



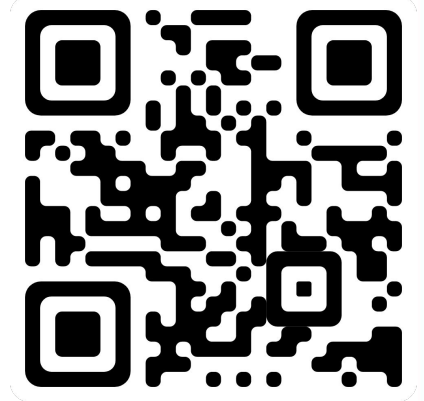
# Previsão de Séries Temporais Epidemiológicas na Prática

**Prof. Dr. Ramon Gomes da Silva**

Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS)

# Sobre mim

- **Bacharel em Engenharia de Produção**  
Universidade do Estado do Pará - UEPA
- **Graduação Sanduíche em Eng. Industrial**  
University of Southern Indiana - USI
- **Mestre em Pesquisa Operacional**  
Pontifícia Universidade Católica do Paraná - PUCPR
- **Doutor em Machine Learning e Data Science**  
Pontifícia Universidade Católica do Paraná - PUCPR
- **Professor e Pesquisador**  
PPG em Eng. de Produção e Sistemas - PPGEPS / PUCPR



[ramongss.github.io](https://ramongss.github.io)

# Agenda

## 01 Contextualização & Fundamentos

Por que Prever Séries Epidemiológicas?

Conceitos Fundamentais de Séries Temporais

Desafios Específicos de Dados Epidemiológicos

## 02 Caixa de Ferramentas

Nixtlaverse: Ecossistema de *Forecasting* em Python

Modelos Estatísticos no StatsForecast

ML e Deep Learning para *Forecasting*

HierarchicalForecast: O Coração do Framework

## 03 Hands-on



# 01 Contextualização & Fundamentos

# Por que Prever Séries Epidemiológicas?

- **Gestão Hospitalar**

Antecipar internações, demanda por leitos UTI e insumos críticos.

- **Campanhas de Vacinação**

Planejar cobertura vacinal antes dos picos sazonais.

- **Alertas Precoces**

Detectar surtos antes que se tornem emergências de saúde pública.

- **Alocação de Recursos**

Distribuir agentes de saúde e medicamentos com base em projeções.

- **Políticas Públicas**

Subsidiar decisões de lockdown, isolamento e medidas de controle.

- **Vigilância Epidemiológica**

Acompanhar tendências de doenças transmissíveis em tempo quase-real.



# Conceitos Fundamentais de Séries Temporais

- **Estacionaridade**

Média e Variância constantes ao longo do tempo

- **Tendência**

Direção geral de longo prazo – crescimento ou declínio sistemático

- **Sazonalidade**

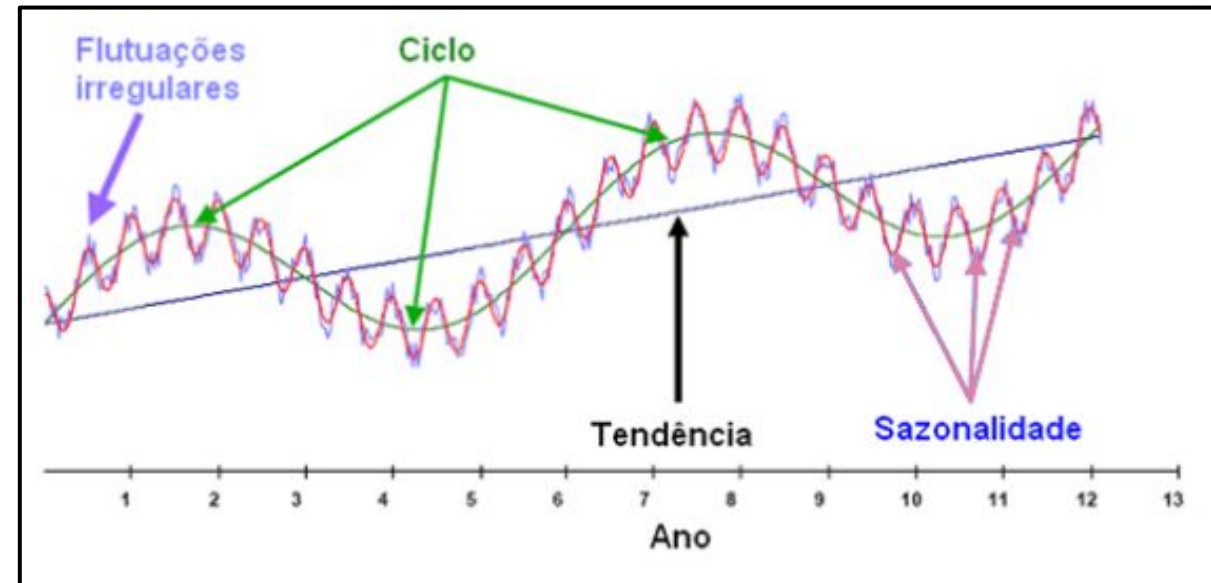
Padrões recorrentes em intervalos fixos (semana, mês, ano epidemiológico)

- **Ciclicidade**

Padrões recorrentes em intervalos não regulares (2 anos, 5 anos etc)

- **Componente Irregular**

Variação devido fatores imprevisíveis, sem repetição de padrão



[1] <https://otexts.com/fpppy/>

# Desafios Específicos de Dados Epidemiológicos

## Atraso de Notificação

Casos são notificados dias ou semanas após o evento

## Sazonalidade Múltipla

Dengue tem padrão anual; influenza tem padrão semanal + anual; interação dificulta modelos simples

## Sub-notificação

Dados oficiais subestimam casos reais, especialmente em regiões com baixa cobertura laboratorial

## Quebras Estruturais

Epidemias são eventos raros e abruptos — modelos treinados em períodos endêmicos falham

## Heterogeneidade Espacial

Doenças se propagam de forma heterogênea entre municípios, estados e regiões

## Dependência Hierárquicas

Brasil > Região > UF > Município: previsões inconsistentes entre níveis são um problema real



# **02**

# **Caixa de**


# **Ferramentas**

# Nixtlaverse: Ecossistema de *Forecasting* em Python




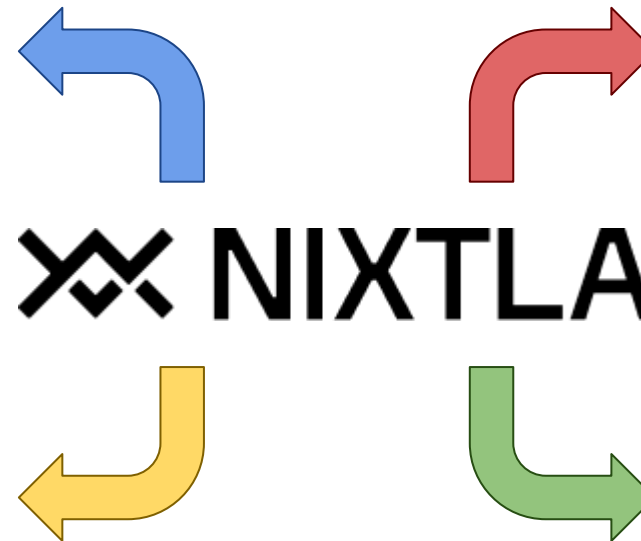
## StatsForecast

ARIMA, ETS, Theta  
Modelos estatísticos clássicos




## NeuralForecast

LSTM, NHITS, NBEATS, TFT  
Redes neurais para séries



## MLForecast

LightGBM, XGBoost, Ridge  
Feature engineering



## HierarchicalForecast

Reconciliação, Bottom-Up  
MinT, Top-Down

[2] <https://nixtlaverse.nixtla.io/>

# ⚡ Modelos Estatísticos no StatsForecast

## AutoArima

$ARIMA(p,d,q)(P,D,Q)$

Seleção automática de ordens. Lida com sazonalidade e tendência via *differencing*.

*ex. Dengue semanal por município*

## AutoETS

*Error, Trend, Seasonal*

Suavização exponencial generalizada. Rápido e robusto para sazonalidade estocástica.

*ex. Influenza mensal por UF*

## AutoTheta

*Decomposição Theta*

Campeão em competições M3/M4. Decompõe tendência linear + sazonalidade simples.

*ex. SRAG semanal nacional*

## CES

*Complex Exponential Smoothing*

Estende ETS com componentes complexos. Captura ciclos epidemiológicos longos.

*ex. COVID-19 ondas pandêmicas*

# ML e Deep Learning para *Forecasting*



## MLForecast

### LightGBM / XGBoost

Gradient boosting – robusto, lida com features externas (temperatura, chuva, mobilidade)

### Feature Engineering Automático

Lags, rolling stats, dummies sazonais criados automaticamente pela biblioteca

### Variáveis Exógenas

Incorpora covariáveis epidemiológicas (população, IDH, cobertura vacinal)

### Vantagem

Generaliza bem entre séries similares  
Integração com scikit-learn e outros pacotes



## NeuralForecast

### NHITS

Neural Hierarchical Interpolation – top performer em múltiplos horizontes, ótimo para epidemias

### NBEATS

Interpretável: separa tendência de sazonalidade – excelente explicabilidade

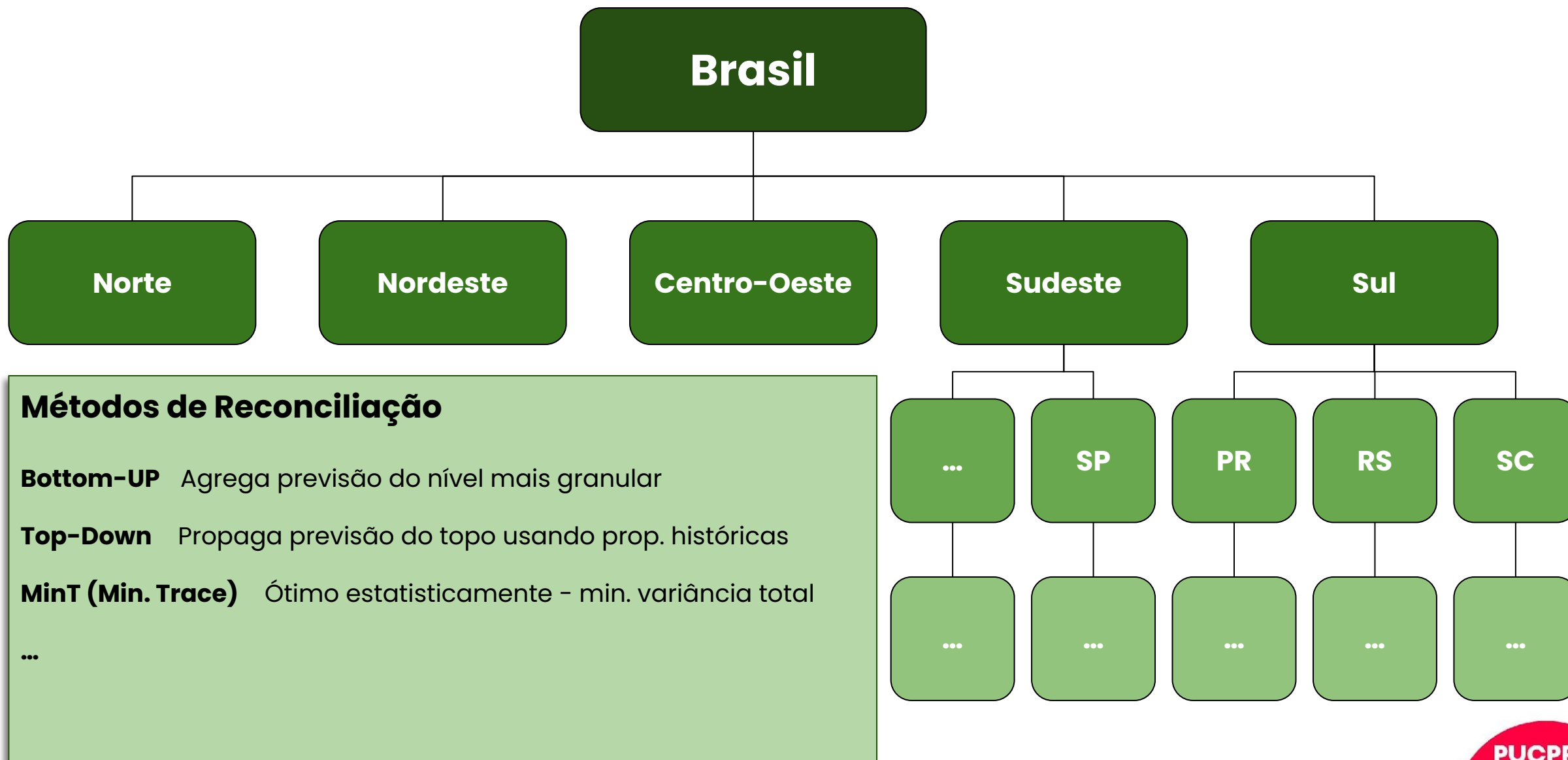
### TFT (Temporal Fusion Transformer)

Suporta variáveis estáticas (UF, bioma) + dinâmicas – ideal para dados espacialmente heterogêneos

### Quando usar?

Grandes volumes de dados, padrões complexos, múltiplas covariáveis

# 👑 Hierarchical Forecast: O Coração do Framework



# Referências

- [1] R. J. Hyndman, G. Athanasopoulos, A. Garza, C. Challu, M. M. Canseco, and K. G. Olivares, Forecasting: Principles and Practice, the Pythonic Way. Melbourne, Australia: OTexts, 2025. [Online]. Available: <https://otexts.com/fpppy/>
- [2] "Nixtlaverse," Nixtla. Accessed: Apr. 20, 2026. [Online]. Available: <https://nixtlaverse.nixtla.io>



# **03**

## **Hands-on**

# Hands-on: Hora de por a mão na massa

## 01 Acesse e baixe os dados

ramongss.github.io > Talks > Card Palestra > Dataset > data.csv

## 02 Crie um Jupyter Notebook (pré-requisito: Conta Google)

<https://colab.research.google.com>

## 03 Subir dados no Google Colab e executar

```
import pandas as pd
data = pd.read_csv('data.csv')
data.head(5)
```



[ramongss.github.io](https://ramongss.github.io)

**Obrigado!**

**Perguntas?**