

Case *ifood* : Biometrias

Avaliação da performance dos serviços de autenticação biométrica para entregadores





Contexto

Objetivo da análise:

Avaliar a performance dos serviços de autenticação biométrica utilizados para validar a identidade dos entregadores, identificando padrões de falhas, potenciais comportamentos suspeitos e oportunidades de melhoria no processo de prevenção à fraude.

Cenário:

- O processo de autenticação facial dos entregadores utiliza até 3 serviços (A, B e C).
- O sistema tenta o serviço A inicialmente; caso falhe, recorre a B e C como fallback.
- O status final da sessão é considerado “MATCH” se a similaridade calculada for ≥ 0.80 .



Estrutura de análise:

Com base nos registros históricos de biometrias, foram identificados:

1. Qual serviço apresenta mais falhas de provedor;
2. Quais perfis de entregadores têm maior dificuldade de autenticação;
3. O impacto de elevar o limiar de similaridade (0.8 \rightarrow 0.9);
4. Possível relação entre cancelamentos de pedidos e falhas biométricas;
5. Padrões de comportamento que indiquem “empréstimo de conta” ao longo do período.

Foco da análise:

Utilizar dados operacionais para gerar insights acionáveis para o time de Prevenção à Fraude, apoiando decisões de segurança e eficiência operacional.





Preparação dos Dados

Ambiente de análise:

- BigQuery — ingestão das tabelas CSV, execução de SQL e criação de views analíticas.
- Looker Studio — conexão direta ao BigQuery para gerar gráficos.

Datasets (.csv):

- `biometry` — sessão final de biometria (1 linha por sessão)
- `biometry_execution` — execuções de até 3 serviços por sessão (A, B, C)
- `drivers` — cadastro do entregador
- `orders` — pedidos associados ao entregador



Etapas da metodologia:

1. Qualidade de dados:

- Checagem de nulos, duplicidades e integridade referencial.
- Padronização de tipos e substituição de #N/D por NULL em orders.Driver_ID.
- Para manter o foco principal da apresentação, as queries de checagem estão disponíveis [nesta pasta](#).

2. Modelagem analítica (views criadas):

- drivers_typed → padronização de tipo em Driver_ID.
- orders_typed → padronização de tipos e remoção de registros com Driver_ID nulo.
- biometry_typed / biometry_execution_typed → padronização de IDs.
- vw_biometry_base → união de biometry + biometry_execution.
- vw_biometry_enriched → adição de dados do entregador.
- vw_orders_daily → agregação diária de pedidos e cancelamentos.

3. Análises SQL (A-E):

- Comparação de taxas de falha e match;
- Simulação de novo limiar;
- Correlação entre cancelamentos e falhas;
- Identificação de dias suspeitos (empréstimo de conta).



Queries da modelagem analítica (criação de views)

Views `drivers_typed`, `orders_typed`, `biometry_typed` e `biometry_execution_typed`

```
CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.drivers_typed` AS
SELECT CAST(Driver_ID AS STRING) AS Driver_ID,
       Category,
       Modal,
       Register_Dt,
       Device_ID
FROM `ifood-prev-fraude-case.fraude_biometria.drivers`;

CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.biometry_typed` AS
SELECT CAST(Driver_ID AS STRING) AS Driver_ID,
       CAST(Session_ID AS STRING) AS Session_ID,
       Status,
       Action,
       Event_Dt
FROM `ifood-prev-fraude-case.fraude_biometria.biometry`;

CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.biometry_execution_typed` AS
SELECT CAST(Session_ID AS STRING) AS Session_ID,
       Provider,
       Status,
       Similarity,
       Event_Dt
FROM `ifood-prev-fraude-case.fraude_biometria.biometry_execution`;

CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.orders_typed` AS
SELECT CAST(Order_ID AS STRING) AS Order_ID,
       NULLIF(Driver_ID, '#N/D') AS Driver_ID,
       Order_Dt,
       Order_Status,
       Order_Value
FROM `ifood-prev-fraude-case.fraude_biometria.orders`;
```



Queries da modelagem analítica (criação de views)

View vw_biometry_base

```
CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.vw_biometry_base` AS
SELECT b.Session_ID,
       b.Driver_ID,
       b.Status AS Final_Status,
       b.Action,
       b.Event_Dt,
       -- Status por provedor
       MAX(IF(e.Provider = 'A', e.Status, NULL)) AS Status_A,
       MAX(IF(e.Provider = 'B', e.Status, NULL)) AS Status_B,
       MAX(IF(e.Provider = 'C', e.Status, NULL)) AS Status_C,
       -- Similaridade por provedor
       MAX(IF(e.Provider = 'A', e.Similarity, NULL)) AS Sim_A,
       MAX(IF(e.Provider = 'B', e.Similarity, NULL)) AS Sim_B,
       MAX(IF(e.Provider = 'C', e.Similarity, NULL)) AS Sim_C,
       -- Maior similaridade na sessão (para simular threshold)
       COALESCE(
           SAFE_CAST(MAX(e.Similarity) AS FLOAT64),
           SAFE_CAST(REPLACE(MAX(e.Similarity), ',', '.') AS FLOAT64)
       ) AS Max_Service_Similarity
FROM `ifood-prev-fraude-case.fraude_biometria.biometry_typed` b
LEFT JOIN `ifood-prev-fraude-case.fraude_biometria.biometry_execution_typed` e ON b.Session_ID = e.Session_ID
GROUP BY b.Session_ID,
         b.Driver_ID,
         b.Status,
         b.Action,
         b.Event_Dt;
```



Queries da modelagem analítica (criação de views)

View `vw_biometry_enriched`

```
CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched` AS
SELECT v.*,
       d.Category,
       d.Modal,
       d.Register_Dt,
       d.Device_ID
FROM `ifood-prev-fraude-case.fraude_biometria.vw_biometry_base` v
LEFT JOIN `ifood-prev-fraude-case.fraude_biometria.drivers_typed` d ON v.Driver_ID = d.Driver_ID;
```



Queries da modelagem analítica (criação de views)

View `vw_orders_daily`

```
CREATE OR REPLACE VIEW `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily` AS
SELECT Driver_ID,
       DATE(Order_Dt) AS Dt,
       COUNT(*) AS Total_Orders,
       COUNTIF(Order_Status = 'CANCELLED') AS Cancels,
       SAFE_DIVIDE(COUNTIF(Order_Status = 'CANCELLED'), COUNT(*)) AS Cancel_Rate
FROM `ifood-prev-fraude-case.fraude_biometria.orders_typed`
WHERE Driver_ID IS NOT NULL
GROUP BY Driver_ID,
         Dt;
```

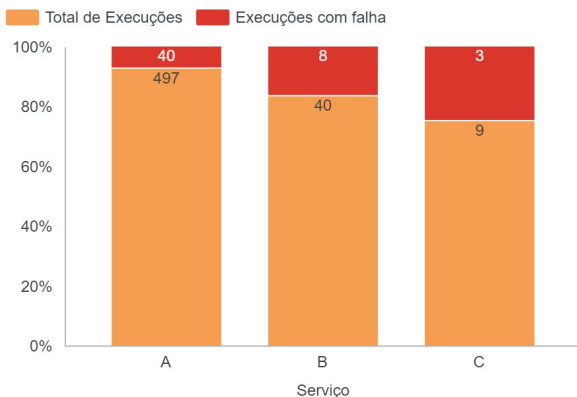




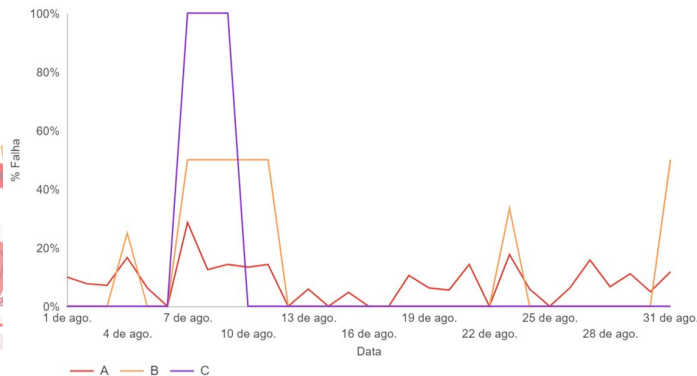
Resultados da Análise

A. Qual dos serviços mais falha? (status = PROVIDER_FAILED)

Total de Execuções x Taxa de Falhas por Serviço



Taxa de Falha dos Serviços de Biometria ao Longo do Período



O **Serviço C** apresentou a **maior taxa de falhas (33%)**, seguido do **Serviço B (20%)**, indicando **baixa confiabilidade dos fallbacks** em relação ao serviço principal. A **taxa total de falha do sistema foi de apenas 0,6%**, sustentada pela maior disponibilidade do Serviço A e pela estratégia robusta de resiliência.

Ao longo do período, o **Serviço C** registrou um **pico concentrado de falhas entre 07 e 09/08**, enquanto o **Serviço B** apresentou **instabilidades recorrentes**, com picos distribuídos em diversas datas, indicando problemas de estabilidade crônicos.

Reduzir as falhas em B e C é essencial para garantir **resiliência operacional** em caso de indisponibilidade do A.

✓ Recomendação

Engenharia: revisar logs, autenticação e estabilidade da API dos provedores B e C, priorizando a correção das falhas crônicas do B e a análise pontual do incidente de C (07–09/08).

A. Qual dos serviços mais falha? (status = PROVIDER_FAILED)

Queries executadas:

```
-- A: Falhas de provedor por serviço
SELECT Provider,
       COUNTIF(Status = 'PROVIDER_FAILED') AS provider_failed,
       COUNT(*) AS total_execucoes,
       SAFE_DIVIDE(COUNTIF(Status = 'PROVIDER_FAILED'), COUNT(*)) AS pct_failed
FROM `ifood-prev-fraude-case.fraude_biometria.biometry_execution_typed`
GROUP BY Provider
ORDER BY pct_failed DESC,
       provider_failed DESC;
```

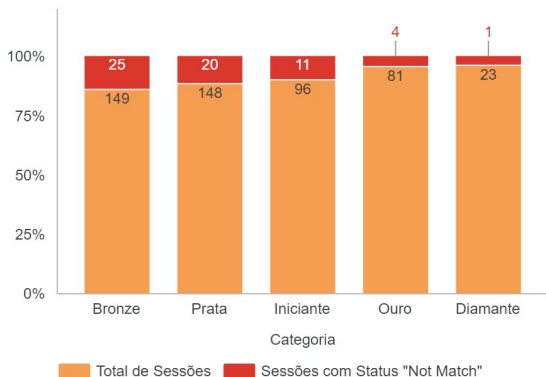
```
-- Taxa de falha total no processo de biometria
SELECT COUNT(*) AS total_sesoes,
       COUNTIF(
         COALESCE(Status_A, '') = 'PROVIDER_FAILED'
         AND COALESCE(Status_B, '') = 'PROVIDER_FAILED'
         AND COALESCE(Status_C, '') = 'PROVIDER_FAILED'
       ) AS sesoes_falhas,
       SAFE_DIVIDE(
         COUNTIF(
           COALESCE(Status_A, '') = 'PROVIDER_FAILED'
           AND COALESCE(Status_B, '') = 'PROVIDER_FAILED'
           AND COALESCE(Status_C, '') = 'PROVIDER_FAILED'
         ),
         COUNT(*)
       ) AS taxa_falha_total
FROM `ifood-prev-fraude-case.fraude_biometria.vw_biometry_base`;
```

```
-- Taxa de falha diária por provedor
SELECT DATE(Event_Dt) AS dt,
       Provider,
       COUNT(*) AS total_execucoes,
       COUNTIF(Status = 'PROVIDER_FAILED') AS falhas,
       SAFE_DIVIDE(COUNTIF(Status = 'PROVIDER_FAILED'), COUNT(*)) AS taxa_falha
FROM `ifood-prev-fraude-case.fraude_biometria.biometry_execution_typed`
WHERE Event_Dt IS NOT NULL
GROUP BY dt,
       Provider
ORDER BY dt,
       Provider;
```



B. Qual categoria de entregador possui o maior índice falha na identificação biométrica? (status = NOT_MATCH)

Proporção de Falhas na Biometria por Categoria de Entregador



Bronze apresenta a **maior taxa de falha na autenticação facial (17%)**, seguida pela **Prata (13%)**, além da **maior taxa de cancelamento (24%)** e **alto tempo de plataforma**. Falhas parecem não serem explicadas por inexperience do entregador, mas por **fatores operacionais** (más condições de captura, desatenção) ou **comportamentais** (uso indevido da conta no mesmo aparelho, sem indícios de múltiplos devices).

Os **Iniciantes** também possuem **taxas elevadas de falha e cancelamento**, com um maior tempo de plataforma, indicando possível **uso ocasional de contas antigas** por entregadores pouco ativos, com **menor familiaridade operacional** ou **possível reativação por terceiros**.

Categoria	% Falha	% Cancelamento (média)	Dias de Registro na plataforma (mediana)
Bronze	17%	24%	733
Prata	13%	13%	584
Iniciante	11%	18%	772
Ouro	5%	11%	679
Diamante	4%	5%	810

✓ Recomendação

Engenharia: revisar parâmetros de matching e tolerância do modelo;
Operações: reforçar instruções de selfie e boas práticas no app;
Prevenção à Fraude: analisar geolocalização e horários de sessão, buscando padrões atípicos e indícios de compartilhamento de conta.

B. Qual categoria de entregador possui o maior índice falha na identificação biométrica? (status = NOT_MATCH)

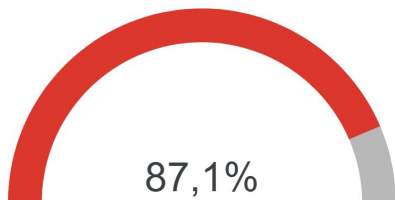
Query executada:

```
-- B: NOT_MATCH + taxa de cancelamento + média de devices +
-- MEDIANA de dias desde o registro
-- NOT_MATCH: sessões que falharam na biometria
WITH base_biometry AS (
    SELECT Category,
           Driver_ID,
           COUNT(*) AS total_sesoes,
           COUNTIF(Final_Status = 'NOT_MATCH') AS not_match
    FROM
        `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
    GROUP BY Category,
           Driver_ID
),
-- cancelamentos por driver (total e taxa)
base_orders AS (
    SELECT Driver_ID, \
           SUM(Total_Orders) AS total_orders,
           SUM(Cancels) AS cancels,
           SAFE_DIVIDE(SUM(Cancels), NULLIF(SUM(Total_Orders),
0)) AS cancel_rate
    FROM
        `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily`
    GROUP BY Driver_ID
),
-- quantidade média de devices por driver
base_devices AS (
    SELECT Driver_ID,
           COUNT(DISTINCT Device_ID) AS qtd_devices
    FROM
        `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
    WHERE Device_ID IS NOT NULL
    GROUP BY Driver_ID
),
-- mediana por driver dos dias desde o registro até as sessões
tempo_driver AS (
    SELECT Category,
           Driver_ID,
           APPROX_QUANTILES(
               DATE_DIFF(DATE(Event_Dt), DATE(Register_Dt), DAY),
               100
           ) [OFFSET(50)] AS median_days_since_register_driver
    FROM
        `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
    WHERE Register_Dt IS NOT NULL
        AND Event_Dt IS NOT NULL
    GROUP BY Category,
           Driver_ID
),
-- mediana tempo de registro por categoria (entre os drivers da
-- categoria)
tempo_categoria AS (
    SELECT Category,
           APPROX_QUANTILES(median_days_since_register_driver, 100)
[OFFSET(50)] AS median_days_since_register
    FROM tempo_driver
    GROUP BY Category
),
SELECT b.Category,
       -- biometria
       SUM(b.total_sesoes) AS total_sesoes,
       SUM(b.not_match) AS not_match,
       SAFE_DIVIDE(SUM(b.not_match), SUM(b.total_sesoes)) AS
not_match_rate,
       -- cancelamentos
       ROUND(AVG(o.cancel_rate), 4) AS avg_cancel_rate,
       -- devices
       ROUND(AVG(d.qtd_devices), 2) AS
avg_devices_per_driver,
       -- mediana de dias desde o registro (por categoria)
       tempo.median_days_since_register
FROM base_biometry b
    LEFT JOIN base_orders o USING (Driver_ID)
    LEFT JOIN base_devices d USING (Driver_ID)
    LEFT JOIN tempo_categoria tempo USING (Category)
GROUP BY b.Category,
         tempo.median_days_since_register
ORDER BY not_match_rate DESC,
         avg_cancel_rate DESC;
```

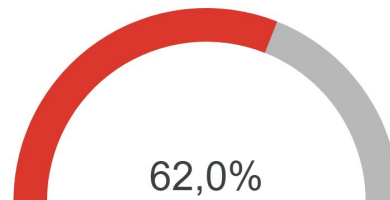


C. Calcule quanto o índice geral de MATCH seria se aumentássemos a similaridade mínima do MATCH para 0.90.

Índice Geral de Match - Similaridade ≥ 0.80



Índice Geral de Match - Similaridade ≥ 0.90



▼ **28,9 p.p.**
Queda Relativa de **~33%**

A elevação do limiar de similaridade de 0.80 para 0.90 **reduziria o índice geral de match de 87,1% para 62%,** representando uma queda de **28,9 p.p. (~33% relativa).**

Embora o sistema se torne **mais rigoroso e seguro**, também rejeitaria um terço das autenticações atualmente aceitas, **afetando a fluidez operacional e a disponibilidade do sistema.**

✓ Recomendação

Prevenção à Fraude: Avaliar o novo limiar de forma segmentada por perfil de entregador, considerando histórico de falhas e categorias de risco. Testar modelos adaptativos ou limiares dinâmicos, equilibrando segurança e eficiência operacional.

C. Calcule quanto o índice geral de MATCH seria se aumentássemos a similaridade mínima do MATCH para 0.90.

Query executada:

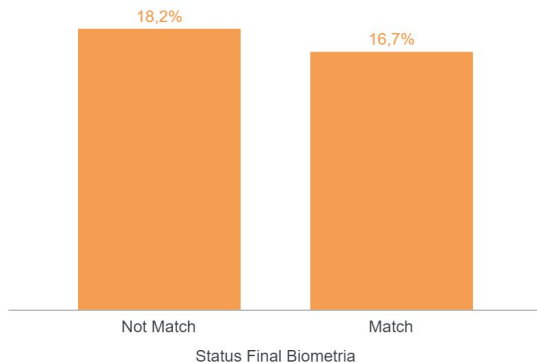
```
-- C: Comparação de match rate atual (0.80) vs match rate por threshold de similaridade 0.90
WITH base AS (
  SELECT Final_Status,
         Max_Service_Similarity
  FROM `ifood-prev-fraude-case.fraude_biometria.vw_biometry_base`
)
SELECT COUNT(*) AS total_sesoes,
       COUNTIF(Final_Status = 'MATCH') AS match_atual,
       SAFE_DIVIDE(COUNTIF(Final_Status = 'MATCH'), COUNT(*)) AS match_rate_atual,
       COUNTIF(Max_Service_Similarity >= 0.90) AS match_sim_090,
       SAFE_DIVIDE(
         COUNTIF(Max_Service_Similarity >= 0.90),
         COUNT(*)
       ) AS match_rate_sim_090,
       SAFE_DIVIDE(
         COUNTIF(Max_Service_Similarity >= 0.90) - COUNTIF(Final_Status = 'MATCH'),
         NULLIF(COUNTIF(Final_Status = 'MATCH'), 0)
       ) AS delta_relativo_vs_atual
FROM base;
```



D. Você diria que há alguma relação entre volume de pedidos cancelados (status = CANCELLED) de um entregador e o status final do processo de identificação biométrica dele? Justifique sua resposta.

% Cancelamento x Status Biometria

% Cancelamento



**Correlação (r) entre
Cancelamento x Status**

0,04

Os entregadores com falha na biometria (**NOT_MATCH**) apresentaram taxa média de cancelamento **ligeiramente maior (18,1%)** que os com **MATCH (16,5%)**, mas a **diferença foi pequena e estatisticamente não significativa.**

A correlação entre as variáveis de **-0** aponta para a **ausência de relação linear**, indicando que **cancelamentos e falhas biométricas provavelmente ocorrem de forma independente**, refletindo fatores operacionais mais do que comportamentais.

✓ Recomendação

Operações: foco em outras causas operacionais de cancelamento (rotas, atrasos, indisponibilidade).

Prevenção à Fraude: Expandir o monitoramento para um período maior de observação, garantindo amostras equivalentes de entregadores, para validar estatisticamente a ausência de correlação entre falhas biométricas e cancelamentos.



D. Você diria que há alguma relação entre volume de pedidos cancelados (status = CANCELLED) de um entregador e o status final do processo de identificação biométrica dele?

Justifique sua resposta.

Queries executadas:

```
-- D1: visão por driver: sessões de biometria + pedidos + taxa de cancelamento (explorada, mas não utilizada nos slides)
WITH sess_por_driver AS (
  SELECT Driver_ID,
    COUNT(*) AS sessoes,
    COUNTIF(Final_Status = 'MATCH') AS qtd_match,
    COUNTIF(Final_Status = 'NOT_MATCH') AS qtd_not_match
  FROM `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
  WHERE Driver_ID IS NOT NULL
  GROUP BY Driver_ID
),
canc_por_driver AS (
  SELECT Driver_ID,
    SUM(Total_Orders) AS orders_total,
    SUM(Cancels) AS cancels_total,
    SAFE_DIVIDE(SUM(Cancels), NULLIF(SUM(Total_Orders), 0)) AS cancel_rate
  FROM `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily`
  WHERE Driver_ID IS NOT NULL
  GROUP BY Driver_ID
)
SELECT s.Driver_ID,
  s.ssessoes,
  SAFE_DIVIDE(s.qtd_not_match, s.ssessoes) AS not_match_rate,
  c.orders_total,
  c.cancels_total,
  c.cancel_rate
FROM sess_por_driver s
  LEFT JOIN canc_por_driver c USING (Driver_ID)
ORDER BY not_match_rate DESC,
  cancel_rate DESC;
```



D. Você diria que há alguma relação entre volume de pedidos cancelados (status = CANCELLED) de um entregador e o status final do processo de identificação biométrica dele?

Justifique sua resposta.

Queries executadas:

```
-- D2: Média da taxa de cancelamento por grupo de biometria
WITH sess_por_driver AS (
  SELECT
    Driver_ID,
    COUNT(*) AS sessoes,
    COUNTIF(Final_Status = 'MATCH') AS qtd_match,
    COUNTIF(Final_Status = 'NOT_MATCH') AS qtd_not_match
  FROM
    `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
  WHERE Driver_ID IS NOT NULL
  GROUP BY Driver_ID
),
canc_por_driver AS (
  SELECT
    Driver_ID,
    SUM(Total_Orders) AS orders_total,
    SUM(Cancels) AS cancels_total,
    SAFE_DIVIDE(SUM(Cancels), NULLIF(SUM(Total_Orders), 0))
  AS cancel_rate
  FROM
    `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily`
  WHERE Driver_ID IS NOT NULL
  GROUP BY Driver_ID
),
joined AS (
  SELECT
    s.Driver_ID,
    SAFE_DIVIDE(s.qtd_not_match, s.sesoes) AS
    not_match_rate,
    c.cancel_rate
  FROM sess_por_driver s
  LEFT JOIN canc_por_driver c USING (Driver_ID)
  WHERE c.cancel_rate IS NOT NULL
)
SELECT
  CASE
    WHEN not_match_rate = 1 THEN 'Somente NOT_MATCH'
    WHEN not_match_rate = 0 THEN 'Somente MATCH'
    ELSE 'Misto'
  END AS grupo_biometria,
  COUNT(*) AS qtd_drivers,
  ROUND(AVG(cancel_rate), 4) AS media_cancel_rate
FROM joined
GROUP BY grupo_biometria;

-- Correlação entre status biometria e taxa de cancelamento
WITH por_driver AS (
  SELECT Driver_ID,
    SAFE_DIVIDE(COUNTIF(Final_Status = 'NOT_MATCH'), COUNT(*)) AS not_match_rate
  FROM `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
  WHERE Final_Status IN ('MATCH', 'NOT_MATCH')
  GROUP BY Driver_ID
),
canc_por_driver AS (
  SELECT Driver_ID,
    SAFE_DIVIDE(SUM(Cancels), NULLIF(SUM(Total_Orders), 0)) AS cancel_rate
  FROM `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily`
  WHERE Driver_ID IS NOT NULL
  GROUP BY Driver_ID
)
SELECT CORR(p.not_match_rate, c.cancel_rate) AS correlacao
FROM por_driver p
LEFT JOIN canc_por_driver c USING (Driver_ID);
```



D. Você diria que há alguma relação entre volume de pedidos cancelados (status = CANCELLED) de um entregador e o status final do processo de identificação biométrica dele?

Justifique sua resposta.

Queries executadas:

```
-- Teste t
WITH por_driver AS (
  SELECT
    Driver_ID,
    SAFE_DIVIDE(SUM(Cancels), NULLIF(SUM(Total_Orders), 0))
  AS cancel_rate,
  CASE
    WHEN Final_Status = 'NOT_MATCH' THEN 'NOT_MATCH'
    WHEN Final_Status = 'MATCH' THEN 'MATCH'
    ELSE NULL
  END AS grupo
  FROM
    `ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriche
d` b
  JOIN
    `ifood-prev-fraude-case.fraude_biometria.vw_orders_daily` o
    USING (Driver_ID)
  WHERE Final_Status IN ('MATCH', 'NOT_MATCH')
  GROUP BY Driver_ID, grupo
),

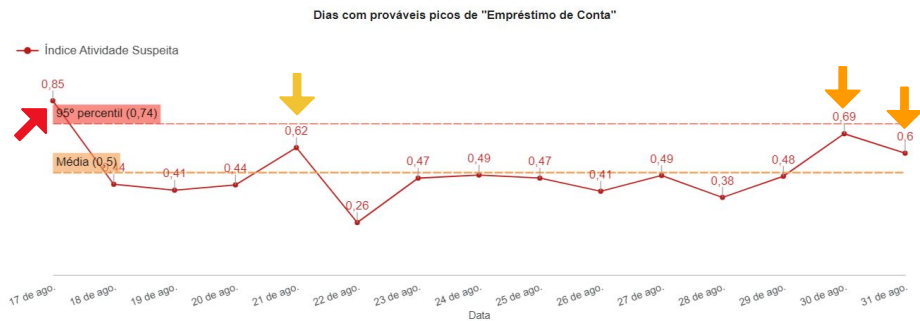
-- Estatísticas por grupo (MATCH vs NOT_MATCH)
stats AS (
  SELECT
    grupo,
    COUNT(*) AS n,
    AVG(cancel_rate) AS mean_cancel,
    STDDEV_POP(cancel_rate) AS std_cancel
  FROM por_driver
  GROUP BY grupo
),

-- Combina estatísticas em uma única linha
combined AS (
  SELECT
    MAX(CASE WHEN grupo = 'MATCH' THEN n END) AS n1,
    MAX(CASE WHEN grupo = 'MATCH' THEN mean_cancel END) AS mean1,
    MAX(CASE WHEN grupo = 'MATCH' THEN std_cancel END) AS std1,
    MAX(CASE WHEN grupo = 'NOT_MATCH' THEN n END) AS n2,
    MAX(CASE WHEN grupo = 'NOT_MATCH' THEN mean_cancel END) AS mean2,
    MAX(CASE WHEN grupo = 'NOT_MATCH' THEN std_cancel END) AS std2
  FROM stats
),

-- Cálculo do t-statistic
SELECT
  *,
  (mean1 - mean2) AS diff_means,
  SQRT(POW(std1, 2)/n1 + POW(std2, 2)/n2) AS se_diff,
  SAFE_DIVIDE((mean1 - mean2), SQRT(POW(std1, 2)/n1 + POW(std2, 2)/n2)) AS t_stat
FROM combined;
```



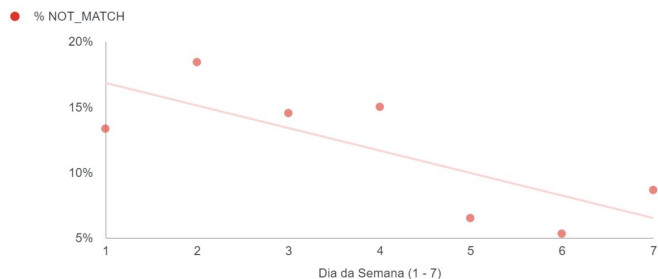
E. Em quais dias provavelmente tivemos um aumento do comportamento indevido “empréstimo de conta”?



- **17/08 (terça):** Dia mais provável - dia útil + alto índice.
- **21/08 (sábado):** Elevação operacional sem padrão de fraude (fim de semana).
- **30 e 31/08 (segunda e terça):** Dias úteis, coerente com padrão de empréstimo em dias de maior demanda.

O **Índice de Suspeita de Empréstimo de Conta**, que pondera reprovações biométricas, intensidade de autenticações e produtividade fora do padrão, indicou um **pico expressivo em 17/08 (0,85)** e **elevações moderadas em 21, 30 e 31/08.**

Correlação Dia da Semana x % NOT_MATCH

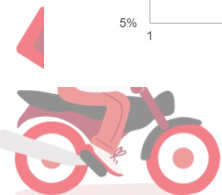


A **forte correlação negativa (-0,76)** entre a **taxa de falhas biométricas e o avanço da semana**, sugere que o empréstimo de conta ocorre com **maior frequência em dias úteis**, especialmente no início da semana.

✓ Recomendação

Prevenção à Fraude: concentrar auditorias e monitoramento de biometria em dias úteis (segunda à quinta), que historicamente apresentam as maiores taxas de reprovação.

Engenharia: avaliar se os provedores estão calibrando consistentemente o modelo de matching durante a semana (descartar interferência técnica).



E. Em quais dias provavelmente tivemos um aumento do comportamento indevido “empréstimo de conta”?

Premissas adotadas para o Índice de Suspeita de Empréstimo de Conta

- O empréstimo de conta pode se manifestar por **falhas na autenticação facial (NOT_MATCH)**, **aumento anormal de sessões biométricas** e **picos de produtividade fora do padrão (entregadores com muitos pedidos no dia)**.
- Como **nem todo aumento de atividade indica fraude**, o índice pondera comportamento biométrico e operacional, dando **maior peso à taxa de falhas** na autenticação.

Métricas componentes

1. Z-Score de NOT_MATCH (peso 0,5):

Mede o quanto a taxa diária de falhas biométricas está acima da média.

→ Maior peso por refletir forte indício de autenticação indevida.

2. Z-Score de Taxa de Biometria (peso 0,3):

Representa o desvio na proporção de entregadores auditados via biometria.

→ Aumentos bruscos indicam reforço de checagem por identificação de comportamento anômalo pelo sistema.

3. Z-Score de Entregadores acima do P95 de Pedidos (peso 0,2):

Mede a proporção de entregadores com produtividade fora do padrão (outliers).

→ Pode indicar uso simultâneo de conta por múltiplas pessoas à mesma conta.

Lógica de cálculo

Cada métrica é convertida em um z-score (número de desvios-padrão acima da média).

Esses valores são então ponderados pelos pesos definidos e somados, gerando um índice bruto, posteriormente normalizado entre 0 e 1 para facilitar a leitura:

$$\text{Índice} = 0.5 \times Z(\text{NotMatch}) + 0.3 \times Z(\text{Biometria}) + 0.2 \times Z(\text{P95Pedidos}) \rightarrow \text{normalizado (0-1)}$$



E. Em quais dias provavelmente tivemos um aumento do comportamento indevido “empréstimo de conta”?

Queries executadas:

```
-- E1: Índice de suspeita diário de fraude por
compartilhamento de conta
-- Combina métricas de biometria (NOT_MATCH), cobertura
biométrica e produtividade atípica
WITH pedidos_por_driver_dia AS (
  SELECT DATE(Order_Dt) AS dt,
         Driver_ID,
         COUNT(*) AS pedidos_no_dia
  FROM `ifood-prev-fraude-case.fraude_biometria.orders`
  WHERE Order_Dt IS NOT NULL
        AND Driver_ID IS NOT NULL
  GROUP BY dt,
         Driver_ID
),
-- Calcula o percentil 95 (P95) de pedidos por dia
p95_por_dia AS (
  SELECT dt,
         APPROX_QUANTILES(pedidos_no_dia, 100) [OFFSET(95)] AS
p95_pedidos_no_dia
  FROM pedidos_por_driver_dia
  GROUP BY dt
),
-- Identificar % de entregadores com pedidos acima do P95
(outliers de produtividade)
drivers_acima_p95 AS (
  SELECT p.dt,
         COUNT(DISTINCT p.Driver_ID) AS drivers_ativos,
         SAFE_DIVIDE(
           COUNTIF(p.pedidos_no_dia > q.p95_pedidos_no_dia),
           COUNT(DISTINCT p.Driver_ID)
         ) AS pct_acima_p95
  FROM pedidos_por_driver_dia p
  JOIN p95_por_dia q USING (dt)
  GROUP BY p.dt
),
-- Métricas biométricas por driver/dia
sessoes_por_driver_dia AS (
  SELECT DATE(Event_Dt) AS dt,
         Driver_ID,
         COUNT(*) AS sessoes_bio_no_dia,
         COUNTIF(Final_Status = 'NOT_MATCH') AS
not_match_bio_no_dia
  FROM
`ifood-prev-fraude-case.fraude_biometria.vw_biometry_enriched`
  WHERE Event_Dt IS NOT NULL
        AND Driver_ID IS NOT NULL
  GROUP BY dt,
         Driver_ID
),
-- Agrega métricas diárias combinando operação e biometria
dia_agregado AS (
  SELECT p.dt,
         COUNT(DISTINCT p.Driver_ID) AS drivers_ativos_no_dia,
         COUNT(DISTINCT s.Driver_ID) AS drivers_biometria_no_dia,
         -- % de entregadores auditados via biometria
         SAFE_DIVIDE(
           COUNT(DISTINCT s.Driver_ID),
           COUNT(DISTINCT p.Driver_ID)
         ) AS taxa_cobertura_bio,
         -- taxa de NOT_MATCH diária
         SAFE_DIVIDE(
           SUM(s.not_match_bio_no_dia),
           NULLIF(SUM(s.sessoes_bio_no_dia), 0)
         ) AS taxa_not_match,
         -- % de entregadores com produtividade acima do P95 do dia
         da.pct_acima_p95
  FROM pedidos_por_driver_dia p
  LEFT JOIN sessoes_por_driver_dia s ON p.dt = s.dt
  AND p.Driver_ID = s.Driver_ID
  LEFT JOIN drivers_acima_p95 da ON p.dt = da.dt
  GROUP BY p.dt,
         da.pct_acima_p95
),
-- Continua no próximo slide
```



E. Em quais dias provavelmente tivemos um aumento do comportamento indevido “empréstimo de conta”?

Queries executadas:

```
-- Continuação da query anterior
-- Estatísticas globais para z-scores
estat AS (
  SELECT AVG(taxa_cobertura_bio) AS media_bio,
         STDDEV_SAMP(taxa_cobertura_bio) AS sd_bio,
         AVG(taxa_not_match) AS media_not,
         STDDEV_SAMP(taxa_not_match) AS sd_not,
         AVG(pct_acima_p95) AS media_p95,
         STDDEV_SAMP(pct_acima_p95) AS sd_p95
  FROM dia_agregado
),
-- Calcula z-scores individuais e índice ponderado
final AS (
  SELECT d.dt,
         (d.taxa_cobertura_bio - e.media_bio) / NULLIF(e.sd_bio, 0) AS
z_bio,
         (d.taxa_not_match - e.media_not) / NULLIF(e.sd_not, 0) AS z_not,
         (d.pct_acima_p95 - e.media_p95) / NULLIF(e.sd_p95, 0) AS z_p95,
         -- pesos: NOT_MATCH 0.5, biometria 0.3, produtividade outlier 0.2
         (
           0.5 * (
             (d.taxa_not_match - e.media_not) / NULLIF(e.sd_not, 0)
           ) + 0.3 * (
             (d.taxa_cobertura_bio - e.media_bio) / NULLIF(e.sd_bio, 0)
           ) + 0.2 * (
             (d.pct_acima_p95 - e.media_p95) / NULLIF(e.sd_p95, 0)
           )
         ) AS indice_bruto
  FROM dia_agregado d
  CROSS JOIN estat e
),
-- Normaliza o índice para 0-1
normalizado AS (
  SELECT dt,
         z_bio,
         z_not,
         z_p95,
         indice_bruto,
         -- 1) divide por 5 para reduzir escala (assume "5 desvios-padrão acima" como risco extremo)
         -- 2) desloca +0.5 para recentrar
         -- 3) trunca entre 0 e 1
         ROUND(
           LEAST(
             GREATEST(indice_bruto / 5 + 0.5, 0),
             1
           ),
           3
         ) AS indice_suspeita_normalizado,
  FROM final
)
SELECT dt,
       ROUND(z_not, 2) AS z_notmatch,
       ROUND(z_bio, 2) AS z_bio,
       ROUND(z_p95, 2) AS z_prod_outlier,
       ROUND(indice_bruto, 2) AS indice_bruto_ponderado,
       indice_suspeita_normalizado,
  FROM normalizado
ORDER BY dt;
```



E. Em quais dias provavelmente tivemos um aumento do comportamento indevido “empréstimo de conta”?

Queries executadas:

```
-- E2: Correlação entre dia da semana e taxa de
NOT_MATCH
WITH diario AS (
  SELECT DATE(Event_Dt) AS dt,
    EXTRACT(
      DAYOFWEEK
      FROM Event_Dt
    ) AS dow, -- 1 = segunda, 7 = domingo
    COUNT(*) AS sessoes,
    COUNTIF(Final_Status = 'NOT_MATCH') AS not_match
  FROM
    `ifood-prev-fraude-case.fraude_biometria.vw_biometry
    _enriched`
  WHERE Event_Dt IS NOT NULL
  GROUP BY dt,
    dow
  HAVING sessoes >= 10
),

taxa_por_dia_semana AS (
  SELECT dow,
    ROUND(AVG(SAFE_DIVIDE(not_match, sessoes)), 4) AS not_match_rate_media,
    COUNT(*) AS dias_agrupados
  FROM diario
  GROUP BY dow
)
SELECT dow,
  CASE
    dow
    WHEN 1 THEN 'Segunda'
    WHEN 2 THEN 'Terça'
    WHEN 3 THEN 'Quarta'
    WHEN 4 THEN 'Quinta'
    WHEN 5 THEN 'Sexta'
    WHEN 6 THEN 'Sábado'
    WHEN 7 THEN 'Domingo'
  END AS dia_semana,
  not_match_rate_media,
  dias_agrupados,
  CORR(dow, not_match_rate_media) OVER() AS corr_dia_notmatch
FROM taxa_por_dia_semana
ORDER BY dow;
```





Conclusões

1. Confiabilidade dos provedores:

O sistema apresenta **alta disponibilidade geral (99,4%)**, mas com **falhas concentradas nos fallbacks** (B e C). A resiliência depende da estabilização desses provedores.

2. Perfis críticos de entregadores:

As categorias **Bronze, Prata e Iniciantes concentram as maiores taxas de falha e cancelamento**, sugerindo fatores operacionais problemáticos (dificuldades técnicas e/ou inexperiência por uso ocasional) e **possíveis usos indevidos de conta** no mesmo aparelho.

3. Ajuste de limiar biométrico:

Elevar o limiar de 0,80 para 0,90 **aumenta a segurança**, mas **reduz a eficiência operacional em ~33%**, exigindo **calibração segmentada por perfil de risco**.

4. Falhas vs. cancelamentos:

Não há correlação significativa ($r \approx 0$), indicando que **cancelamentos parecem refletir mais problemas operacionais que tentativas de fraude**. No entanto, análises em um período de tempo e amostras maiores são recomendadas.

5. Padrões de “empréstimo de conta”:

O índice de suspeita combinada (falhas biométricas, checagens e produtividade atípica) destacou **17/08 como o dia de maior atividade suspeita de empréstimo de conta**, havendo **maior risco de compartilhamento em dias úteis**.



Próximos Passos

- **Prevenção à Fraude**
 - **Ampliar o monitoramento em dias críticos**, cruzando **geolocalização, device e frequência de login**.
 - Ampliar o **Índice de Suspeita de Compartilhamento de Contas** para um **modelo preditivo**, incorporando **device fingerprinting e biometria comportamental**.
 - Priorizar **biometria passiva** (liveness detection) em **segmentos Bronze/Prata/Iniciantes**, **reduzindo falsos negativos e melhorando a retenção da frota legítima**, especialmente entre entregadores com dispositivos de menor qualidade.
- **Engenharia**
 - **Evoluir de limiar fixo** (0.8/0.9) para **autenticação adaptativa baseada em risco**, ajustando dinamicamente por dispositivo, localização e comportamento histórico.
 - Revisar **tolerância e fallback** dos **provedores B e C**, formalizando SLAs de performance com métricas de **disponibilidade e latência**.
- **Operações**
 - **Reforçar orientações de selfie e boas práticas no app**, com foco em categorias de maior incidência de falha na sessão (Bronze, Prata e Iniciantes).
- **Governança de dados**
 - Automatizar **alertas de picos de falha e Índice de Suspeita de Compartilhamento de Contas**, viabilizando **monitoramento contínuo e dashboards de risco**.



Fortalecer a biometria é investir em um ecossistema mais seguro, ágil e humano, onde a tecnologia protege o negócio sem interromper quem faz o iFood acontecer todos os dias.



Obrigada :)

Silvania G. Correia

✉ silvania.goularte@gmail.com

 [silvania-correia](#)

