



# Hackathon 2024

Estágio SantoDigital





Sejam Muito  
**Bem-Vind@s!**



# Desafio 0

Lógica de programação (Obrigatório)



# Desafio 0

## Lógica de Programação



### Exercício I

Escreva uma função que, dado um número inteiro  $n$ , retorne uma lista de  $n$  strings de tal forma que a string  $i$  contém  $i$  asteriscos. Por exemplo, para  $n=5$ , a lista retornada seria `["*", "**", "***", "****", "*****"]`.

### Exercício II

Dado um array de números inteiros, escreva uma função que retorne o par de números com a menor diferença absoluta. Se houver mais de um par com a mesma diferença, retorne todos eles em uma lista.

### Exercício III

Escreva uma função que retorne todos os subconjuntos de um conjunto de números. Por exemplo, se a entrada for `[1, 2]`, a saída deve ser `[[], [1], [2], [1, 2]]`.

*Observação: Utilize a linguagem de programação que você mais domina*

# Desafio 0

## Lógica de Programação



### Exercício II (Avançado)



Dado um array de números inteiros, escreva uma função que retorne o par de números com a menor diferença absoluta. Se houver mais de um par com a mesma diferença, retorne todos eles em uma lista. Além disso, a função deve permitir os seguintes parâmetros opcionais:

- ***allow\_duplicates*** (booleano)  
Se definido como False, os pares de números não podem conter valores duplicados.
- ***sorted\_pairs*** (booleano)  
Se definido como True, os pares no resultado devem estar ordenados em ordem crescente.
- ***unique\_pairs*** (booleano)  
Se definido como True, a função deve retornar apenas pares únicos (ou seja, (a, b) e (b, a) são considerados o mesmo par).



# Desafio 0

## Lógica de Programação



### Exercício III (Avançado)



Escreva uma função que retorne todos os subconjuntos de um conjunto de números. A função deve permitir os seguintes parâmetros opcionais:

- ***max\_size*** (inteiro): Limita o tamanho máximo dos subconjuntos.
- ***min\_size*** (inteiro): Define o tamanho mínimo dos subconjuntos.
- ***distinct\_only*** (booleano): Se definido como True, a função deve garantir que os subconjuntos não contenham elementos duplicados.
- ***sort\_subsets*** (booleano): Se definido como True, os subconjuntos e os elementos dentro dos subconjuntos devem ser retornados em ordem crescente.



# Desafio I

Engenharia de Dados





# Atenção!

Utilize o *dataset* público [AdventureWorks](#) para as próximas questões



O conjunto de dados AdventureWorks é um banco de dados de exemplo que representa uma empresa fictícia de comércio de bicicletas e acessórios, chamada Adventure Works Cycles.



A base de dados engloba diversas áreas de negócio, desde a venda de produtos até a manutenção de estoques e recursos humanos, permitindo uma ampla variedade de análises e perguntas de negócio.

As principais entidades no banco de dados AdventureWorks incluem:

- **Vendas:** Contém tabelas relacionadas a clientes, detalhes de vendas, pedidos, itens de linha de vendas, e histórico de vendas. Esta área permite análises relacionadas ao desempenho de vendas, tendências e previsão de vendas.
- **Produção:** Inclui tabelas relacionadas à produção de produtos, como lista de materiais, instruções de roteamento, e detalhes de produto. Esta seção permite análises relacionadas à gestão de estoque e produção.
- **Compras:** Contém tabelas relacionadas a fornecedores e compras de insumos. É útil para análises relacionadas à gestão de fornecedores e compras.
- **Recursos Humanos:** Inclui tabelas relacionadas aos funcionários, como detalhes do empregado e departamentos. Esta seção permite análises relacionadas à gestão de recursos humanos.



# Desafio I

## Engenharia de Dados



### Tarefa I

Faça o download/importe e carregue o dataset AdventureWorks.

- (Recomendado) Crie o Diagrama Entidade Relacionamento
- (Recomendado) Criar base SQL a partir do diagrama e arquivos csv

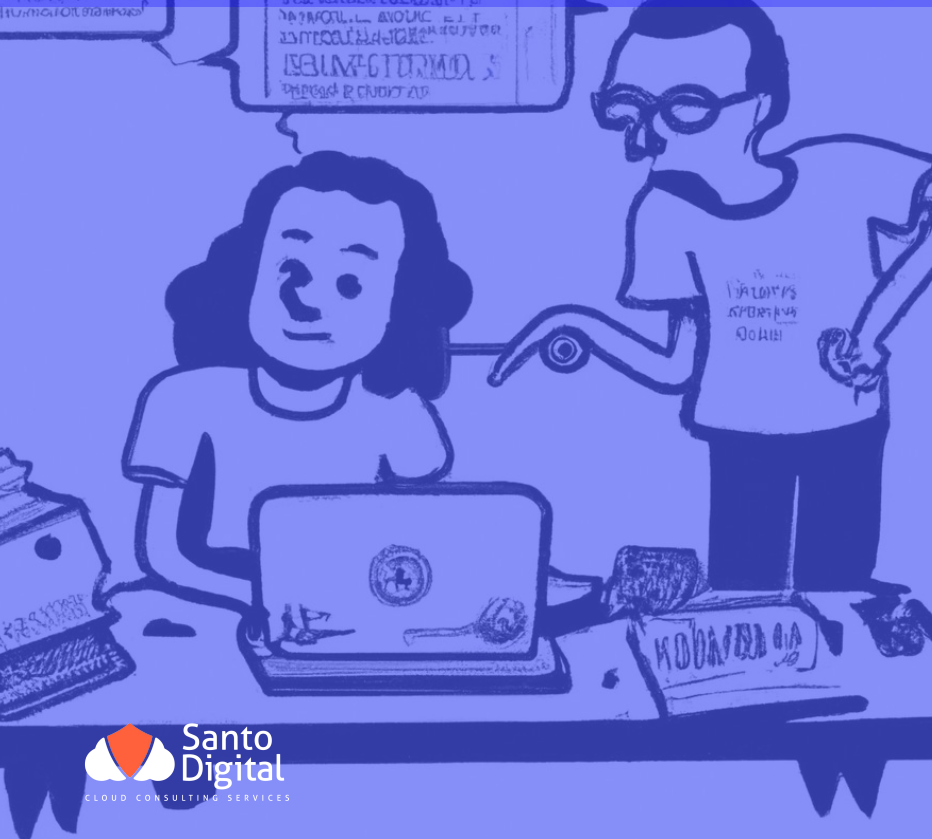
### Tarefa II

Elabore Queries SQL para responder às seguintes perguntas:

- Quais são os 10 produtos mais vendidos (em quantidade) na categoria "Bicicletas", considerando apenas vendas feitas nos últimos dois anos?
- Qual é o cliente que tem o maior número de pedidos realizados, considerando apenas clientes que fizeram pelo menos um pedido em cada trimestre do último ano fiscal?
- Em qual mês do ano ocorrem mais vendas (em valor total), considerando apenas os meses em que a receita média por venda foi superior a 500 unidades monetárias?
- Quais vendedores tiveram vendas com valor acima da média no último ano fiscal e também tiveram um crescimento de vendas superior a 10% em relação ao ano anterior?
- **Extra:** Elabore e responda uma pergunta de negócio do seu interesse que envolva a utilização de subconsultas, funções de janela ou CTEs.

# Desafio I

## Engenharia de Dados



## Tarefa Extra

Utilizando o mesmo conjunto de dados AdventureWorks, crie cinco visualizações de dados que fornecem insights sobre o desempenho da empresa.

### Visualização #1:

- Crie um gráfico de linha mostrando a tendência das vendas totais ao longo do tempo (mensal). O gráfico deve destacar os meses de pico de vendas e adicionar uma linha de tendência para prever vendas futuras.

### Visualização #2:

- Crie um gráfico de barras que apresente os 10 produtos mais vendidos na categoria de "Bicicletas". O gráfico deve mostrar a quantidade vendida de cada produto e o lucro gerado por produto.

### Visualização #3:

- Crie um mapa de calor que ilustre as vendas por região e por mês. As regiões devem estar no eixo Y, os meses no eixo X, e a intensidade da cor deve representar o volume total de vendas. Adicione a capacidade de filtrar por categorias de produtos.

### Visualização #4:

- Crie um gráfico de dispersão que mostre a relação entre o número de vendas e o valor total das vendas por cliente. Adicione uma linha de regressão para destacar a tendência.

### Visualização #5:

- Crie um gráfico de barras empilhadas que compare as vendas mensais de dois anos consecutivos. O gráfico deve permitir uma análise detalhada das tendências de crescimento ou declínio por categoria de produto.

**Observação:** As visualizações podem ser criadas utilizando bibliotecas como Matplotlib, Seaborn, Plotly, etc. Utilize o framework da sua preferência.



# Orientações - Desafio I



## Desenvolvimento da Solução

Mantenha seus artefatos organizados.

Identifique quais tecnologias você utilizou para realizar o desafio.

Explique o passo-a-passo utilizado.

## Entrega dos resultados

A Entrega deve ser realizada através de um github público com README.MD explicando o desenvolvimento da solução.

Exportar também, em CSV, os resultados de cada query.

## Interpretações

Caso necessário, explique as premissas utilizadas para solucionar o exercício e demais interpretações que achar relevante.



# Desafio II

Back-end e APIs



# Desafio II

## Back-end e APIs

Neste desafio, você deverá desenvolver uma API REST usando o framework de sua escolha (preferencialmente FastAPI) para interagir com o banco de dados AdventureWorks. Sua API deve ser capaz de executar as seguintes tarefas, além de implementar funcionalidades avançadas para garantir segurança, eficiência e manutenibilidade.

### Tarefa 1:

Funcionalidades Básicas:

- Create: Rota POST `/products/` que aceita um corpo de requisição com detalhes do produto (nome, número do produto, cor, preço padrão, etc.) e adiciona o novo produto à tabela de produtos.
- Read: Rota GET `/products/` que retorna uma lista de todos os produtos com paginação, filtragem e ordenação. Rota GET `/products/{id}` que retorna os detalhes de um único produto baseado no ID do produto.
- Update: Rota PUT `/products/{id}` que aceita um corpo de requisição com detalhes do produto e atualiza o produto com o ID correspondente.
- Delete: Rota DELETE `/products/{id}` que remove o produto com o ID correspondente da tabela de produtos

Funcionalidades avançadas:

- Autenticação e Autorização: Implementar autenticação via JWT (JSON Web Token) para garantir que apenas usuários autenticados possam acessar as rotas de criação, atualização e exclusão. Implementar autorização baseada em roles (admin, user) para que apenas usuários com a role de admin possam criar, atualizar ou excluir produtos.



# Desafio II

## Back-end e APIs

- **Validação de Dados:** Utilizar bibliotecas de validação (ex: pydantic no FastAPI) para garantir que os dados recebidos nas requisições estejam no formato correto.
- **Paginação, Filtragem e Ordenação:** Adicionar paginação nas rotas de listagem (GET /products/) para retornar um número limitado de produtos por página. Adicionar filtros para permitir busca por atributos específicos (ex: cor, preço) na rota de listagem. Implementar ordenação (ex: ordenar por preço, nome) na rota de listagem.
- **Manipulação de Transações:** Garantir que as operações de criação, atualização e exclusão sejam transacionais, ou seja, que todas as operações de banco de dados em uma única transação sejam confirmadas ou revertidas juntas.
- **Testes Unitários e de Integração:** Criar testes unitários para cada rota do CRUD para garantir que as funcionalidades básicas funcionem corretamente. Criar testes de integração para verificar se o sistema inteiro funciona conforme o esperado quando integrado com o banco de dados.
- **Criação de Logs:** Implementar um sistema de logs para monitorar as operações CRUD realizadas. Os logs devem registrar detalhes como data e hora da operação, usuário que realizou a operação, e os dados envolvidos.
- **Documentação Automática:** Utilizar as funcionalidades do FastAPI para gerar documentação automática da API utilizando Swagger ou Redoc.



# Desafio II

## Back-end e APIs

### Tarefa 2:

Neste desafio, você deverá desenvolver testes unitários e de integração para as rotas da API REST que você desenvolveu anteriormente. O objetivo é garantir que todas as funcionalidades da API estejam funcionando corretamente e que todos os casos de erro sejam tratados de forma adequada. Você deve usar o framework de testes PyTest. Mostrar o percentual de cobertura de testes será um diferencial. Siga os requisitos técnicos abaixo.

#### Requisitos Técnicos:

- **Configuração do Ambiente de Testes:** Configurar um ambiente de testes isolado utilizando um banco de dados SQLite em memória. Utilizar pytest como framework de testes.
- **Testes Unitários:** Criar testes unitários para cada rota da API (POST /products/, GET /products/, GET /products/{id}, PUT /products/{id}, DELETE /products/{id}). Verificar se cada rota está funcionando corretamente com entradas válidas. Verificar se cada rota lida corretamente com entradas inválidas e retorna os erros esperados.
- **Testes de Integração:** Criar testes de integração para verificar a interação entre diferentes componentes da API e o banco de dados. Verificar se as transações estão sendo realizadas corretamente (ex: criação seguida de leitura para verificar persistência).
- **Cobertura de Testes:** Gerar relatórios de cobertura de testes utilizando pytest-cov. Mostrar o percentual de cobertura de testes.

# Desafio II

## Back-end e APIs



### Tarefa 3:

Neste desafio, você deverá desenvolver novas rotas na API REST para responder às perguntas de negócio previamente definidas. Essas rotas irão recuperar e processar dados do banco de dados AdventureWorks e retornar as respostas em formato JSON. Além disso, você deve documentar todas as rotas utilizando a documentação do Swagger (ou similar) para que os usuários da API possam entender facilmente como utilizá-la. Siga os requisitos técnicos abaixo.

#### Requisitos Técnicos:

- **Configuração da API:** Utilizar FastAPI para criar as novas rotas. Configurar o Swagger para a documentação automática das rotas.
- **Rotas de Perguntas de Negócio:**
  - Rota 1:** GET /sales/top-products/category/{category}  
Retorna os 10 produtos mais vendidos (em quantidade) na categoria fornecida.
  - Rota 2:** GET /sales/best-customer  
Retorna o cliente com o maior número de pedidos realizados.
  - Rota 3:** GET /sales/busiest-month  
Retorna o mês com mais vendas (em valor total).
  - Rota 4:** GET /sales/top-sellers  
Retorna os vendedores que tiveram vendas com valor acima da média no último ano fiscal.

# Desafio II

## Back-end e APIs

- **Documentação:** Documentar todas as rotas utilizando o Swagger, incluindo exemplos de requisições e respostas. Fornecer descrições claras e detalhadas para cada parâmetro e campo de resposta.
- **Testes Unitários e de Integração:** Criar testes unitários e de integração para cada uma das novas rotas, garantindo que as respostas estejam corretas e que todos os casos de erro sejam tratados adequadamente. Utilizar pytest para os testes e pytest-cov para gerar relatórios de cobertura.



# Orientações - Desafio II



## Desenvolvimento da Solução

Mantenha seus artefatos organizados.

Identifique quais tecnologias você utilizou para realizar o desafio.

Explique o passo-a-passo utilizado.

Descreva os comandos necessários para executar a API.

## Entrega dos resultados

A Entrega deve ser realizada através de um github público com README.MD explicando o desenvolvimento da solução, como subir a API e testar.

Será um diferencial se a API subir em Containers Docker ou similares.

## Interpretações

Caso necessário, explique as premissas utilizadas para solucionar o exercício e demais interpretações que achar relevante.





# Desafio III

Metodologias Ágeis e Gerenciamento de Projetos



# Desafio III

## Metodologias Ágeis



### Considere o seguinte cenário:

Seu time de desenvolvimento vai criar uma aplicação web de análise de dados para os stakeholders da empresa fictícia AdventureWorks.

Esta aplicação deve permitir que os usuários visualizem e interajam com os dados de vendas da empresa de maneira intuitiva e informativa.

### Requisitos de negócio:

- **Visualização das Vendas Totais ao Longo do Tempo:** A aplicação deve permitir aos usuários visualizar as vendas totais ao longo do tempo com granularidade ajustável (diária, semanal, mensal, trimestral, anual). Os usuários devem poder aplicar filtros avançados para visualizar vendas por diferentes segmentos, como categorias de produtos, regiões e canais de venda. A visualização deve incluir a capacidade de identificar e destacar tendências e anomalias usando técnicas de machine learning, como a detecção de outliers e análise de séries temporais.
- **Produtos Mais Vendidos em Cada Categoria:** Os usuários devem ser capazes de ver os produtos mais vendidos em cada categoria, com opções para visualizar métricas adicionais como receita gerada, margem de lucro, e número de clientes únicos. A aplicação deve permitir a comparação entre diferentes categorias e produtos, oferecendo visualizações dinâmicas que ajustam conforme os filtros aplicados. Deve ser possível exportar os relatórios em formatos como CSV, Excel e PDF, e agendar o envio automático desses relatórios por email.
- **Desempenho de Vendas de Cada Vendedor:** A aplicação deve permitir aos usuários ver o desempenho de vendas de cada vendedor, com a possibilidade de visualizar KPIs como a taxa de conversão, valor médio do pedido, e taxa de retenção de clientes. Deve ser possível visualizar o desempenho em diferentes períodos e comparar o desempenho de diferentes vendedores em gráficos lado a lado. Os usuários devem ter acesso a uma análise detalhada do funil de vendas de cada vendedor, desde o primeiro contato até o fechamento da venda, incluindo o tempo médio para conversão e os principais gargalos.

(Continua na próxima página)



# Desafio III

## Metodologias Ágeis



### Requisitos de negócio:

- **Vendas por Região:** A aplicação deve permitir aos usuários visualizar as vendas por região, com mapas interativos que mostram não apenas o volume de vendas, mas também métricas como a densidade de vendas, receita gerada e crescimento percentual em diferentes regiões. Deve ser possível segmentar as visualizações por outros critérios, como categorias de produtos e períodos de tempo, para identificar padrões regionais específicos. A aplicação deve suportar previsões de vendas por região, utilizando algoritmos de machine learning para projetar tendências futuras com base nos dados históricos.
- **CRUD para Registros de Produtos:** A aplicação deve permitir aos usuários criar, atualizar e deletar registros de produtos, com validação robusta dos dados de entrada e controle de versões para garantir a integridade dos dados. As operações CRUD devem ser auditadas, registrando informações como quem fez a alteração, quando foi feita e quais foram os dados alterados. A aplicação deve oferecer funcionalidades avançadas, como a importação em massa de produtos a partir de arquivos CSV ou Excel, e a capacidade de definir e gerenciar categorias, subcategorias e atributos personalizados para os produtos.

# Desafio III

## Metodologias Ágeis



### Tarefa:

Com base nesses requisitos de negócio, crie um backlog de histórias de usuário. Cada história de usuário deve incluir critérios de aceitação e as tarefas técnicas necessárias para completá-la. Inclua os stakeholders que julgar relevantes.

Considere a seguinte equipe:

- Desenvolvedores: Back-end, Front-end, DevOps e Eng. de Dados
- Tech Lead
- Head Data
- Product Manager
- Analista de Negócios
- Usuários Finais (Stackholders da Empresa)

[O que são as histórias de usuário e como escrevê-las bem?](#)

# Desafio III

## Metodologias Ágeis



### Exemplo:

#### História de usuário #1:

Como um gestor de vendas, quero visualizar as vendas totais ao longo do tempo para poder identificar tendências e períodos de pico.

#### Crítérios de aceitação:

- Os dados de vendas devem ser exibidos em um gráfico de linha.
- Devo ser capaz de escolher o intervalo de tempo que quero visualizar.
- Devo ser capaz de visualizar os dados por dia, semana, mês ou ano.

#### Tarefas técnicas:

- Implementar rota de API para buscar dados de vendas (Desenvolvedor Back-end).
- Implementar função para agregar dados de vendas por intervalo de tempo selecionado (Engenheiro de Dados).
- Criar interface de usuário para visualizar dados de vendas (Desenvolvedor Front-end).
- Implementar testes para a nova rota da API (Desenvolvedor Back-end).
- Configurar o ambiente de produção para suportar a nova funcionalidade (DevOps).

**Pontuação de Planning Poker: 5**

# Desafio III

## Metodologias Ágeis



### Extra:

O candidato deve organizar as histórias de usuário em sprints de duas semanas, levando em consideração a pontuação de cada história e a capacidade do time.

Como premissas adicionais, o candidato pode considerar ferramentas e tecnologias que a equipe está confortável em usar, as dependências entre as histórias de usuário, e os possíveis riscos ou desafios que podem surgir durante o desenvolvimento.



# Orientações - Desafio III



## Desenvolvimento da Solução

Mantenha seus artefatos organizados.

Organize o backlog de maneira cronológica e de maneira que facilite o entendimento.

Utilize também tabelas e diagramas que julgar necessário.

## Entrega dos resultados

A Entrega deve ser realizada através de um github público com README.MD explicando o desenvolvimento da solução.

## Interpretações

Caso necessário, explique as premissas utilizadas para solucionar o exercício e demais interpretações que achar relevante.



# Desafio IV

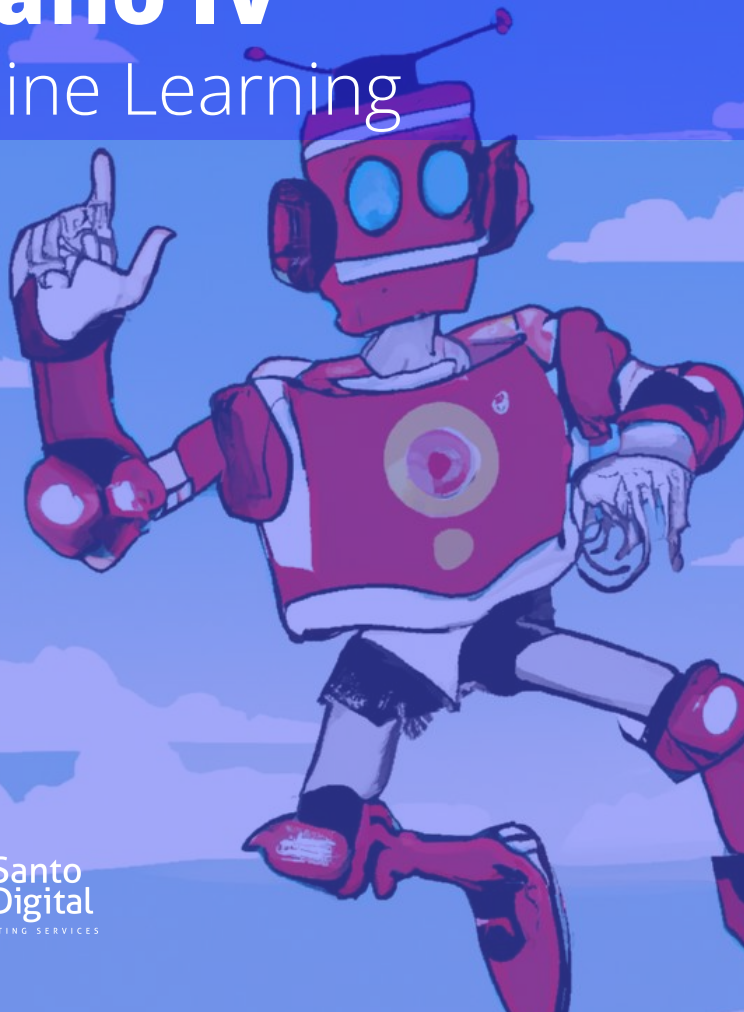
Machine Learning





# Desafio IV

## Machine Learning



### Contexto:

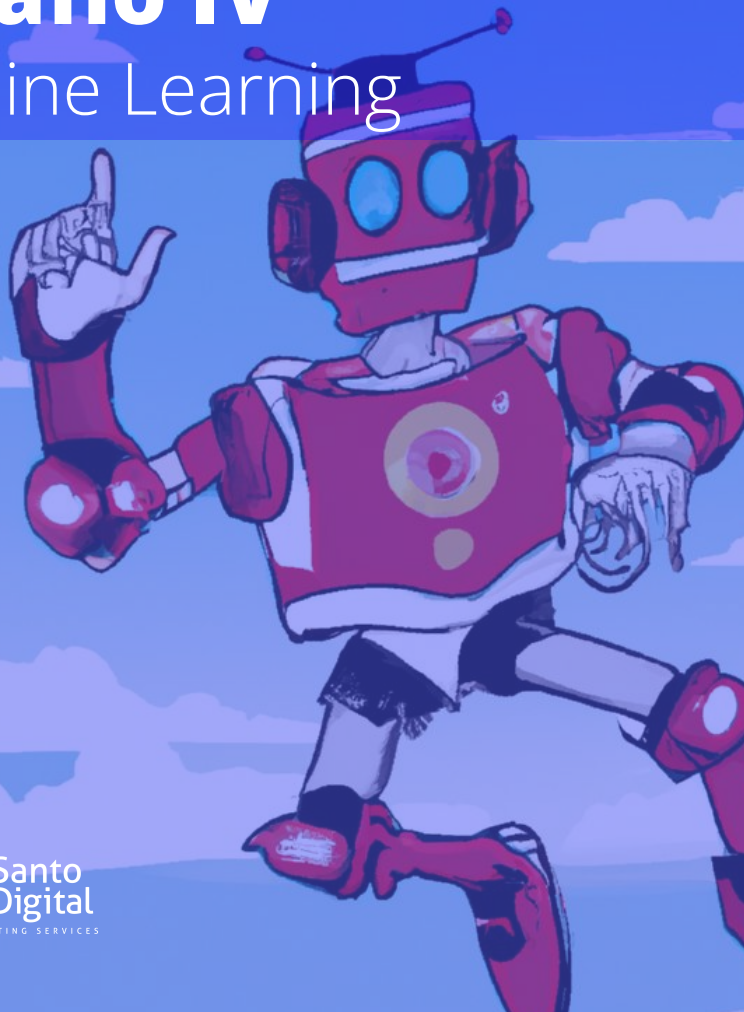
Para este desafio, você estará utilizando o conjunto de dados *AdventureWorks*. O objetivo é criar um modelo de machine learning que possa prever a demanda de vendas de um produto específico para o próximo mês.

### Parte 1: Análise Exploratória de Dados (EDA)

- Realize uma análise exploratória detalhada dos dados de vendas. Isso deve incluir: Visualizações gráficas avançadas utilizando diferentes tipos de gráficos (como histogramas, gráficos de dispersão, boxplots, etc.) para explorar a distribuição das vendas ao longo do tempo, por região, e por categoria de produto.
- Análise de estatísticas descritivas não apenas para vendas totais, mas também para segmentações específicas dos dados, como vendas por canal de distribuição ou por tipo de cliente.
- Identificação e tratamento de outliers de forma aprofundada, utilizando métodos estatísticos robustos e visualizações detalhadas para entender o impacto desses outliers nas análises subsequentes.

# Desafio IV

## Machine Learning



### Parte 2: Pré-processamento de Dados

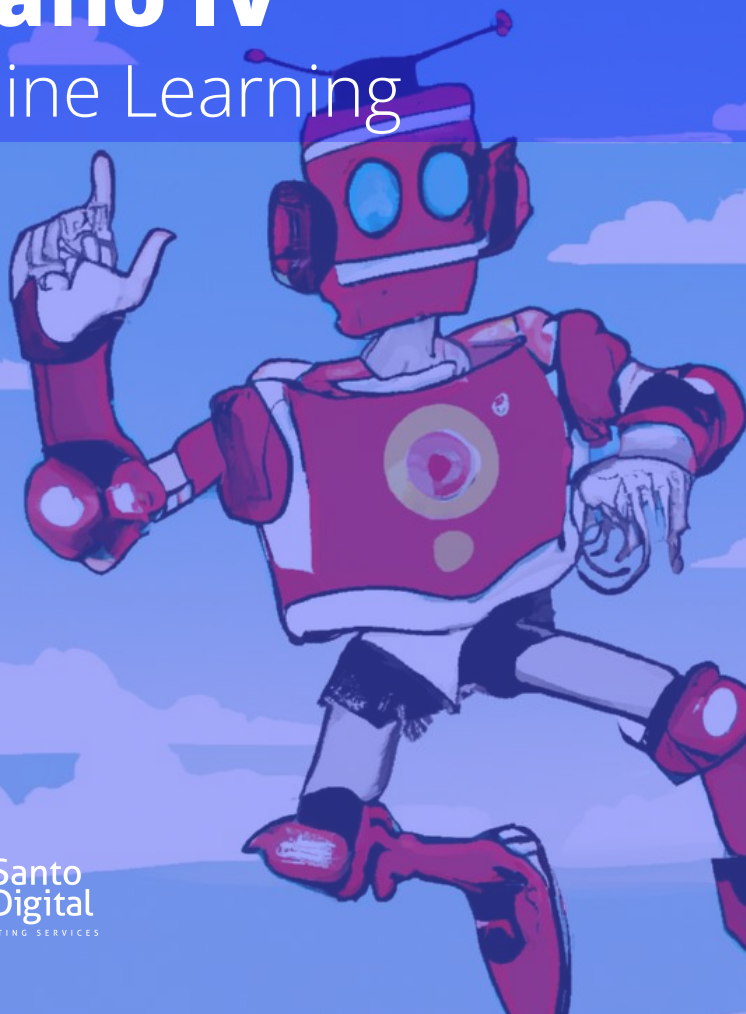
- Prepare os dados para o treinamento do modelo de machine learning, considerando: Uma abordagem abrangente de limpeza de dados que inclui detecção e tratamento de valores faltantes utilizando técnicas como imputação baseada em modelos ou exclusão seletiva de linhas.
- Tratamento de outliers de maneira mais sofisticada, utilizando métodos como transformações estatísticas ou métodos de winsorização para suavizar o impacto desses valores extremos.
- Codificação de variáveis categóricas com técnicas avançadas como codificação one-hot, codificação alvo (target encoding) considerando a relação com a variável de destino (vendas), ou uso de embeddings em modelos mais complexos.
- Normalização ou padronização das variáveis numéricas de forma a otimizar o desempenho dos algoritmos de machine learning, considerando a distribuição específica das vendas e outras características relevantes dos dados.

### Parte 3: Feature Engineering

- Identifique e desenvolva características (features) que possam ser cruciais para prever as vendas do próximo mês, utilizando: Técnicas de engenharia de features como criação de variáveis de séries temporais, como média móvel das vendas dos últimos meses, variação percentual das vendas em relação ao período anterior, ou sazonalidade ajustada para diferentes períodos do ano.
- Incorporação de dados externos relevantes, como dados meteorológicos que possam afetar o padrão de vendas.
- Criação de features interativas que capturam relações não lineares entre diferentes variáveis, utilizando transformações polinomiais ou interações de características.

# Desafio IV

## Machine Learning



### Parte 4: Modelagem

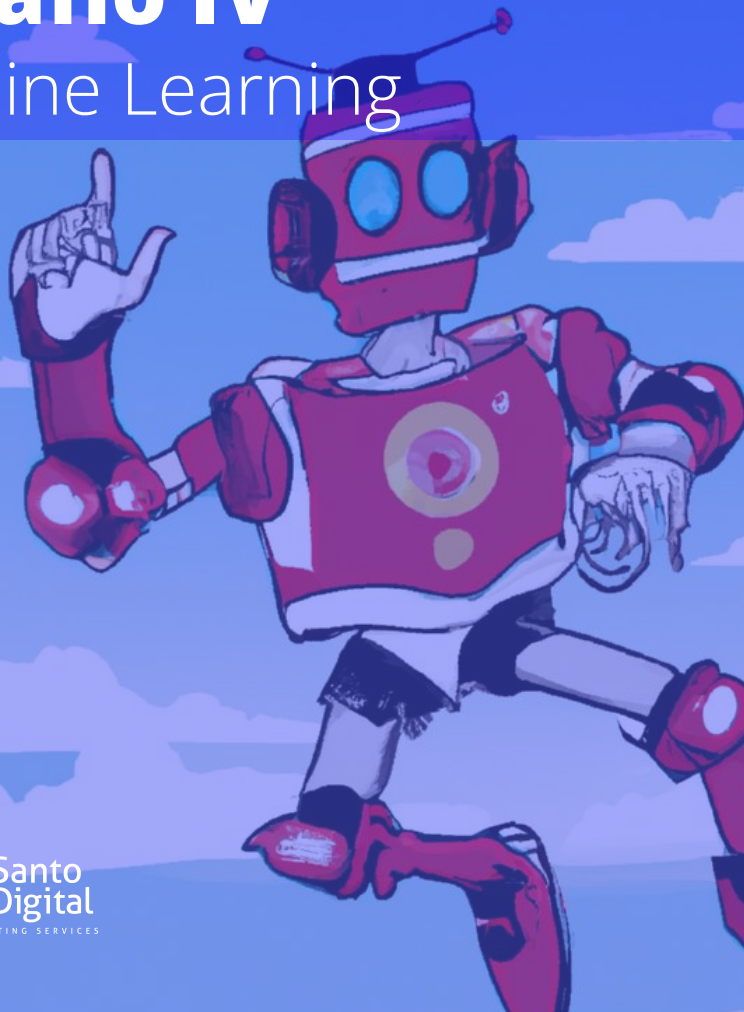
- Desenvolva e treine um modelo de machine learning para prever as vendas do próximo mês.
- Justifique sua escolha de algoritmo considerando: Uma revisão detalhada da literatura atual sobre modelos de previsão de vendas, destacando as vantagens e desvantagens dos principais algoritmos utilizados, como modelos lineares, árvores de decisão, redes neurais, e métodos ensemble.
- Seleção do algoritmo mais apropriado com base nas características específicas dos dados de vendas, como sazonalidade, não linearidades, e potencial para interações complexas entre variáveis.
- Utilização de métricas de avaliação de desempenho robustas, como RMSE (Root Mean Square Error) ou MAE (Mean Absolute Error), e implementação de validação cruzada estratificada para garantir a generalização do modelo.

### Parte 5: Ajuste de Modelo

- Explore maneiras de otimizar o desempenho do seu modelo, incluindo: Ajuste de hiperparâmetros através de técnicas avançadas como busca em grade combinada com otimização bayesiana, levando em consideração a sensibilidade do modelo a diferentes parâmetros como taxa de aprendizado, profundidade da árvore, e número de neurônios.
- Experimentação com diferentes algoritmos de machine learning, como gradient boosting, redes neurais profundas, ou modelos baseados em kernels, e compare suas performances utilizando validação cruzada.
- Incorporação de novas features derivadas da Parte 3 ou de insights adicionais obtidos durante a análise exploratória para enriquecer a representação dos dados e melhorar a capacidade preditiva do modelo.

# Desafio IV

## Machine Learning



### Parte 6:

#### Interpretação do Modelo

- Realize uma análise aprofundada dos resultados do seu modelo, considerando: Identificação das características mais importantes para a previsão de vendas através de técnicas de interpretabilidade de modelos, como importância de features ou perfis de variáveis
- Avaliação da coerência intuitiva do modelo em relação aos padrões de vendas observados, considerando aspectos sazonais, efeitos promocionais, e impactos econômicos externos.
- Discussão sobre eventuais limitações do modelo, como vieses de seleção de dados, efeitos não capturados por variáveis disponíveis, ou limitações intrínsecas dos algoritmos escolhidos para lidar com complexidades específicas dos dados de vendas.



# Orientações - Desafio IV



## Desenvolvimento da Solução

Mantenha seus artefatos organizados.

Utilize também tabelas e diagramas que julgar necessário.

## Entrega dos resultados

Para este desafio, você deve submeter um notebook Jupyter (ou similar) que mostra todo o seu processo de trabalho, desde a análise exploratória de dados até a interpretação do modelo.

Espera-se que o notebook seja bem documentado, com explicações sobre sua abordagem e pensamento em cada passo.

## Interpretações

Caso necessário, explique as premissas utilizadas para solucionar o exercício e demais interpretações que achar relevante.

Lembre-se, o objetivo não é apenas criar o modelo de machine learning mais preciso, mas também entender os dados, fazer suposições razoáveis, e criar um modelo que possa ser interpretado e usado por outros na empresa.



# Desafio V

Infraestrutura Cloud





# Desafio V

## Infraestrutura Cloud

### Contexto:

Seu desafio é criar um conjunto de arquivos de configuração Terraform que provisione a infraestrutura descrita abaixo. Além disso, você deve fornecer um documento explicativo que justifique suas escolhas de configuração e detalhe o processo de aplicação da infraestrutura.

### Automação de Infraestrutura com Terraform

- Crie um conjunto de arquivos de configuração Terraform para provisionar uma infraestrutura básica no GCP, incluindo uma rede VPC, duas instâncias de VM, e um balanceador de carga HTTP(S).
  - Crie uma rede VPC chamada "hackathon-vpc" com duas sub-redes.
    - subnet1: 192.168.1.0/24 (us-central1)
    - subnet2: 10.152.0/24 (us-east1)
  - Crie 01 instâncias de VM dentro de cada Subnet da VPC criada.
    - Chamadas "vm-instance-01" e "vm-instance-02"
    - Instale o servidor web Apache em ambas as instâncias usando scripts de inicialização.
  - Configure um balanceador de carga HTTP(S) para distribuir o tráfego entre as duas instâncias de VM.
    - Crie um serviço de backend e um health check para garantir a disponibilidade das VMs.
  - Configure regras de firewall para permitir o tráfego HTTP (porta 80) para as VMs e o balanceador de carga.

# Desafio V

## Infraestrutura Cloud

- Inclua instruções detalhadas sobre como aplicar os arquivos Terraform e como verificar se os recursos foram criados corretamente.
  - Documente todos os passos necessários para aplicar os arquivos Terraform.
  - Justifique suas escolhas de configuração, incluindo tipo de instância, sub-redes e regras de firewall.



# Orientações - Desafio V



## Desenvolvimento da Solução

Mantenha seus artefatos organizados.

## Entrega dos resultados

A Entrega deve ser realizada através de um github público com README.MD explicando o desenvolvimento da solução.



# Submissão





# Submissão | GitHub



Enviar os links do github com as respostas dos desafios via Google Forms:

<https://forms.gle/zBqnc3565bHhZ5P9>

## Deadline

- **01/09/2024 - Domingo, 23h55**
- Crie sua conta no github e suba seu código em um repositório público.
  - <https://github.com/>





# Boa sorte!

