



# Extreme Digital Solutions

## Avaliação Data Engineer

Versão 2.0



A **EDS** é uma empresa brasileira de inovação e tecnologia que tem como missão tornar negócios mais simples e eficientes, apoiando clientes na busca por resultados, criando soluções com inovação corporativa e tornando processos mais dinâmicos com qualidade e criatividade.

Como membro da equipe de **Big Data & Analytics**, você terá a oportunidade trabalhar em problemas complexos de nossos clientes. Construirá soluções inteligentes para coletar, processar e carregar grandes volumes de dados. Também deverá entender o negócio de nossos clientes, modelar métricas e insights, que serão materializados em painéis de suporte à tomada de decisões.

Para avaliarmos seus conhecimentos elaboramos um conjunto de problemas que frequentemente temos que resolver em nosso dia a dia. Lembre-se que, mais importante que a resposta certa, é ser capaz de apresentar um fluxo coerente de raciocínio e saber explicar as decisões tomadas para solucionar cada questão.

## Contexto

Nosso cliente da área de saúde gostaria de ter uma visão consolidada de seus atendimentos médicos, que hoje estão distribuídos em diversas bases de dados, sendo uma instância da base para cada hospital da rede.

Um banco de dados relacional foi configurado pela nossa equipe para funcionar como uma área de staging. Nesse banco já encontram-se os dados de todos os hospitais, organizados em diferentes schemas, da seguinte maneira:

Schema	Descrição
stg_hospital_a	Dados do hospital A
stg_hospital_b	Dados do hospital B
stg_hospital_c	Dados do hospital C

## Problema 1

Precisamos copiar os dados de pacientes dos schemas de cada hospital para o schema stg\_prontuario. Escreva um comando SQL que crie uma tabela chamada PACIENTE, no schema stg\_prontuario, para abrigar os dados de pacientes de todos os hospitais. A tabela PACIENTE tem a seguinte definição no schema de todos os hospitais:

PACIENTE		
Coluna	Tipo	Descrição
id	int	Identificador único sequencial.
nome	varchar	Nome do paciente
dt_nascimento	date	Data de nascimento do paciente
cpf	int	CPF do paciente.
nome_mae	varchar	Nome da mãe do paciente.
dt_atualizacao	timestamp	Date e hora de atualização do registro.

## Problema 2

Escreva um comando SQL que possa ser utilizado para copiar os dados dos schemas dos hospitais para a nova tabela de pacientes que você criou no schema stg\_prontuario.

### Problema 3

Escreva uma consulta em SQL que retorne os pacientes duplicados na tabela PACIENTE do schema stg\_prontuario, caso existam.

Por exemplo, para:

PACIENTE					
id	nome	dt_nascimento	cpf	nome_mae	dt_atualizacao
1	Maria Clara	01/04/1991	11111111111	Paula	27/07/2023 13:40:21
2	Maria C.	01/04/1991	11111111111	Paula	02/05/2023 18:45:27
3	Joao A.	01/05/1984	22222222222	Pedro	21/07/2023 14:50:31
4	Joao Augusto	01/05/1984	22222222222	Pedro	15/01/2023 16:45:22

Sua consulta deve retornar:

Cpf	quantidade
11111111111	2
22222222222	2

## Problema 4

Supondo que você identificou duplicatas no problema anterior, escreva uma consulta em SQL que retorne, para cada conjunto de pacientes repetidos, somente o que tem a data de atualização mais recente.

Por exemplo, para:

PACIENTE					
id	nome	dt_nascimento	cpf	nome_mae	dt_atualizacao
1	Maria Clara	01/04/1991	11111111111	Paula	27/07/2023 13:40:21
2	Maria C.	01/04/1991	11111111111	Paula	02/05/2023 18:45:27
3	Joao A.	01/05/1984	22222222222	Pedro	21/07/2023 14:50:31
4	Joao Augusto	01/05/1984	22222222222	Pedro	15/01/2023 16:45:22

Sua consulta deve retornar:

id	nome	dt_nascimento	cpf	nome_mae	dt_atualizacao
1	Maria Clara	01/04/1991	11111111111	Paula	27/07/2023 13:40:21
3	Joao A.	01/05/1984	22222222222	Pedro	21/07/2023 14:50:31

## Problema 5

Agora vamos precisar importar para nosso schema de staging todos os procedimentos médicos cadastrados no SIGTAP (Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS). Para tal, escreva um código em Python que leia os arquivos da base SIGTAP e grave seu conteúdo em uma tabela do nosso banco de dados. Os arquivos podem ser entrados [aqui](#).

## Problema 6

Escreva um programa python que leia da [API Open Meteo](#) a previsão da pressão atmosférica (hPa) na cidade do Rio de Janeiro, para os próximos 7 dias. O resultado deve ser armazenado em uma tabela de banco de dados com as seguintes colunas:

previsao_pressao_atm	
id	Integer
momento	Timestamp
valor	Float

## Problema 7

Dado que um atendimento médico pode ter vários exames solicitados, como você modelaria essas duas tabelas no banco de dados de staging?

## Problema 8

Sabendo que existem três tipos de atendimentos na base de dados (I=Internação, U=Urgência e A=Ambulatório), escreva uma consulta em SQL que retorne a quantidade média de medicamentos prescritos dos atendimentos do tipo U.

Por exemplo, para:

ATENDIMENTO				
id	tp_atend	...	...	...
1	U	...	...	...
2	U	...	...	...
3	I	...	...	...
4	A	...	...	...

ATENDIMENTO_PRESCRICAO		
id	id_atend	id_prescricao
1	1	123
2	1	434
3	2	234
4	2	633
5	2	967
6	3	456
7	4	123

Sua consulta deve retornar:

2,5

## Problema 9

Seu trabalho agora é importar prescrições de medicamentos a partir de um sistema legado. Entretanto, é sabido que existe um bug nesse sistema: ele permite que medicamentos sejam prescritos independente de sua disponibilidade no estoque. Com o objetivo final de apenas importar as prescrições com todos os medicamentos no estoque, você deve escrever um programa em Python que verifique a viabilidade de uma prescrição, dado o estado do estoque naquele momento.

A entrada do seu programa deve ser duas sequências de caracteres. A primeira sequência indica quais medicamentos foram prescritos e a segunda quais medicamentos estão no estoque. Em ambas as sequências, cada letra representa um medicamento e a frequência de cada letra representa a dose desse medicamento.

### Exemplos:

Entrada: prescrição=a, estoque=b                      Saída: false

Entrada: prescrição=aa, estoque=b                      Saída: false

Entrada: prescrição=aa, estoque=aab                      Saída: true

Entrada: prescrição=aba, estoque=cbaa                      Saída: true

## Problema 10

Utilizando Python, crie uma visualização que indique a quantidade de atendimentos médicos por dia. A entrada de seu programa deve ser uma lista de datas, onde cada data representa um atendimento que ocorreu naquele dia. Cole abaixo o código fonte de seu programa e exemplos de visualizações para diferentes entradas.