

## 3.2 - Documento de Descrição da Aplicação de Monitoramento de POS com Alerta de Incidentes

### 1. Introdução

Este documento descreve a implementação de um sistema de monitoramento de POS (Point of Sales) que identifica e alerta sobre transações anômalas em tempo real. O sistema foi desenvolvido para detectar e notificar falhas, reversões e negações de transações, com o objetivo de melhorar a eficiência e a segurança das operações da Cloudwalk e seus produtos como o InfinitePay.

### 2. Descrição Geral do Sistema

O sistema de monitoramento de POS é composto pelos seguintes componentes principais:

1. **\*\*Endpoint de Recebimento de Dados\*\***: Recebe dados de transações em tempo real e retorna recomendações de alertas para anomalias.
2. **\*\*Modelo de Detecção de Anomalias\*\***: Utiliza aprendizado de máquina para identificar transações anômalas.
3. **\*\*Banco de Dados\*\***: Organiza e armazena os dados de transações.
4. **\*\*Gráficos de Monitoramento em Tempo Real\*\***: Visualiza os dados de transações e as anomalias detectadas.
5. **\*\*Sistema de Notificação de Anomalias\*\***: Envia alertas automáticos para a equipe de operações.

### 3. Dados Utilizados

Foram utilizados os seguintes arquivos de dados:

- `transactions\_1.csv`: Dados de transações iniciais.

As transações a serem monitoradas são classificadas em três tipos principais:

- **\*\*Denied\*\***: Transações negadas por risco dos emissores.
- **\*\*Reversed\*\***: Transações revertidas.
- **\*\*Failed\*\***: Transações falhadas.

## 4. Fluxo de Implementação

### 1. **\*\*Carregamento e Pré-processamento dos Dados\*\***:

- Leitura do arquivo CSV.
- Criação do Banco de Dados Mysql para armazenar todas as transações com o objetivo de ter dados futuros para análises e criação de novos modelos.

```
CREATE TABLE `transactions` (
  `id` int NOT NULL AUTO_INCREMENT,
  `time` timestamp NULL DEFAULT NULL,
  `status` varchar(30) CHARACTER SET latin1 COLLATE latin1_swedish_ci
  DEFAULT NULL,
  `count` int DEFAULT NULL,
  `Alert` int DEFAULT NULL,
  `final_decision` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8553 DEFAULT CHARSET=latin1;
```

### 2. **\*\*Treinamento do Modelo de Detecção de Anomalias\*\***:

- Utilização do modelo `Isolation Forest` para identificar anomalias.
- Divisão dos dados em conjuntos de treinamento e teste.
- Treinamento do modelo com dados padronizados.
- Arquivo: isolation\_forest.ipynb

```

✓ [21] %%time
0s
# Treinamento do modelo
model = IsolationForest(n_estimators=100, contamination=contamination, random_state=42)
model.fit(df)

CPU times: user 204 ms, sys: 227 µs, total: 205 ms
Wall time: 205 ms

IsolationForest
IsolationForest(contamination=0.05, random_state=42)

```

### 3. **\*\*Avaliação do Modelo\*\*:**

- Cálculo de métricas de desempenho como acurácia e relatório de classificação.

### 4. **\*\*Desenvolvimento do Endpoint de Recebimento de Dados\*\*:**

- Implementação de um endpoint que recebe dados de transações e retorna recomendações de alerta.
- Arquivo API: api.py - criado para testar o envio de request

```

# ---- recebendo da api ----
# Função para obter dados da API
def get_data_from_api():
    api_url = "http://localhost:5000/api/get_record" # URL da sua API Flask
    try:
        response = requests.get(api_url)
        response.raise_for_status() # Lança um erro para códigos de status HTTP não 200
        return response.json()
    except requests.exceptions.RequestException as e:
        st.error(f"Erro ao se conectar à API: {e}")
        return None

# Obter dados da API
data = get_data_from_api()

```

### 5. **\*\*Sistema de Notificação de Anomalias\*\*:**

- Implementação de um sistema de notificação que alerta automaticamente a equipe sobre anomalias detectadas por email.

```
# --- função para enviar email -----
def send_alert_email(to_email, transaction_details):
    # Configurações do servidor SMTP
    smtp_server = "smtp.gmail.com" # Altere conforme necessário
    smtp_port = 587 # Para TLS
    smtp_user = "danielle.ballester.896@ufrn.edu.br" # Altere para seu e-mail
    smtp_password = "Horus@#24" # Altere para sua senha

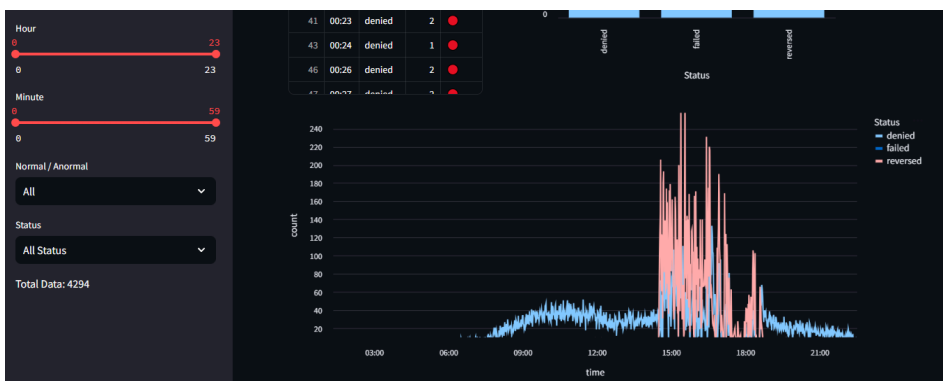
    # Criação da mensagem
    message = EmailMessage()
    message.set_content(f"Alerta de Transação:\n\n{transaction_details}")
    message["Subject"] = "Alerta de Anomalia em Transação"
    message["From"] = smtp_user
    message["To"] = to_email
```

## 5. Código Final

O código final é o “monitor.py” que se encontra com os demais no github.

## 6. Sobre o app

- Para criar uma aplicação rápida, foi utilizada a biblioteca Python “Streamlit”.
- A ideia principal é criar algo rápido para visualizar as transações do POS com as flags informados (failed, denied, reversed).
- O app tem uma área esquerda com controles para visualizar as transações do dia.
- No lado direito uma tabela, um gráfico de barras com os status monitorados. Tendo na parte de baixo um gráfico para monitorar a sequência temporal ao longo do dia.



## 7. Observações

O sistema de monitoramento de POS descrito neste documento foi feito localmente, portanto, para testar, será necessário rodar ambos os arquivos abaixo:

- `streamlit run monitor.py`
- `python api.py`

## 8. Conclusão

O sistema de monitoramento de POS descrito neste documento foi projetado para detectar e alertar sobre transações anômalas em tempo real. Ele utiliza técnicas avançadas de aprendizado de máquina para identificar padrões suspeitos nas transações e notificar automaticamente a equipe de operações. Este sistema não apenas melhora a segurança e a eficiência das operações, mas também oferece uma base sólida para futuras melhorias e expansões.

**Link do projeto no github:**

<https://github.com/daniballester-ai/monitoring>

++++

## 3.1 - Análise de Dados de Checkout

A análise dos dados de checkout fornecidos em "checkout\_1.csv" e "checkout\_1.csv" foi realizada utilizando Python e SQL, com os resultados e código detalhados no arquivo "cloudwalk\_code\_test.ipynb" presente no repositório do github.

<https://github.com/daniballester-ai/monitoring>

## Insights de checkout\_1.csv

### 1. Média das Vendas:

- **Hoje** (21.92) e **ontem** (21.79) tem médias de vendas bastante próximas.
- As vendas no mesmo dia da semana passada (23.83) são **um pouco mais altas** que hoje e ontem.
- A média das vendas na última semana (15.48) e no último mês (14.63) são **significativamente mais baixas do que hoje**.

Isso sugere *uma melhora no desempenho das vendas* em comparação com períodos anteriores.

### 2. Variação de Vendas Considerável:

- **Desvio padrão:** Os desvios padrão para "hoje", "ontem", "mesmo dia da semana passada", "semana passada" e "mês passado" são relativamente altos. Isso indica grandes flutuações nas vendas ao longo do dia, o que pode ser incomum.

### 3. Limite Inferior de Vendas Baixo:

- **Mínimo:** O mínimo para toda "today" e "yesterday" é 0.00, indicando que em alguns horários, as vendas podem chegar a zero. O que já não ocorreu na semana e mês passados.

Essa informação é importante para entender a dinâmica das vendas e identificar períodos de baixa atividade.

## 4. Distribuição Desigual das Vendas:

- **Quartis:** A diferença entre os quartis indica uma distribuição desigual das vendas. Por exemplo, o quartil 75% é muito mais alto que o quartil 25%, indicando que uma grande parte das vendas se concentra em um período menor do dia.

Essa informação pode ser útil para otimizar o planejamento de recursos, como funcionários e estoque, durante as horas de pico.

## 5. Possíveis Oportunidades de Crescimento:

- **Comparação com a Média:** As vendas "hoje" estão significativamente acima da média da semana passada e do mês passado.

Essa diferença indica um potencial para aumentar as vendas e otimizar o desempenho, explorando fatores que contribuíram para o bom desempenho "hoje".

### Em resumo, os resultados sugerem que:

- As vendas "hoje" estão acima da média em relação a outros períodos, indicando um desempenho positivo.
- Existem períodos de baixa atividade e concentração de vendas em momentos específicos do dia.
- Há um potencial para aumentar as vendas, explorando as razões para o bom desempenho "hoje" e otimizando a estratégia de vendas.
- Embora as vendas estejam acima da média, alguns períodos estão sem vendas, diferente da semana e mês passados.

## Insights de checkout\_2

### 1. Média de Vendas:

- **Hoje (17.79):** A média de vendas de hoje é menor do que a de ontem (21.92) e do mesmo dia da semana passada (20.33). Isso pode indicar uma queda recente nas vendas.

- **Média Semanal (11.94) e Mensal (14.53):** A média de hoje ainda é maior do que a média semanal, mas menor do que a média mensal.

## 2. Variabilidade:

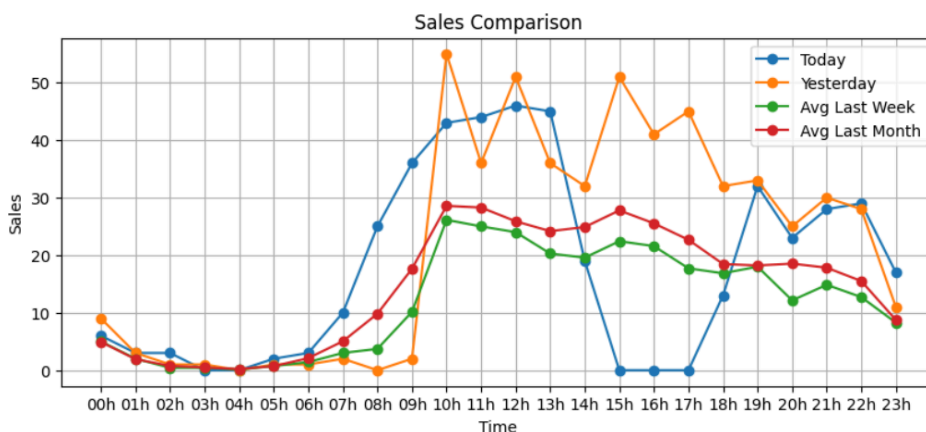
- **Desvio Padrão:** O desvio padrão de hoje (16.70) e de ontem (19.41) também é alto, indicando grandes flutuações nas vendas.

## 3. Mínimo e Máximo:

- **Mínimo de 0.00:** Períodos sem vendas também são observados, o que pode ser anômalo.
- **Máximo de 46.00 (Hoje) e 55.00 (Ontem):** As vendas máximas são menores do que ontem, mas ainda consideráveis.

## 4. Distribuição:

- A mediana (15.00) de hoje é menor do que a de ontem (26.50), indicando que a maioria das vendas de hoje está concentrada em um valor menor comparado a ontem.



## Insights e Possíveis Anomalias

### 1. Aumento e Queda nas Vendas:

- checkout\_1.csv é o arquivo de ontem para checkout\_2.csv.



- Em checkout\_1.csv, há um aumento recente nas vendas que pode ser devido a uma promoção ou evento específico.
- Em checkout\_2.csv, há uma queda recente nas vendas de hoje comparado a ontem e ao mesmo dia da semana passada.
- O comportamento de hoje tem mediana menor do que a média, diferente de ontem, e médias da semana passada e mês passado, **no entanto**, é um comportamento semelhante ao mesmo dia da semana passada, o que pode demonstrar um padrão para esse dia da semana.

## 2. **Alta Variabilidade:**

- Ambos os arquivos mostram alta variabilidade nas vendas, o que pode indicar flutuações significativas demonstrando horários de picos e quedas de vendas.

## 3. **Períodos sem Vendas:**

- A presença de períodos sem vendas (mínimo de 0.00) pode indicar problemas com o sistema de ponto de venda (POS) ou horários de operação que precisam ser revisados.

## 4. **Picos de Vendas:**

- Os valores máximos altos indicam que há picos de vendas significativos que podem estar associados a horários específicos do dia ou a eventos especiais.

# Próximos Passos

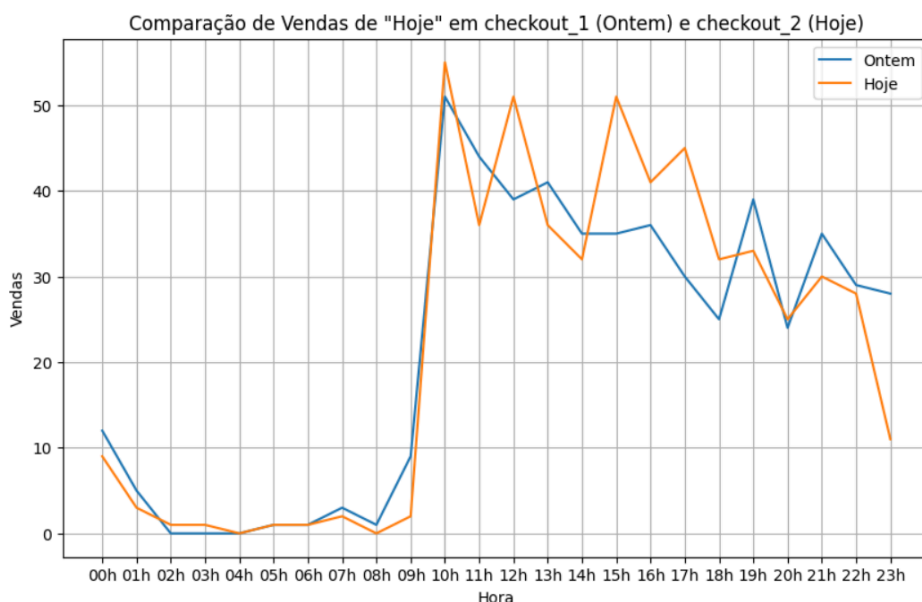
Para entender melhor essas anomalias, recomenda-se:

1. **Analisar Horários Específicos:** Verificar em que horas ocorrem os picos e quedas nas vendas para identificar padrões temporais.
2. **Comparar com Eventos:** Verificar se há eventos, promoções ou mudanças operacionais que coincidem com os aumentos ou quedas nas vendas.

3. **Monitorar o Sistema POS:** Garantir que o sistema POS está funcionando corretamente e que não há problemas técnicos causando os períodos sem vendas.

## Consulta SQL e Gráfico

Criação de uma consulta em SQL e crie um gráfico dela e tente explicar o comportamento da anomalia que você encontrou.



## Explicação da anomalia:

### Minhas conclusões finais:

- O gráfico mostra que as vendas de "hoje" (checkout\_2) são geralmente mais baixas do que as de "ontem" (checkout\_1) na maioria dos horários.
- Isso sugere uma queda no desempenho das vendas no dia atual.
- Uma possível explicação para essa anomalia pode ser um evento específico ou promoção que ocorreu no dia anterior e não está presente hoje.

- Outras causas possíveis incluem mudanças no comportamento do consumidor, fatores externos como clima ou feriados, ou problemas operacionais internos.

<https://github.com/daniballester-ai/monitoring>

arquivo: cloudwalk\_code\_test.ipynb

Case Técnico desenvolvido por: Danielle Ballester