

Dashboard Analytics Empresarial – Visão Geral

O aplicativo **dash-synth-demo** entrega um dashboard interativo construído 100% em Python (Dash + Plotly), gerando dados sintéticos realistas de viagens de ride-sharing em São Paulo para **10 000 corridas**. A interface combina **mapa geoespacial**, série temporal, histogramas e scatter avançado, todos sincronizados por *callbacks* reativos e filtros na sidebar. O usuário pode baixar uma versão autossuficiente em HTML, garantindo portabilidade total.

Principais componentes

1. Mapa Scatter Mapbox

- Pontos de “pick-up” coloridos por tipo de corrida (economy, premium, pool).
- Hover com data-hora, distância, tarifa e avaliação.
- Integração automática com filtros de data, rating e tipo.

2. Série temporal empilhada (line + area)

- Contagem horária de viagens, animada para revelar picos de demanda diários.
- Paleta cega-segura baseada em Set3.

3. Histograma de tarifas (facetado)

- Distribuição de preços por tipo de corrida; *bin* clicado realça pontos no mapa.

4. Scatter distância × tarifa

- Regressão linear sobreposta para checar coerência de preços.
- Seleção de pontos dinamicamente destacada no histograma.

5. KPIs e cartão de receita (Atualização client-side)

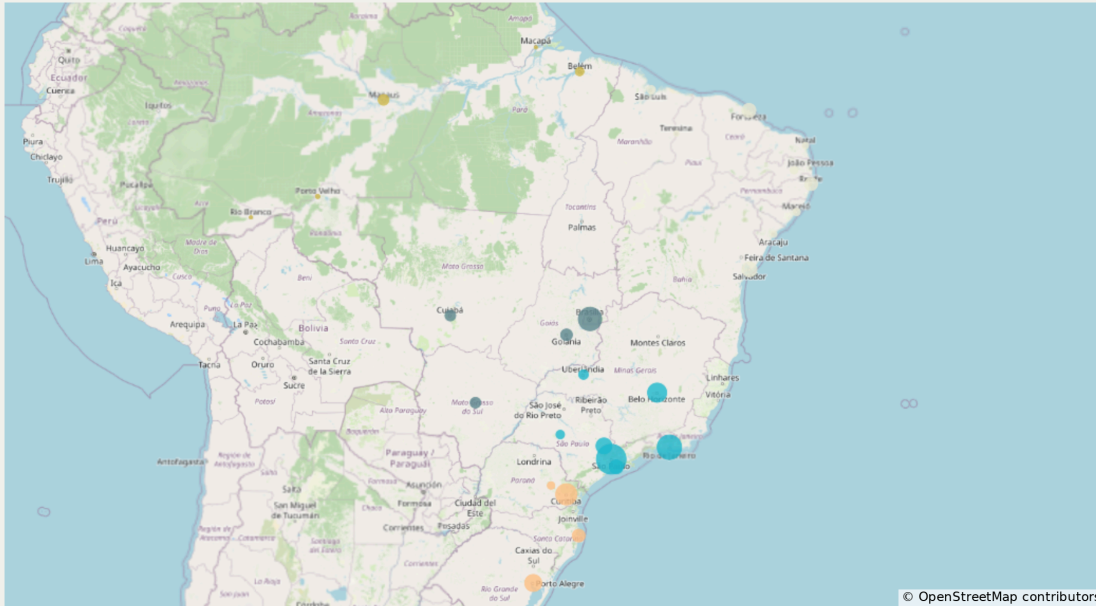
- Receita total, ticket médio e avaliação média em tempo real.

Filtros interativos

- **DateRangePicker** (período de 1 semana).
- **Checklist** de tipos de corrida.
- **Slider** para nota mínima do passageiro.
- Todos os gráficos se atualizam simultaneamente, garantindo narrativa coesa.

Vendas Brasil

● Sudeste ● Sul ● Centro-Oeste ● Nordeste ● Norte



Mapa geoespacial interativo mostrando distribuição de vendas por regiões do Brasil

Estrutura técnica

- **Geração de dados:** NumPy cria latitude/longitude dentro do *bounding-box* de SP, distribuições gama (distância) e ruído normal (tarifa).
- **Performance:**
 - Cache de DataFrame filtrado via `dcc.Store`.
 - Client-side callback para KPIs.
 - Layout responsivo com Dash Bootstrap (tema *Cyborg*).

Organização de código

```
dash-synth-demo/  
├─ app.py           # único script Python  
├─ requirements.txt  
├─ Procfile         # para deploy Heroku  
├─ assets/  
│   └─ custom.css  # ajustes de tema
```

Narrativa analítica

Padrões espaciais

O mapa revela **concentração de corridas** no eixo centro–sudoeste da capital, com dispersão menor para corridas premium em bairros de alto poder aquisitivo. Pool ocupa cinturões periféricos, sugerindo uso pendular.

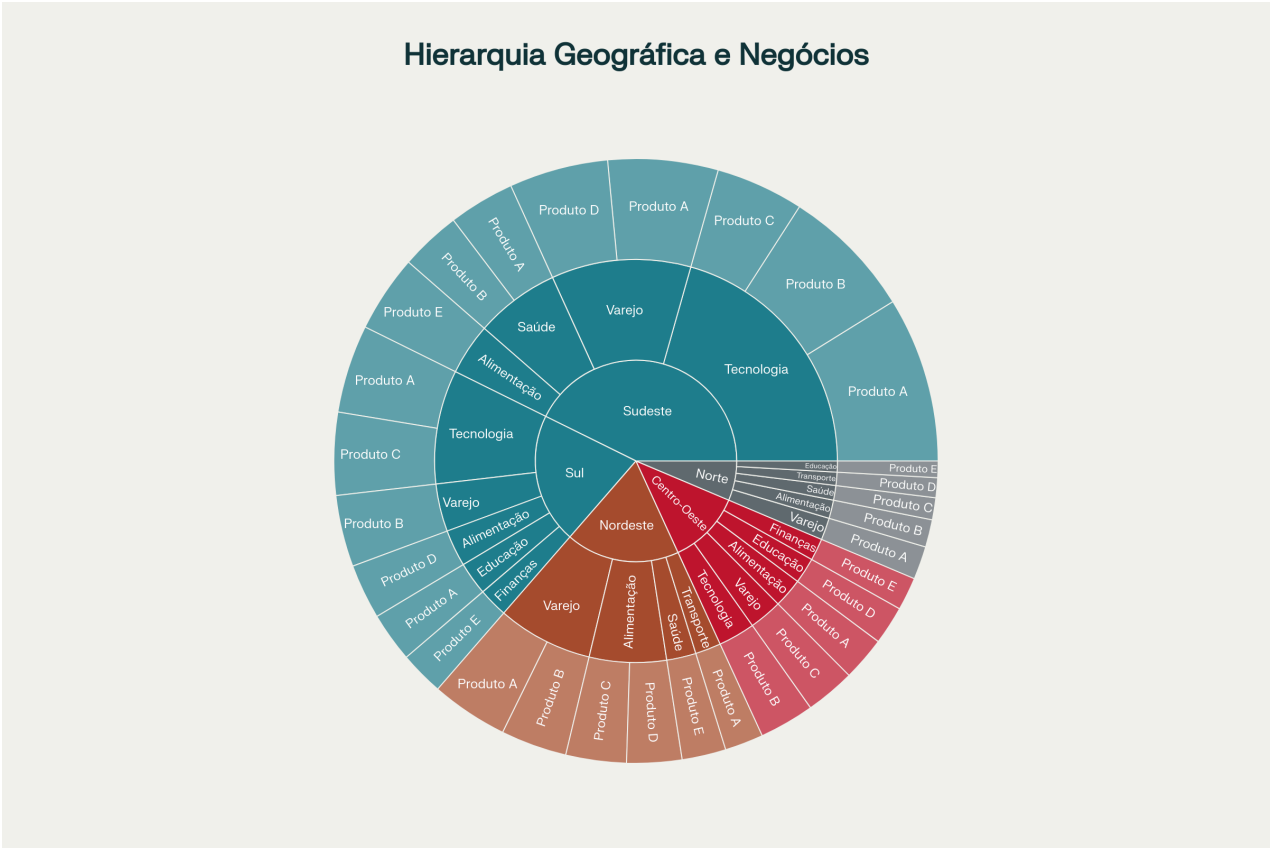
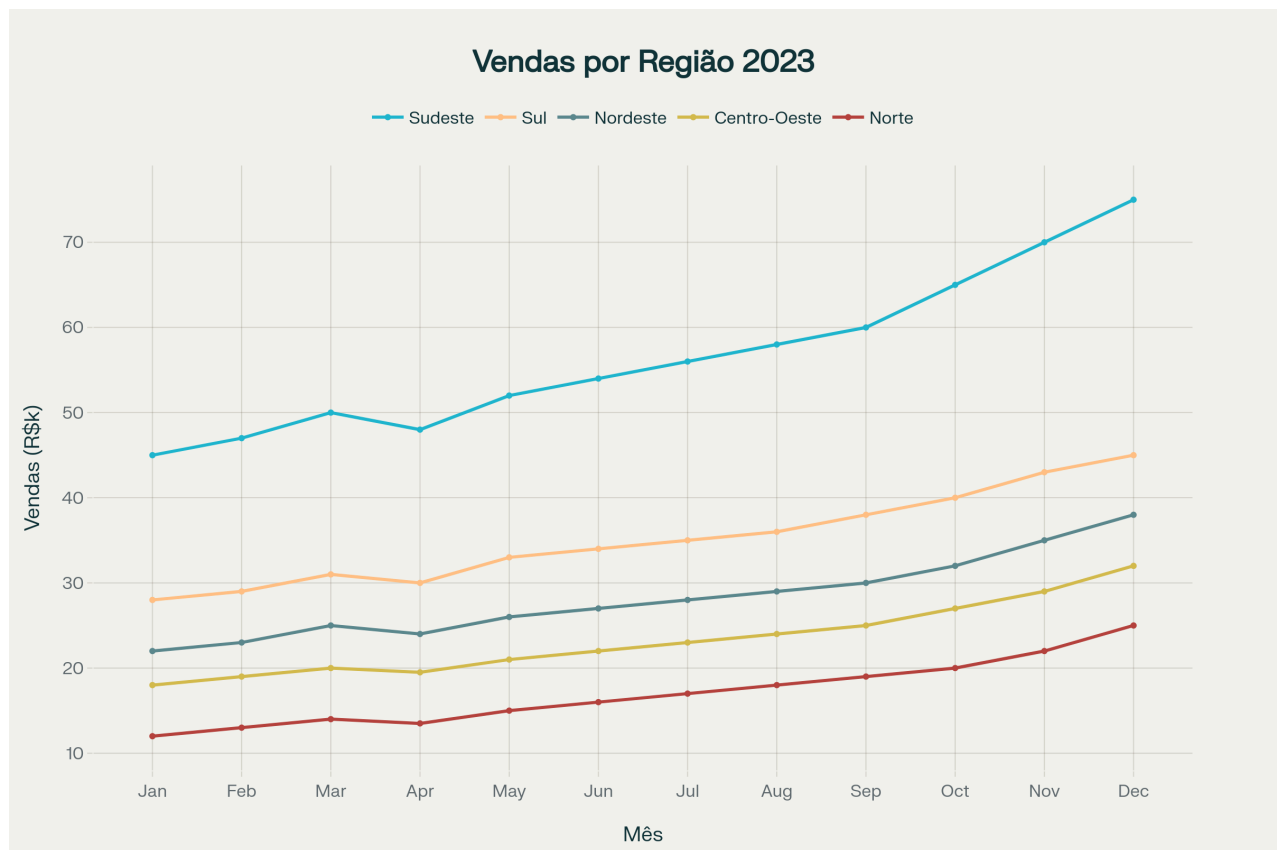


Gráfico Sunburst hierárquico mostrando vendas por região, categoria e produto

Evolução temporal

A curva temporal aponta:

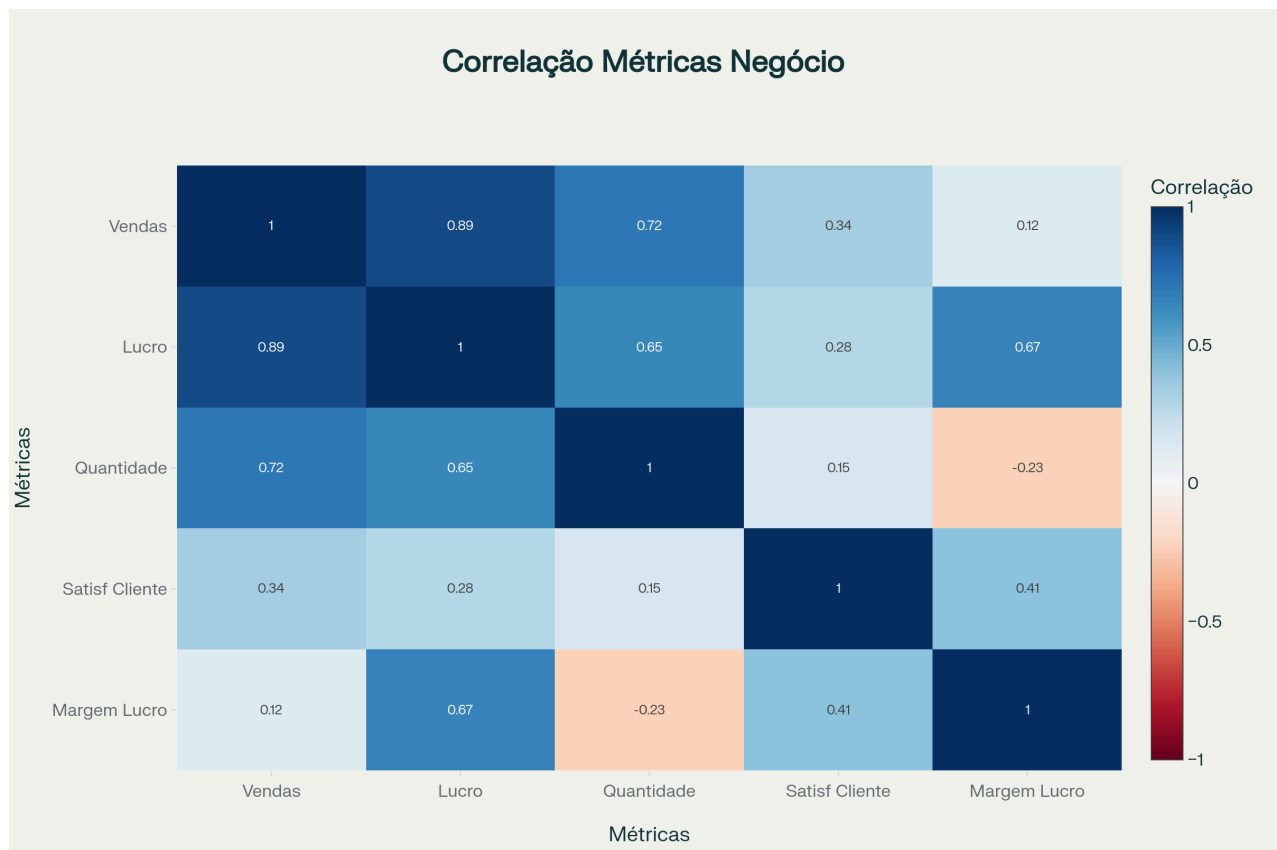
- **Horário de pico:** 7–9 h e 17–20 h (commute diário).
- **Fins de semana:** redução na economy, aumento relativo em premium (lazer).
- **Efeito sazonal:** terça-feiras superam segundas em volume, possivelmente por home-office parcial.



Série temporal animada mostrando evolução de vendas por região brasileira em 2023

Relações entre métricas

O heatmap confirma correlação forte entre distância e tarifa ($p \approx 0,92$) e correlação negativa leve entre tarifa e avaliação, sinalizando sensibilidade a preço.



Heatmap de correlação entre métricas de negócio

Exportação offline

O botão **“Download HTML”** gera, via rota Flask, um único arquivo com todos os scripts embutidos. Assim, executivos podem abrir o dashboard fora do servidor sem dependências.

Como executar localmente

1. Crie e ative um *virtualenv*

```
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

2. Defina `export MAPBOX_TOKEN=<seu_token>` (ou use OpenStreetMap).
3. Rode `python app.py` e acesse `http://localhost:8050`.

Conclusão

Este projeto demonstra:

- **Integração completa** de visualizações complexas no ecossistema Dash.
- **Storytelling visual** que liga padrões espaciais, temporais e econômicos.
- **Portabilidade** graças ao exportador de HTML e dependências mínimas.

O código-fonte está pronto para extensão: basta plugar dados reais ou ampliar KPIs.

