

Juro Real Ex-Ante e Juro Neutro no Brasil: Uma Análise Baseada em Expectativas de Mercado

Walter C. Neto

12 de dezembro de 2025

Resumo

Este documento apresenta uma análise empírica da evolução do juro real ex-ante e do juro neutro no Brasil, com base em expectativas de mercado coletadas junto ao Banco Central do Brasil (BCB) e dados de mercado disponibilizados pelo IPEA/B3. O objetivo é avaliar o grau de aperto ou afrouxamento da política monetária ao longo do tempo, comparando o juro real observado com uma proxy de taxa real neutra construída a partir de expectativas de médio prazo.

Sumário

1	Introdução	3
2	Fundamentos Teóricos	3
2.1	Juro Real Ex-Ante	3
2.2	Juro Neutro	3
3	Base de Dados e Metodologia	3
3.1	Fontes de Dados	3
3.2	Procedimento Empírico	4
4	Resultados Empíricos	4
5	Interpretação Econômica	5
6	Conclusão	5
7	Fontes de Dados	5
A	Apêndice A – Código em Python	6

1 Introdução

A taxa de juros real desempenha papel central na condução da política monetária, influenciando decisões de consumo, investimento, crédito e alocação de capital. Em particular, a comparação entre o **juro real ex-ante** e o **juro real neutro** fornece uma medida objetiva do grau de estímulo ou restrição imposto pela política monetária em determinado período.

Enquanto o juro real ex-ante reflete as condições efetivas enfrentadas pelos agentes econômicos, o juro neutro representa uma taxa de equilíbrio de médio prazo, consistente com inflação estável e produto no nível potencial.

2 Fundamentos Teóricos

2.1 Juro Real Ex-Ante

O juro real ex-ante corresponde à taxa de juros nominal descontada da inflação esperada. Formalmente, utilizando a equação de Fisher:

$$r_t^{ex-ante} = \left(\frac{1 + i_t}{1 + \mathbb{E}_t[\pi_{t+1}]} \right) - 1 \quad (1)$$

onde:

- i_t é a taxa nominal observada (neste estudo, o swap DI pré 360 dias);
- $\mathbb{E}_t[\pi_{t+1}]$ é a expectativa de inflação acumulada para os próximos 12 meses.

Essa medida captura o custo real do dinheiro antecipado pelos agentes econômicos.

2.2 Juro Neutro

O juro neutro é a taxa real consistente com uma economia operando em seu potencial, sem gerar pressões inflacionárias ou deflacionárias. Como essa taxa não é observável diretamente, constrói-se uma **proxy empírica** a partir de expectativas de mercado de médio prazo.

Neste trabalho, o juro neutro é aproximado por:

- Expectativa da taxa Selic nominal para três anos à frente;
- Expectativa do IPCA para o mesmo horizonte;
- Aplicação da equação de Fisher para obtenção da taxa real.

3 Base de Dados e Metodologia

3.1 Fontes de Dados

Os dados utilizados são provenientes de fontes oficiais e amplamente utilizadas na análise macroeconômica:

- **Banco Central do Brasil (BCB):**
 - Sistema de Expectativas de Mercado (Focus);

- Expectativas anuais de IPCA e Selic;
- Expectativas de inflação acumulada em 12 meses.
- **IPEA / B3:**
 - Taxa de swap DI pré 360 dias.

3.2 Procedimento Empírico

O código implementa os seguintes passos:

1. Coleta das expectativas anuais de IPCA e Selic;
2. Seleção das expectativas com horizonte de três anos à frente;
3. Construção do juro neutro via equação de Fisher;
4. Coleta da taxa de swap DI pré 360 dias;
5. Cálculo do juro real ex-ante utilizando inflação esperada para 12 meses;
6. Agregação mensal dos dados;
7. Geração do gráfico comparativo.

4 Resultados Empíricos



Figura 1: Juro Real Ex-Ante versus Juro Neutro

O Gráfico 1 evidencia três regimes distintos:

- **2016–2019:** afrouxamento monetário;
- **2020–2021:** estímulo monetário excepcional;
- **2022 em diante:** forte aperto monetário.

5 Interpretação Econômica

Quando o juro real ex-ante se encontra acima do juro neutro, a política monetária atua de forma contracionista. Juros reais abaixo do neutro indicam estímulo monetário.

6 Conclusão

A evidência empírica sugere que o Brasil atravessou, desde 2022, um dos ciclos de maior aperto monetário real da sua história recente.

7 Fontes de Dados

- Banco Central do Brasil – Sistema de Expectativas de Mercado (Focus): <https://www.bcb.gov.br>
- IPEADATA / B3 – Taxas de Swap DI: <https://www.ipeadata.gov.br>

A Apêndice A – Código em Python

O código completo utilizado na análise encontra-se no arquivo `juros_reais.py`, localizado na mesma pasta deste documento.

```
import pandas as pd
import datetime
from matplotlib import pyplot as plt
import seaborn as sns

from bcb import Expectativas
from datetime import timedelta

# Cria fun o para subtrair os dias do ano
def years_sub(days):
    sub_year_days = pd.to_datetime(datetime.datetime.today().strftime('%Y
    -%m-%d')) - timedelta(days = 10 * 365)
    sub_year_days = sub_year_days.strftime("%Y-%m-%d")
    return sub_year_days

# Cria fun o para calcular a data de refer ncia a partir da data de
    observa o
def reference_date(date):
    years = pd.DatetimeIndex(date).year.values + 3 # Calcula 3 anos a
        frente
    years = years.tolist()
    years = [str(i) for i in years]
    return years

# Conecta com a API das Expectativas de Mercado Anuais
em = Expectativas()
ep = em.get_endpoint('ExpectativasMercadoAnuais')

# Importa as expectativas do IPCA anuais e realiza os filtros
ipca_raw = (
    ep.query()
    .filter(ep.Indicador == "IPCA")
    .filter(ep.Data >= years_sub(10 * 365))
    .filter(ep.baseCalculo == 0)
    .select(ep.Data, ep.Mediana, ep.Minimo, ep.Maximo, ep.DataReferencia)
    .collect()
)

# Realiza o filtro para a data de refer ncia 3 anos a frente das obs.
ipca = ipca_raw[(
    ipca_raw
    .DataReferencia == reference_date(ipca_raw['Data'])
)]

# Renomeia as colunas
ipca = ipca.rename(columns = {'Data' : 'date',
    'Mediana' : 'ipca_e',
    'Minimo' : 'ipca_e_min',
    'Maximo' : 'ipca_e_max'}).drop(['
    DataReferencia'], axis = 1)
```

```

# Importa as expectativas da Selic anuais e realiza os filtros
selic_raw = (
    ep.query()
    .filter(ep.Indicador == "Selic")
    .filter(ep.Data >= years_sub(10 * 365))
    .filter(ep.baseCalculo == 0)
    .select(ep.Data, ep.Mediana, ep.Minimo, ep.Maximo, ep.DataReferencia)
    .collect()
)

# Realiza o filtro para a data de referencia 3 anos a frente das obs.
selic = selic_raw[(
    selic_raw
    .DataReferencia == reference_date(selic_raw['Data'])
)]

# Renomeia as colunas
selic = selic.rename(columns = {'Data' : 'date',
                                'Mediana' : 'selic_e',
                                'Minimo' : 'selic_e_min',
                                'Maximo' : 'selic_e_max'}).drop(['DataReferencia'], axis = 1)

# Junta os dados em um data frame
proxy_neutro = pd.merge(left = ipca,
                        right = selic,
                        how = 'inner',
                        on = 'date')

# Cria a função de Fisher para calcular o juro neutro
def fisher(juros, inflacao):
    juros_neutro = (((1 + (juros / 100)) / (1 + inflacao / 100))) - 1) * 100
    return juros_neutro

# Realiza os cálculos
proxy_neutro = proxy_neutro.assign(neutro = lambda x : fisher(x.selic_e,
                                                                x.ipca_e),
                                neutro_min = lambda x : fisher(x.selic_e_min, x.ipca_e_min),
                                neutro_max = lambda x : fisher(x.selic_e_max, x.ipca_e_max))

# Realiza a mudança para a classe datetime e period da coluna date e insere no índice do data frame
proxy_neutro = proxy_neutro.set_index(pd.to_datetime(proxy_neutro['date']).dt.to_period('D')).drop(['date'], axis = 1)

# Realiza a mudança para período mensal com base na média dos valores
proxy_neutro = proxy_neutro.resample('M').mean()

# !pip install ipeadatapy

```

```

import ipeadatapy as ip

# Taxa referencial - swaps - DI pr -360 dias - m dia do periodo (
    IPEADATA/B3)
swaps = (
    ip.timeseries('BMF12_SWAPDI36012')
    .rename(columns = {'VALUE ((% a.a.))' : 'swaps'})[['swaps']]
)

# Muda a coluna de data para o ndice

swaps = swaps.set_index(swaps.index.to_period('M'))

# Expectativa m dia do IPCA - tx. acumulada para os pr ximos 12 meses
    (Expectativas)
ifl_ep = em.get_endpoint('ExpectativasMercadoInflacao12Meses')

ipca_expec_12m_raw = (
    ifl_ep.query()
    .filter(ifl_ep.Suavizada == 'S', ifl_ep.baseCalculo == 0, ifl_ep.
        Indicador == 'IPCA')
    .collect()
)

# Muda a classe da coluna de data para date time e period, renomeia as
    colunas seleciona a coluna dos valores
ipca_expec_12m = (
    ipca_expec_12m_raw
    .set_index(pd.to_datetime(ipca_expec_12m_raw['Data']))
    .dt.to_period('D'))
    .rename(columns = {'Mediana' : 'ipca_exp_12m'})
    [['ipca_exp_12m']]
)

# Mensaliza os dados com base na m dia dos valores
ipca_expec_12m = ipca_expec_12m.resample('M').mean()

# Junta o df do swap e expectativas do ipca
ex_ante = ipca_expec_12m.join(swaps)

# Calcula o juro real ex-ante
ex_ante = ex_ante.assign(juro_real = lambda x: fisher(x.swaps, x.
    ipca_exp_12m))

# Junta o juro neutro e o juro real; renomeia as colunas
df_juros = (
    proxy_neutro
    .join(ex_ante)
    [['juro_real', 'neutro']]
    .rename(columns = {'juro_real' : 'Juro Real',
        'neutro' : 'Juro Neutro'}))

# Muda a coluna de data para timestamp (para gerar o gr fico)
df_juros.index = df_juros.index.to_timestamp(freq='M')

# Transforma de wide para long
df_juros_long = pd.melt(df_juros.reset_index(),
    id_vars = 'date',
    value_vars = ['Juro Real', 'Juro Neutro'],

```



```

        var_name = 'variable',
        value_name = 'values')

# Configura o tema do gráfico
theme = {'figure.figsize' : (15, 10),
        'axes.facecolor' : 'white',
        "axes.spines.right": False,
        "axes.spines.top": False}

# Cor das variáveis
colors = ['#b22200', '#282f6b']

# Adiciona as configurações
sns.set_theme(style = 'white',
              palette = colors,
              rc = theme)

# Plota
sns.lineplot(x = 'date', y = 'values',
             data = df_juros_long,
             hue = 'variable',
             linewidth = 2).set(xlabel = "",
                               ylabel = "%a.a",
                               title = "Juro Real ex-ante vs. Juro
                                       Neutro")

# Adiciona a fonte no gráfico
plt.annotate(f'Fonte: BCB/IPEA acessado em {datetime.datetime.today().
             strftime("%d/%m/%Y")}',
            xy = (1.0, -0.06),
            xycoords='axes fraction',
            ha='right',
            va="center",
            fontsize=10)

plt.show()

output_path = r"C:\Users\Lenovo\Desktop\Desktop\Mestrado FGV\JurosReais\
             df_juros_long.xlsx"
df_juros_long.to_excel(output_path, index=False)

print(f"Arquivo salvo com sucesso em: {output_path}")

```