Stay
where MONTH(Stay.StayStart)= 5 ;

]select count(Stay.StayID) as 'Pacientes Internados em Junho'
from Stay
where MONTH(Stay.StayStart)= 6 ;

```

```
-- Procedimento cirúrgico por mês com o lucro.
-- Fizemos um calculo dentro da consultas, tirando do custo do procedimento o valor de comissão que ganha o médico e a enfermeira

SELECT
  p.name as 'Procedimento Cirúrgico',
  Datename(Month, u.DateUndergoes) AS 'Mês',
  (p.cost - (s.salaryphysician + s.salarynurse)) as 'Lucro'
FROM Procedures AS p
JOIN Salary AS s ON s.CodeProcedure = p.Code
join Undergoes as u ON u.Procedures = p.Code
Order by Mês asc;
```

Saída:

Procedimento Cirúrgico	Mês	Lucro
Colostomia	Abril	3640
Blefaroplastia	Abril	4550
Rinoplastia reversa	Abril	8400
Gastrotomia	Abril	3150
Apendicectomia	Abril	2100
Biopsia cirurgica	Abril	1050
Apendicectomia	Abril	2100
Pancreatectomia	Abril	4550
Biopsia cirurgica	Abril	1050

Fizemos nossas consultas e construímos com elas tabelas com o código SELECT INTO, no SQL. Com as consultas transformadas em tabelas conseguimos exportar elas em um documento CSV para ser tratadas no Colab com Python.

```
--Cantidade de medicamento recetados

select
  medicat.Name as 'Medicamento',
  COUNT(presc.Medication) AS 'Quantidade de vezes Recetado'
into Consul_Medicamentos
from Medication as medicat
JOIN Prescribes as presc on presc.Medication = medicat.Code
Group by medicat.Name
ORDER BY COUNT (presc.Medication) DESC;

-- Informação geral das consultas realizadas no segundo trimestre

SELECT
  de.Name as 'Serviço',
  Ph.EmployeeID as 'Código do Médico',
  Ph.Name as 'Médico',
  Ap.AppointmentID as 'Código da Consulta',
  Ap.Patient as 'Código do paciente',
  Ap.PrepareNurse as 'Enfermeira',
  Ap.ExaminationRoom as 'Sala de Avaliação',
  Convert(NVARCHAR, Ap.Starto, 23) AS 'Data'
Into Consulta_geral -- Nome de minha nova tabela, vai ser criada com esta consulta
FROM Appointment as Ap
Join Physician as Ph ON Ph.EmployeeID = Ap.Physician
Join Affiliated_With as af on af.Physician = Ph.EmployeeID
join Department as de ON de.DepartmentID = af.Department
Order by Data;
```



```
--INFORMAÇÃO GERAL DE MEDICAMENTOS PRESCRITOS JUNTANDO A DESCRIÇÃO DO MEDICAMENTO, NOME DO PACIENTE, NOME DO MÉDICO E  
-- O SERVIÇO OA QUAL ESTÁ AFILIADO
```

```
SELECT  
pac.name as 'Paciente',  
ph.name as 'Médico',  
ph.Position as 'Cargo',  
de.name as 'Serviço',  
Med.Name as 'Nome do medicamento',  
Med.Description AS 'Descrição',  
datetime (mm, pre.Date) 'Data'  
into Consulta_pres ---- Nome da nova tabela  
FROM Prescribes as pre  
Join Patient as pac on pac.SSN = pre.Patient  
join Medication as med on med.Code = pre.Medication  
join Physician as ph on ph.EmployeeID = pre.Physician  
join Affiliated_With as af on af.Physician = Ph.EmployeeID  
join Department as de ON de.DepartmentID = af.Department  
order by Data asc; -- Organizamos a tabela pelo nome do médico
```

Python

Ferramenta usada: Google Colab

Dados a tratar:

- Dados nulos
- Tipo de dado
- Excluir duplicados
- Formatação das datas
- Modificar o ano das datas para 2021

Códigos comentados

```
[ ] # Importamos as bibliotecas pandas e numpy
import pandas as pd
import numpy as np

# Importamos a tabela de Appointment num dataframe, pegamos o caminho do documento no ficheiro
Appoint = pd.read_csv('/content/Appointment_python.csv')
```

```
▶ # Confirmamos as informações e o tipo de dado dentro de nossa tabela

Appoint.info()
```

```
↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 136 entries, 0 to 135
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   AppointmentID    136 non-null   int64
1   Patient          136 non-null   int64
2   PrepNurse        117 non-null   float64
3   Physician         136 non-null   int64
4   Starto           136 non-null   object
5   Endo             136 non-null   object
6   ExaminationRoom  136 non-null   object
dtypes: float64(1), int64(3), object(3)
memory usage: 7.6+ KB
```

```
▶ # Percebemos que temos dados nulos na tabela PrepNurse e pelo mesmo o tipo de dado é float.
# A coluna Starto e Endo estão como tipo de dato object. Essas colunas são do tipo DateTime.

#Vamos observar as colunas com as 5 primeiras filas, como o seguinte código:
Appoint.head()
```

```
↳
```

	AppointmentID	Patient	PrepNurse	Physician	Starto	Endo	ExaminationRoom
0	1313	100000034	103.0	2	2008-06-05 09:00:00	2008-06-05 10:00:00	C
1	2517	100000013	NaN	3	2008-06-05 12:00:00	2008-06-05 13:00:00	C
2	3241	100000013	NaN	3	2008-04-05 12:00:00	2008-04-05 13:00:00	C
3	5689	100000013	NaN	3	2008-05-05 12:00:00	2008-05-05 13:00:00	C
4	44321	100000010	102.0	1	2008-06-16 10:00:00	2008-06-16 11:00:00	B

```
[ ] # Confirmamos a quantidade de dados nulos em nossa tabela com o seguinte código:

Appoint.isnull().sum()
```

```
AppointmentID    0
Patient          0
PrepNurse        19
Physician         0
Starto           0
Endo             0
ExaminationRoom  0
dtype: int64
```

```
##Substituir os valores nulos por enfermeira de apoio '107'
Appoint.fillna('107', inplace= True)
Appoint.info()

[ ] ## Modificar o tipo de dato na Coluna PrepNurse
Appoint['PrepNurse']= Appoint['PrepNurse'].astype(np.int)
```

Vamos mudar o tipo de dado e o formato das colunas de datas

```
[4] ## Transformando a coluna Starto em tipo Datetime
AppointData= Appoint['Starto'].astype(np.datetime64)

[ ] ## Agora que já modificamos o tipo, vamos a modificar o formato e o ano para 2021
## Usando a função da.strptime e digitamos a o formato que queremos
Appoint['Starto'] = AppointData.dt.strptime('%d/%m/2021 %H:%M')

[ ]

[ ] Appoint
```

	AppointmentID	Patient	PrepNurse	Physician	Starto	Endo	ExaminationRoom
0	1313	100000034	103	2	05/06/2021 09:00	05/06/2021 10:00	C
1	2517	100000013	107	3	05/06/2021 12:00	05/06/2021 13:00	C
2	3241	100000013	107	3	05/04/2021 12:00	05/04/2021 13:00	C

As tabelas criadas por consultas no SQL server, apresentaram um problema para poder ser importadas como documento CSV dentro de Python. O programa não podia ler alguns caracteres dentro da tabela. Para solucionar isso, achamos um código que deve ser inserido dentro do caminho de importação do documento: encoding = 'latin-1'.

```
#Importando tabela Presc
Presc = pd.read_csv('/content/Consul_Pres.csv',encoding='latin-1')
Presc.head(7)
```

	Paciente	Médico	Cargo	Serviço	Nome do medicamento	Descrição	Data
0	John Smith	John Dorian	Staff Internist	General Medicine	Hidroclorotiazida	Anti-hipertensivo	Abril
1	John Smith	John Dorian	Staff Internist	General Medicine	Atenol	Anti-hipertensivo	Abril
2	Random J. Patient	John Dorian	Staff Internist	General Medicine	Alpram	Ansiolítico	Junho
3	Dennis Doe	John Dorian	Staff Internist	General Medicine	Captopril	Anti-hipertensivo	Junho
4	Vitoria Guerra	John Dorian	Staff Internist	General Medicine	Losartana	Anti-hipertensivo	Maio
5	Luis Garcia	John Dorian	Staff Internist	General Medicine	Hidroclorotiazida	Anti-hipertensivo	Maio
6	Luis Garcia	John Dorian	Staff Internist	General Medicine	?NovoRapid FlexPen	Antidiabético	Junho

```
#Procurando dados nulos.
Presc.isnull().values.any()

False

[ ] #Percebemos que temos dados erroneos na tabela Presc,usamos o *inplace* = true para alterar o dataframe original.*
Presc.replace(['?NovoRapid FlexPen'],'NovoRapid FlexPen',inplace=True)
Presc
```

```
[ ] #para verificar o tipo de dados
Presc_pro.dtypes
```

```
Procedimento    object
Data            object
Medicamento     object
Nome do Médico  object
Nome do Paciente object
dtype: object
```

```
▶ #verificando mudança de tipo de dado
Presc_proData=Presc_pro['Data'].astype(np.datetime64)
Presc_proData
```

```
0    2008-05-26
1    2008-05-26
2    2008-05-26
3    2008-05-26
4    2008-05-26
...
577  2008-06-02
578  2008-06-02
579  2008-06-02
580  2008-06-02
581  2008-06-02
Name: Data, Length: 582, dtype: datetime64[ns]
```

```
[ ] #codigo utilizado para modificar a formato data, por só mês.
Presc_pro['Data']= Presc_proData.dt.month_name()
```

Saída:

	Procedimento	Data	Medicamento	Nome do Médico	Nome do Paciente
0	Rinoplastia reversa	May	Lidocaina	John Wen	Vitoria Guerra
1	Rinoplastia reversa	May	Morfina	John Wen	Mauro Morelli
2	Rinoplastia reversa	May	Fentanil	John Wen	Maria Cristina
3	Rinoplastia reversa	May	Lidocaina	John Wen	George Queiroz
4	Rinoplastia reversa	May	Risperidona	John Wen	Souza Rosa

```
▶ #Para corrigir os dados duplicados se utilizo a seguinte função
Presc_pro = Presc_pro.drop_duplicates()
Presc_pro
```

	Procedimento	Data	Medicamento	Nome do Médico	Nome do Paciente
0	Rinoplastia reversa	May	Lidocaina	John Wen	Vitoria Guerra
1	Rinoplastia reversa	May	Morfina	John Wen	Mauro Morelli
2	Rinoplastia reversa	May	Fentanil	John Wen	Maria Cristina
3	Rinoplastia reversa	May	Lidocaina	John Wen	George Queiroz
4	Rinoplastia reversa	May	Risperidona	John Wen	Souza Rosa
...
577	Apendicectomia	June	Fentanil	John Wen	Maria Cristina
578	Apendicectomia	June	Lidocaina	John Wen	George Queiroz
579	Apendicectomia	June	Risperidona	John Wen	Souza Rosa
580	Apendicectomia	June	Morfina	John Wen	Acácio Campos
581	Apendicectomia	June	Lidocaina	John Wen	Murilo Grossi

360 rows × 5 columns

Modificações de nossa tabela que foi criada de uma consulta no SQL server

```
#Observa-se o dataframe da tabela Comissão Médica |
SalarioMedico = p.read_csv('/SalarioMedico.csv',encoding='latin-1')
SalarioMedico
```

	Código Hospitalização	Procedimento	Mês	Salario Medico	Nome do Cirurgião
0	3234	4	Maio	2400	John Wen
1	3234	4	Maio	2400	John Wen
2	3236	1	Junho	600	Todd Quinlan
3	3236	1	Junho	600	Todd Quinlan
4	3237	7	Junho	300	Todd Quinlan
...
141	3251	4	Junho	2400	Christopher Turk
142	3232	5	Maio	900	Christopher Turk
143	3232	5	Maio	900	Christopher Turk
144	3238	1	Junho	600	John Wen
145	3238	1	Junho	600	John Wen

146 rows x 5 columns

```
#Tinha o Código de Hospitalização duplicado e se usou o código "Drop_duplicates"
#para eliminar todos os dados duplicados
SalarioCirugiao = SalarioMedico.drop_duplicates()
SalarioCirugiao
```

	Código Hospitalização	Procedimento	Mês	Salario Medico	Nome do Cirurgião
0	3234	4	Maio	2400	John Wen
2	3236	1	Junho	600	Todd Quinlan
4	3237	7	Junho	300	Todd Quinlan
6	3188	2	Abril	1040	Christopher Turk
8	3222	4	Maio	2400	John Wen
...
136	3246	5	Junho	900	Christopher Turk
138	3247	5	Junho	900	John Wen
140	3251	4	Junho	2400	Christopher Turk
142	3232	5	Maio	900	Christopher Turk
144	3238	1	Junho	600	John Wen

73 rows x 5 columns

A maioria de nossas tabelas precisavam dos mesmos códigos para tratar os dados.

Para exportar as tabelas tratadas usamos o código:

```
# Agora vou exportar minha tabela tratada
Appoint.to_csv('Appoint_BI.csv')
```

```
[ ] #Exportando tabela a csv
Presc_pro.to_csv('Prescpro_BI1.csv')
```

Com todas nossas tabelas prontas, importamos o documento no POWER BI, para começar a fazer nosso Dashboard.

Power BI

Ferramenta usada: Power BI Desktop e Canva.

Fizemos modificações finais na query do Power BI para facilitar o visual de nossas gráficas. Modificamos nomes que estavam em inglês, tipos de dados das colunas, os valores de dinheiro em reais, e criamos algumas colunas de data, só com o mês e o número do mês para poder organizar melhor a ordem cronológica dos meses no visual das gráficas.

O template foi criado na aplicação Canva.

O Dashboard será usado na apresentação e entregue em formato PDF.