

# Projeto: Análise de Dados para AdventureWorks

**Desenvolvido por:** Arthur Curi Kramberger

**Data:** 01/09/2024

## Hackathon SantoDigital

---

### Índice

1. Introdução
2. Histórias de Usuário
  - 2.1. História de Usuário #1: Visualização das Vendas Totais ao Longo do Tempo
  - 2.2. História de Usuário #2: Produtos Mais Vendidos em Cada Categoria
  - 2.3. História de Usuário #3: Desempenho de Vendas de Cada Vendedor
  - 2.4. História de Usuário #4: Vendas por Região
  - 2.5. História de Usuário #5: CRUD para Registros de Produtos
  - 2.6. História de Usuário #6: Agendamento de Envio Automático de Relatórios
3. Organização do Backlog
4. Desenvolvimento da Solução
5. Entrega dos Resultados
6. Interpretações
7. Conclusão

## 1. Introdução

Este documento descreve o planejamento e desenvolvimento de uma aplicação web de análise de dados para a empresa fictícia AdventureWorks. O objetivo é proporcionar aos stakeholders uma ferramenta intuitiva e informativa para visualização e interação com os dados de vendas da empresa, atendendo aos requisitos de negócio descritos.

## 2. Histórias de Usuário

### 2.1. História de Usuário #1: Visualização das Vendas Totais ao Longo do Tempo

- **Descrição:** Como um gestor de vendas, quero visualizar as vendas totais ao longo do tempo para poder identificar tendências e períodos de pico.
- **Critérios de Aceitação:**
  - Os dados de vendas devem ser exibidos em um gráfico de linha.
  - O usuário deve ser capaz de escolher o intervalo de tempo (diário, semanal, mensal, trimestral, anual).
  - O usuário deve ser capaz de aplicar filtros avançados para visualizar vendas por diferentes segmentos (categorias de produtos, regiões, canais de venda).
  - A visualização deve incluir a capacidade de identificar tendências e anomalias.
- **Tarefas Técnicas:**
  - Implementar rota de API para buscar dados de vendas (Back-end).
  - Implementar função para agregar dados de vendas por intervalo de tempo selecionado (Engenheiro de Dados).
  - Implementar algoritmo de detecção de outliers e análise de séries temporais (Engenheiro de Dados).
  - Criar interface de usuário para visualizar dados de vendas com gráficos dinâmicos (Front-end).
  - Implementar testes para a nova rota da API (Back-end).
  - Configurar o ambiente de produção para suportar a nova funcionalidade (DevOps).

### 2.2. História de Usuário #2: Produtos Mais Vendidos em Cada Categoria

- **Descrição:** Como um gerente de categoria, quero ver os produtos mais vendidos em cada categoria para poder tomar decisões sobre estoque e marketing.
- **Critérios de Aceitação:**
  - O usuário deve ser capaz de visualizar os produtos mais vendidos em cada categoria.

- O usuário deve poder 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.
- **Tarefas Técnicas:**
  - Criar rota de API para obter os produtos mais vendidos por categoria (Back-end).
  - Implementar lógica de comparação e filtro para diferentes métricas (Engenheiro de Dados).
  - Desenvolver interface de usuário para visualização dos produtos mais vendidos e suas métricas (Front-end).
  - Implementar exportação de relatórios em CSV, Excel e PDF (Front-end).
  - Implementar testes unitários e de integração para a nova funcionalidade (Back-end).

### 2.3. História de Usuário #3: Desempenho de Vendas de Cada Vendedor

- **Descrição:** Como um supervisor de vendas, quero visualizar o desempenho de vendas de cada vendedor para poder identificar e recompensar os melhores desempenhos.
- **Critérios de Aceitação:**
  - O usuário deve ser capaz de visualizar KPIs como taxa de conversão, valor médio do pedido, e taxa de retenção de clientes.
  - O usuário deve poder comparar o desempenho de diferentes vendedores em gráficos lado a lado.
  - O usuário deve ter acesso a uma análise detalhada do funil de vendas de cada vendedor.
- **Tarefas Técnicas:**
  - Criar rota de API para obter o desempenho de vendas por vendedor (Back-end).
  - Implementar lógica para calcular KPIs e funil de vendas (Engenheiro de Dados).
  - Desenvolver interface de usuário para exibir KPIs e comparações (Front-end).
  - Implementar testes para garantir a precisão dos cálculos de KPIs (Back-end).
  - Configurar ambiente para suportar visualizações comparativas (DevOps).

### 2.4. História de Usuário #4: Vendas por Região

- **Descrição:** Como um analista de mercado, quero visualizar as vendas por região em mapas interativos para identificar áreas de crescimento e oportunidades de mercado.
- **Critérios de Aceitação:**
  - O usuário deve ser capaz de visualizar o volume de vendas por região.

- O usuário deve poder ver métricas como densidade de vendas, receita gerada, e crescimento percentual.
- A aplicação deve suportar previsões de vendas por região utilizando algoritmos de machine learning.
- **Tarefas Técnicas:**
  - Implementar rota de API para buscar dados de vendas regionais (Back-end).
  - Desenvolver mapas interativos com visualizações dinâmicas (Front-end).
  - Implementar previsão de vendas por região com machine learning (Engenheiro de Dados).
  - Implementar testes para validação de mapas interativos e previsões (Back-end).
  - Configurar o ambiente para suportar previsões e mapas interativos (DevOps).

## 2.5. História de Usuário #5: CRUD para Registros de Produtos

- **Descrição:** Como um administrador de produtos, quero criar, atualizar e deletar registros de produtos para manter a base de dados sempre atualizada.
- **Critérios de Aceitação:**
  - O usuário deve ser capaz de realizar operações CRUD nos registros de produtos.
  - As operações CRUD devem ter validação robusta e controle de versões.
  - Deve ser possível importar em massa produtos a partir de arquivos CSV ou Excel.
- **Tarefas Técnicas:**
  - Desenvolver API para operações CRUD em produtos com validação de dados (Back-end).
  - Implementar controle de versões para as operações CRUD (Engenheiro de Dados).
  - Desenvolver interface de usuário para gerenciamento de produtos, incluindo importação em massa (Front-end).
  - Implementar funcionalidades avançadas para definição e gerenciamento de categorias e atributos (Back-end).
  - Implementar auditoria para registrar alterações nos dados (Engenheiro de Dados).
  - Implementar testes para as operações CRUD e funcionalidades avançadas (Back-end).

## 2.6. História de Usuário #6: Agendamento de Envio Automático de Relatórios

- **Descrição:** Como um analista de negócios, quero agendar o envio automático de relatórios por email para compartilhar insights regulares com stakeholders.
- **Critérios de Aceitação:**
  - O usuário deve ser capaz de agendar relatórios para envio automático por email.
  - A aplicação deve suportar exportação de relatórios em CSV, Excel, e PDF.

- Os relatórios devem ser personalizados de acordo com os filtros aplicados pelos usuários.
- **Tarefas Técnicas:**
  - Desenvolver API para agendamento e envio de relatórios por email (Back-end).
  - Implementar exportação de relatórios em múltiplos formatos (Front-end).
  - Implementar lógica para personalização dos relatórios com base em filtros (Engenheiro de Dados).
  - Implementar testes para validação do envio e personalização de relatórios (Back-end).
  - Configurar o ambiente para suportar o envio automático de relatórios (DevOps).

### 3. Organização do Backlog

O backlog do produto foi organizado cronologicamente para facilitar a implementação das funcionalidades mais prioritárias, mantendo uma ordem lógica de desenvolvimento. Abaixo segue a tabela com a organização das histórias de usuário:

ID	Nome da História	Prioridade	Sprint Planejado	Pontuação
1	Visualização das Vendas Totais ao Longo do Tempo	Alta	Sprint 1	8
2	Produtos Mais Vendidos em Cada Categoria	Alta	Sprint 1	7
3	Desempenho de Vendas de Cada Vendedor	Média	Sprint 2	9

4	Vendas por Região	Média	Sprint 2	6
5	CRUD para Registros de Produtos	Alta	Sprint 3	10
6	Agendamento de Envio Automático de Relatórios	Média	Sprint 3	4

#### 4. Desenvolvimento da Solução

Durante o desenvolvimento, os artefatos foram mantidos organizados em um repositório Git, e o backlog foi gerenciado utilizando uma ferramenta de gerenciamento de projetos (como Jira ou Trello). Para cada sprint, as tarefas foram priorizadas e atribuídas de acordo com a capacidade da equipe, garantindo a entrega incremental das funcionalidades.

Diagrama de arquitetura e tabelas de dependências foram utilizados para garantir a clareza das integrações entre diferentes componentes da

aplicação. Além disso, práticas de desenvolvimento ágil, como daily meetings e revisões de sprint, foram adotadas para monitorar o progresso e ajustar o planejamento conforme necessário.

#### 5. Entrega dos Resultados

A entrega dos resultados será realizada através de um repositório público no GitHub, com um arquivo README.md detalhando o desenvolvimento da solução, as decisões técnicas tomadas, e instruções para a execução do projeto.

Link para o repositório:

[<https://github.com/exemplo/adventureworks-analise>](<https://github.com/exemplo/adventureworks-analise>) (substitua pelo link real)

## 6. Interpretações

Durante o desenvolvimento, foram consideradas premissas como a familiaridade da equipe com as tecnologias escolhidas e a necessidade de uma interface intuitiva para os usuários finais. Decisões como a priorização de funcionalidades basearam-se em feedback dos stakeholders e na análise do impacto no negócio.

Outras interpretações relevantes incluem a escolha de algoritmos de machine learning adequados para detecção de outliers e previsões, bem como a implementação de um sistema robusto de controle de versões para as operações CRUD.

## 7. Conclusão

O projeto de análise de dados para a AdventureWorks foi desenvolvido com foco na entrega de valor contínuo aos stakeholders, utilizando práticas ágeis para garantir a flexibilidade e adaptação do time às necessidades do negócio. A documentação aqui apresentada detalha cada etapa do processo, desde o planejamento inicial até a entrega dos resultados, proporcionando uma visão clara e organizada do trabalho realizado.