

Plano de Melhorias — Dashboard Mercado DuBairro

Data: 14/02/2026

Versão: 1.0

Visão Geral

Este documento organiza todas as melhorias identificadas por Raquel + insights do Gemini + sugestões estratégicas do Claude, priorizadas por impacto e urgência.

● PRIORIDADE ALTA — Resolver Agora

1. Logo do Mercado DuBairro

Problema: A logo do mercado não está aparecendo no dashboard.

Solução: Adicionar a logo como imagem na sidebar ou no topo de cada página.

```
python

# Na sidebar (aparece em todas as páginas)
from PIL import Image
logo = Image.open("assets/logo_dubairro.png")
st.sidebar.image(logo, use_container_width=True)
```

2. Tooltips (💡) em TODOS os gráficos

Problema: Só alguns gráficos têm a "lampadinha" explicando os dados. Quem não é analista pode se perder.

Solução: Adicionar `st.info()` ou `st.expander()` explicativo em CADA gráfico.

Modelo padrão para cada gráfico:

```
python

with st.expander("💡 Como interpretar este gráfico?"):
    st.markdown("""
        **O que mostra:** [descrição simples]
        **Como ler:** [instruções práticas]
        **Por que importa:** [o que fazer com essa informação]
        **Exemplo:** [caso concreto]
    """)
```

Exemplo para o gráfico de Sazonalidade 2025:

```
python
```

```

with st.expander("💡 Como interpretar o Índice de Sazonalidade?"):
    st.markdown("****")
    **O que mostra:** O padrão de vendas ao longo dos meses do ano.  

    Um índice de **1.0 = média** (mês normal).  

    Acima de 1.0 = mês acima da média. Abaixo de 1.0 = mês fraco.

    **Como ler:**  

    - Índice **1.15** → esse mês fatura 15% ACIMA da média anual  

    - Índice **0.85** → esse mês fatura 15% ABAIXO da média anual

    **Por que importa:** Ajuda a planejar estoque, promoções e caixa.  

    Meses com índice baixo = precisa de promoção para aquecer vendas.  

    Meses com índice alto = garantir estoque cheio para não perder venda.

    **Exemplo prático:** Se dezembro tem índice 1.30 e janeiro 0.80,  

    isso significa que dezembro fatura 30% acima da média e janeiro  

    fatura 20% abaixo. Ação: planejar promoção de janeiro em dezembro!
    ****)

```

3. Clareza Temporal — "É 2025 completo ou Janeiro/26?"

Problema: Confusão entre dados do ano inteiro vs mês específico.

Soluções combinadas:

a) Título descritivo em cada gráfico:

```

python

# ❌ Ruim
st.subheader("Faturamento 2025")

# ✅ Bom
st.subheader("Faturamento Mensal — Ano 2025 Completo (Jan-Dez)")
# ou
st.subheader("Faturamento — Janeiro/2026 vs Janeiro/2025")

```

b) Comparação mês a mês (a melhor solução):

```
python
```

```

# Criar coluna de comparação MêsAno
df['MesAno_2025'] = 'Jan/25', 'Fev/25', ...
df['MesAno_2026'] = 'Jan/26', 'Fev/26', ...

# Gráfico de barras lado a lado
fig = go.Figure()
fig.add_trace(go.Bar(name='2025', x=meses, y=valores_2025, marker_color="#636EFA"))
fig.add_trace(go.Bar(name='2026', x=meses, y=valores_2026, marker_color="#00CC96"))
fig.update_layout(barmode='group', title='Comparação Mês a Mês: 2025 vs 2026')

```

c) Indicador visual de período:

```

python

# No topo da página, mostrar claramente o período
col1, col2 = st.columns(2)
with col1:
    st.caption("📅 Último mês fechado: **Janeiro/2026**")
with col2:
    st.caption("📊 Comparando com: **Janeiro/2025**")

```

● PRIORIDADE MÉDIA — Implementar em Fevereiro

4. Página "Visão Futurista" / Previsões

Problema: Faltam informações preditivas que ajudem a enxergar o futuro.

Solução: Criar nova página ou expandir a existente com:

a) Projeção de Faturamento (próximos 3-6 meses):

```
python
```

```

# Usando média móvel + sazonalidade
from statsmodels.tsa.holtwinters import ExponentialSmoothing

model = ExponentialSmoothing(
    serie_faturamento,
    seasonal='mul',
    seasonal_periods=12
).fit()
previsao = model.forecast(6) # próximos 6 meses

# Gráfico com faixa de confiança
fig.add_trace(go.Scatter(
    name='Previsão', x=meses_futuros, y=previsao,
    line=dict(dash='dash', color='gold')
))

```

b) Meta de Faturamento Dinâmica:

```

python

# Calcular meta baseada em custo fixo + margem desejada
meta_minima = custo_fixo / margem_media # break-even
meta_ideal = meta_minima * 1.3 # 30% acima do break-even

st.metric("Meta Mínima (Break-even)", f'R$ {meta_minima:.2f}')
st.metric("Meta Ideal (+30%)", f'R$ {meta_ideal:.2f}')

# Gauge / velocímetro visual
fig = go.Figure(go.Indicator(
    mode="gauge+number+delta",
    value=faturamento_atual,
    title={'text': "Faturamento vs Meta"},
    gauge={
        'axis': {'range': [0, meta_ideal * 1.5]},
        'bar': {'color': "green"},
        'steps': [
            {'range': [0, meta_minima], 'color': "red"},
            {'range': [meta_minima, meta_ideal], 'color': "yellow"},
            {'range': [meta_ideal, meta_ideal*1.5], 'color': "lightgreen"}
        ],
        'threshold': {'value': meta_ideal}
    }
))

```

c) Sazonalidade por Categoria (não só faturamento total):

```
python
```

```
# Sazonalidade de Bebidas, Hortifruti, Padaria, etc.  
# Quais categorias explodem em quais meses?  
for categoria in categorias_top:  
    sazonalidade = calcular_indice_sazonal(df[df['Categoria'] == categoria])  
    # Mostra: "Cerveja tem índice 1.8 em Dezembro → dobrar estoque!"
```

d) Cenários "E se?" para Março:

```
python
```

```
st.subheader("🔮 Cenários para Março/2026")  
  
# Cenário Pessimista / Realista / Otimista baseado em sazonalidade  
indice_marco = sazonalidade['Mar']  
base = faturamento_jan26  
  
cenarios = {  
    'Pessimista': base * indice_marco * 0.9,  
    'Realista': base * indice_marco,  
    'Otimista': base * indice_marco * 1.1  
}
```

e) Alertas Inteligentes:

```
python
```

```
# Produtos com estoque crítico vs demanda prevista  
# Categorias em queda que precisam de ação  
# Produtos "Peso Morto" que devem sair de linha  
  
st.warning("⚠️ Cerveja Marca X: vendas caíram 30% nos últimos 3 meses")  
st.success("🚀 Hortifruti orgânico: crescimento de 45% — expandir espaço!")
```

5. Simulador "E se?" na Sidebar (Sugestão Gemini)

```
python
```

```

st.sidebar.subheader("💡 Simulador")
custo_fixo_input = st.sidebar.number_input(
    "Custo Fixo Mensal (R$)", value=16913.46, step=100.0
)
margem_desejada = st.sidebar.slider(
    "Margem Desejada (%)", 10, 50, 25
)
# Recalcular metas e break-even em tempo real
meta_faturamento = custo_fixo_input / (margem_desejada / 100)
st.sidebar.metric("Meta de Faturamento", f'R$ {meta_faturamento:.2f}')

```

6. Filtros Globais por Categoria (Sugestão Gemini)

```

python

# Na sidebar
categorias = df['Categoria'].unique().tolist()
categorias_selecionadas = st.sidebar.multiselect(
    "Filtrar por Categoria",
    categorias,
    default=categorias # todas selecionadas por padrão
)
# Aplicar filtro global
df_filtrado = df[df['Categoria'].isin(categorias_selecionadas)]

```

● PRIORIDADE FUTURA — Março em diante

7. Análise de Cesta (Cross-Selling) — Sugestão Gemini

Se você tem o ID da transação/documento, é possível descobrir:

- "Quem compra carvão também leva cerveja (80%)"
- "Quem compra pão francês leva frios (65%)"
- Ação: criar kits promocionais baseados em dados!

8. Tolerância Visual nos KPIs (Sugestão Gemini)

Variações entre -2% e +2% deveriam ser amarelas (neutras), não verde/vermelho. No varejo, uma variação de 0.5% não significa nada.

```

python

```

```

def cor_variacao(var):
    if var > 2:
        return "green", "▲"
    elif var < -2:
        return "red", "▼"
    else:
        return "orange", "●" # estável

```

9. Segurança do Código (Sugestão Gemini)

Garantir valores padrão quando `yoy_mes` estiver vazio para evitar erros de NaN.

python

```

# Valores padrão seguros
mes_ref = "N/A"
var_receita = 0.0
var_lucro = 0.0

if not yoy_mes.empty:
    mes_ref = yoy_mes.iloc[-1]['Mes']
    var_receita = yoy_mes.iloc[-1]['Var_Receita']
    # etc...

```

Plano de Ação — Final de Fevereiro

Semana	O que fazer	Resultado esperado
Sem 1 (17-21/fev)	Logo + Tooltips em todos os gráficos	Dashboard mais profissional e comprehensível
Sem 1 (17-21/fev)	Corrigir títulos temporais + comparação mês a mês	Zero confusão sobre períodos
Sem 2 (24-28/fev)	Página Futurista: projeções + cenários + metas	Visão clara para planejar Março
Sem 2 (24-28/fev)	Simulador "E se?" + Filtros por categoria	Interatividade e análise mais profunda

Direção para Março

Com base na sazonalidade e nos dados preditivos, o plano de Março deveria focar em:

1. **Categorias em crescimento** → aumentar espaço e estoque
2. **Produtos "Peso Morto"** → promoção de queima ou descontinuar

3. **Cross-selling** → kits promocionais baseados em dados de cesta
 4. **Meta agressiva mas realista** → baseada na projeção sazonal
 5. **Ações por semana** → dashboard mostrando o que fazer cada semana de março
-

Próximo Passo

Para eu implementar as melhorias, preciso que você me envie:

1. O arquivo `.py` do dashboard (ou os arquivos das páginas)
2. Uma amostra do Excel/CSV com os dados (para testar)

Com isso em mãos, eu codifico tudo e te entrego pronto para deploy!