

Desafio Técnico

Neste desafio será tratado um *case* de classificação de contas ilícitas, que objetiva avaliar o candidato nas principais *hard skills* necessárias no dia a dia do time.

Será fornecido o link de um arquivo .zip que ao ser descompactado apresenta um banco de dados SQLite denominado desafio-tecnico.db para o desenvolvimento das análises. **Ressalta-se que os dados presentes neste banco foram simulados e, consequentemente, não representam nenhum cliente ou pessoa real. Além disso, os insights observados nas análises não podem ser extrapolados para a base de clientes Gerencianet.**

Dados

1. accounts

Tabela que apresenta as informações cadastrais de cada conta.

Colunas: id: ID identi cador da tabela

account_number: Número da conta

birth: Data de nascimento

occupation: Tipo de negócio autodeclarado

email: E-mail da conta

address_id: ID identi cador da tabela address

created_at: Data de criação da conta

2. address

Tabela que identifica os pares de estado e cidade. Para verificar a residência de cada conta é necessário realizar o join com esta tabela. id: ID identi cador da tabela

state: Estado de residência do cliente

city: Cidade de residência do cliente

3. levels

Cada conta recebe uma classificação de acordo com a forma que utiliza a plataforma. Contas que utilizam com maior consistência ou com grande potencial podem receber uma melhor classificação (A>B>C>D). Caso identifique-se que a conta possui características suspeitas, de fraude, é atribuída a categoria F e executado o encerramento. id: ID identificador da tabela

account_number: Número da conta

level: A, B, C, D e F

created_at: Data da classificação

4. charges

Tabela apresenta as emissões de boletos realizadas pelos clientes com os respectivos status de pago ou não. id: ID identificador da tabela

account_number: Número da conta

status: Status da cobrança [paid, unpaid]

value: Valor da cobrança (em centavos)

created_at: Data de criação do boleto

5. transactions

Tabela com as transações efetivadas por cada conta, logo, caso um boleto tenha sido pago esta informação estará presente nesta tabela e na tabela charges. id: ID identificador da tabela

account_number: Número da conta

transaction_type_id: ID identificador da tabela transaction_type

value: Valor da transação (em centavos)

created_at: Data da transação

. transaction_type

Tabela que permite identificar qual o tipo de cada transação da tabela transactions. id: ID identificador da tabela

description: boleto_recebido, pix_enviado e pix_recebido

description_long: 'BOLETO RECEBIDO PELO CLIENTE', 'PIX ENVIADO PELO CLIENTE PARA UMA CONTA EXTERNA' e 'PIX RECEBIDO PELO CLIENTE'

Entrega

Espera-se do candidato o desenvolvimento dos seguintes passos:

1. Conectar ao banco e desenvolver as *queries* para obter os dados
2. Realizar análise descritiva dos dados utilizando linguagem R ou Python. Os resultados e códigos desta análise devem car salvos em um arquivo gerado via Rmarkdown ou Jupyter Notebook
3. Desenvolver um modelo, linguagem R ou Python, para classi car as contas em lícitas ou ilícitas utilizando as estratégias e métricas que julgar relevantes dado o escopo do problema.
4. Disponibilizar a classi cação obtida pelo modelo das contas presentes no banco que não possuem classi cação na tabela levels, escolhendo estre as seguintes opções:
Opção 1: Deploy de uma API que disponibiliza um endpoit GET que recebe como parâmetro na URL o número da conta e retorna o valor 0 para conta que o modelo considere lícita e 1 caso o modelo considere ilícita. A rota deve seguir a seguinte estrutura: URL/? account_number=12345

Opção 2: Gerar um csv com as colunas account_number e fraud. Sendo que a coluna fraud deve ter os seguintes valores: 0 para contas lícitas e 1 para ilícitas

O desafio deve ser entregue no espaço em branco abaixo, a partir de um link público do Google Drive com uma pasta contendo os códigos e estrutura desenvolvida.

Será avaliado na entrega a correta interpretação dos dados e do problema, os resultados obtidos pelo modelo e a utilização de boas práticas de programação e arquitetura.

Link para base de dados: <https://s3.amazonaws.com/gerencianet-pub-prod1/printscreens/2021/desafio-tecnico.zip>

Link úteis de apoio:

<https://db.rstudio.com/databases/sqlite/>

<https://docs.python.org/3/library/sqlite3.html>

<https://fastapi.tiangolo.com/deployment/>

<https://icaroagostino.github.io/post/plumber/>