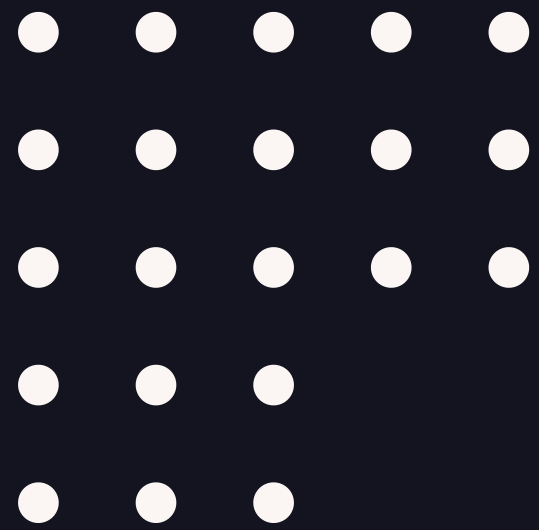




GERENCIAMENTO DE MEMÓRIA

Gabriel Soares, Matheus Rodrigues, Alef Natanael e Gustavo Maia

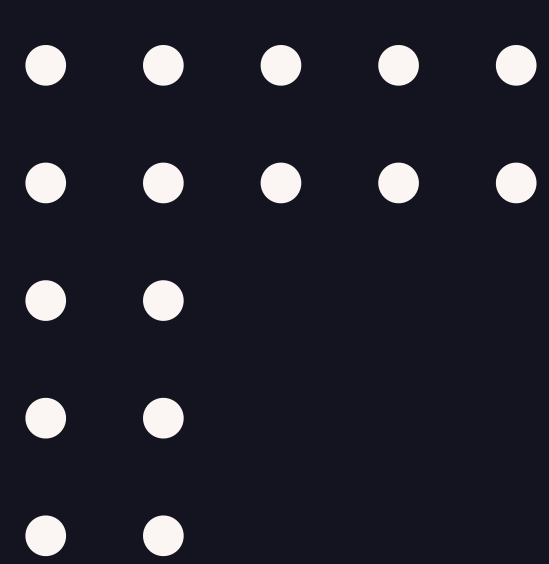




COMPARAÇÃO DE ALGORITMOS DE PAGINAÇÃO

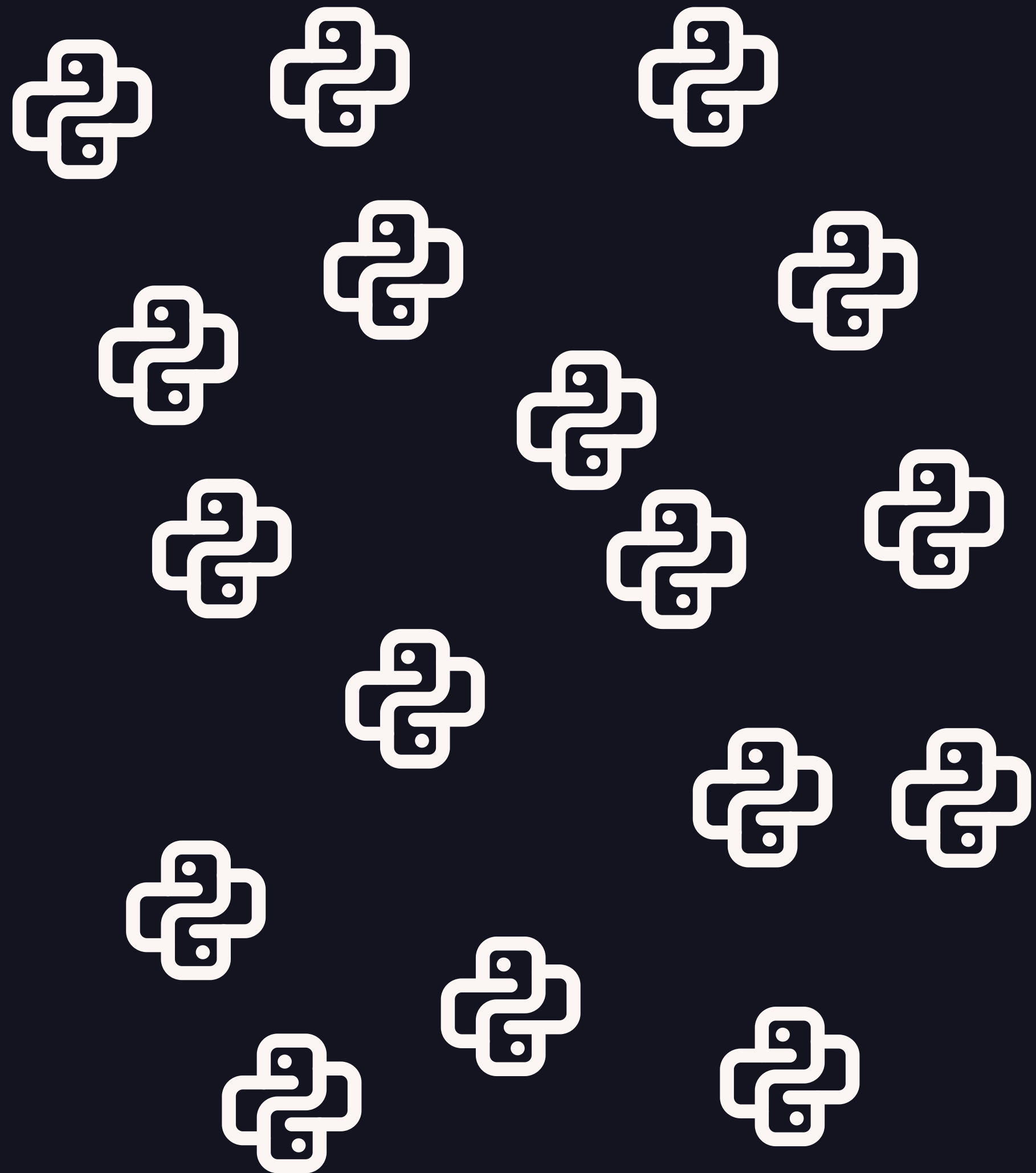
FIFO ~~X~~ ENVELHECIMENTO

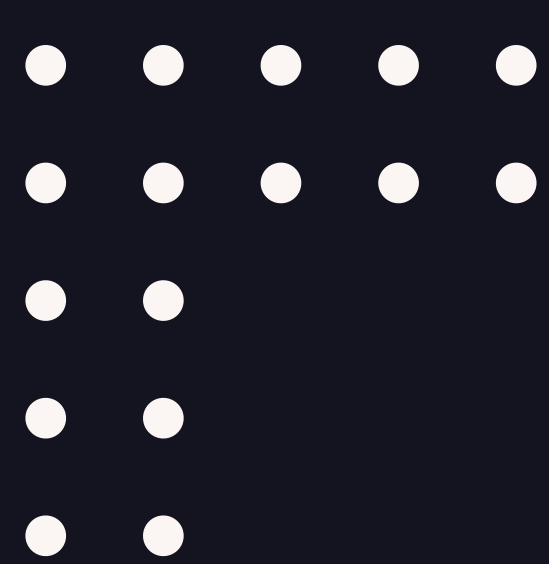




OBJETIVO

- Implementar FIFO e Envelhecimento
- Implementado em Python 3
- Comparar faltas de página por 1000 referências
- Avaliar como parâmetros (nº de molduras, desvio) afetam diferenças





MAS O QUE É O FIFO?

E O QUE É O ALGORÍTMO
DE ENVELHECIMENTO?

```
def gerar_referencias_gaussiana(processos, max_paginas, tamanho_seq_total, fases_por_processo=5):
    sequencias = []
    for _ in range(processos):
        referencias = []
        tamanho_por_fase = tamanho_seq_total // fases_por_processo
        for _ in range(fases_por_processo):
            media = random.randint(0, max_paginas - 1)
            desvio = random.randint(15,25)
            fase = np.random.normal(loc=media, scale=desvio, size=tamanho_por_fase)
            fase = np.clip(np.round(fase), 0, max_paginas - 1).astype(int)
            referencias.extend(fase.tolist())
        sequencias.append(referencias)
    return sequencias
```

```
def fifo(referencias, molduras):  
    memoria = []  
    faltas = 0  
    for pagina in referencias:  
        if pagina not in memoria:  
            faltas += 1  
            if len(memoria) < molduras:  
                memoria.append(pagina)  
            else:  
                memoria.pop(0)  
                memoria.append(pagina)  
    return faltas
```

```
def envelhecimento(referencias, molduras, bits=8):
    memoria = {}
    contador = {}
    faltas = 0

    for pagina in referencias:
        for p in contador:
            contador[p] >>= 1

        if pagina in memoria:
            contador[pagina] |= 1 << (bits - 1)
        else:
            faltas += 1
            if len(memoria) < molduras:
                memoria[pagina] = True
                contador[pagina] = 1 << (bits - 1)
            else:
                menos_usada = min(contador, key=contador.get)
                del memoria[menos_usada]
                del contador[menos_usada]
                memoria[pagina] = True
                contador[pagina] = 1 << (bits - 1)

    return faltas
```

```
def simular():
    processos = 3
    max_paginas = 50
    tamanho_seq = 10000
    molduras_teste = list(range(5, 31, 5))

    sequencias = gerar_referencias_gaussiana(processos, max_paginas, tamanho_seq_total=tamanho_seq)

    with open("referencias_gaussiana.txt", "w") as f:
        for i, seq in enumerate(sequencias):
            f.write(f"Processo {i+1}: {seq}\n")

    todos_resultados = []

    for i, seq in enumerate(sequencias):
        fifo_resultado = []
        envelhecimento_resultado = []

        for m in molduras_teste:
            faltas_fifo = fifo(seq, m)
            faltas_envelhecimento = envelhecimento(seq, m)
            fifo_resultado.append(faltas_fifo / (len(seq) / 1000))
            envelhecimento_resultado.append(faltas_envelhecimento / (len(seq) / 1000))

        todos_resultados.append((fifo_resultado, envelhecimento_resultado))
```

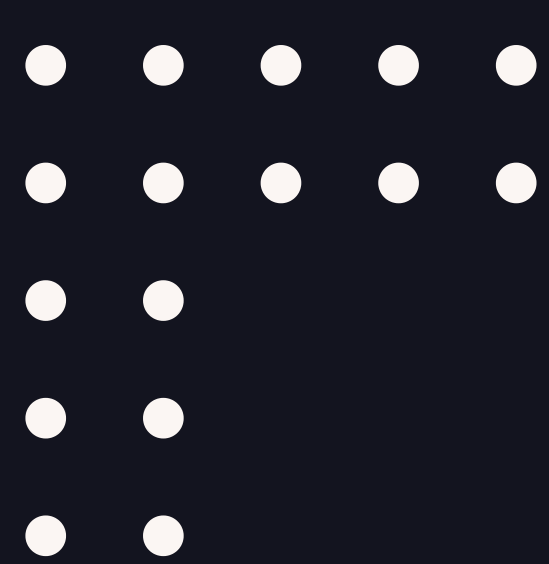


```
plt.figure()
plt.plot(molduras_teste, fifo_resultado, label="FIFO", marker='o')
plt.plot(molduras_teste, envelhecimento_resultado, label="Envelhecimento", linestyle="--", marker='x')
plt.xlabel("Número de Molduras de Página")
plt.ylabel("Faltas por 1000 Referências")
plt.title(f"Processo {i+1}: FIFO x Envelhecimento")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig(f"grafico_processo_{i+1}.png")
plt.show()
```

```
plt.figure()
for i, (fifo_resultado, envelhecimento_resultado) in enumerate(todos_resultados):
    plt.plot(molduras_teste, fifo_resultado, label=f"FIFO - Processo {i+1}", marker='o')
    plt.plot(molduras_teste, envelhecimento_resultado, label=f"Envelhecimento - Processo {i+1}", linestyle="--", marker='x')
```

```
plt.xlabel("Número de Molduras de Página")
plt.ylabel("Faltas por 1000 Referências")
plt.title("Comparação Geral - FIFO x Envelhecimento")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("grafico_geral.png")
plt.show()
```

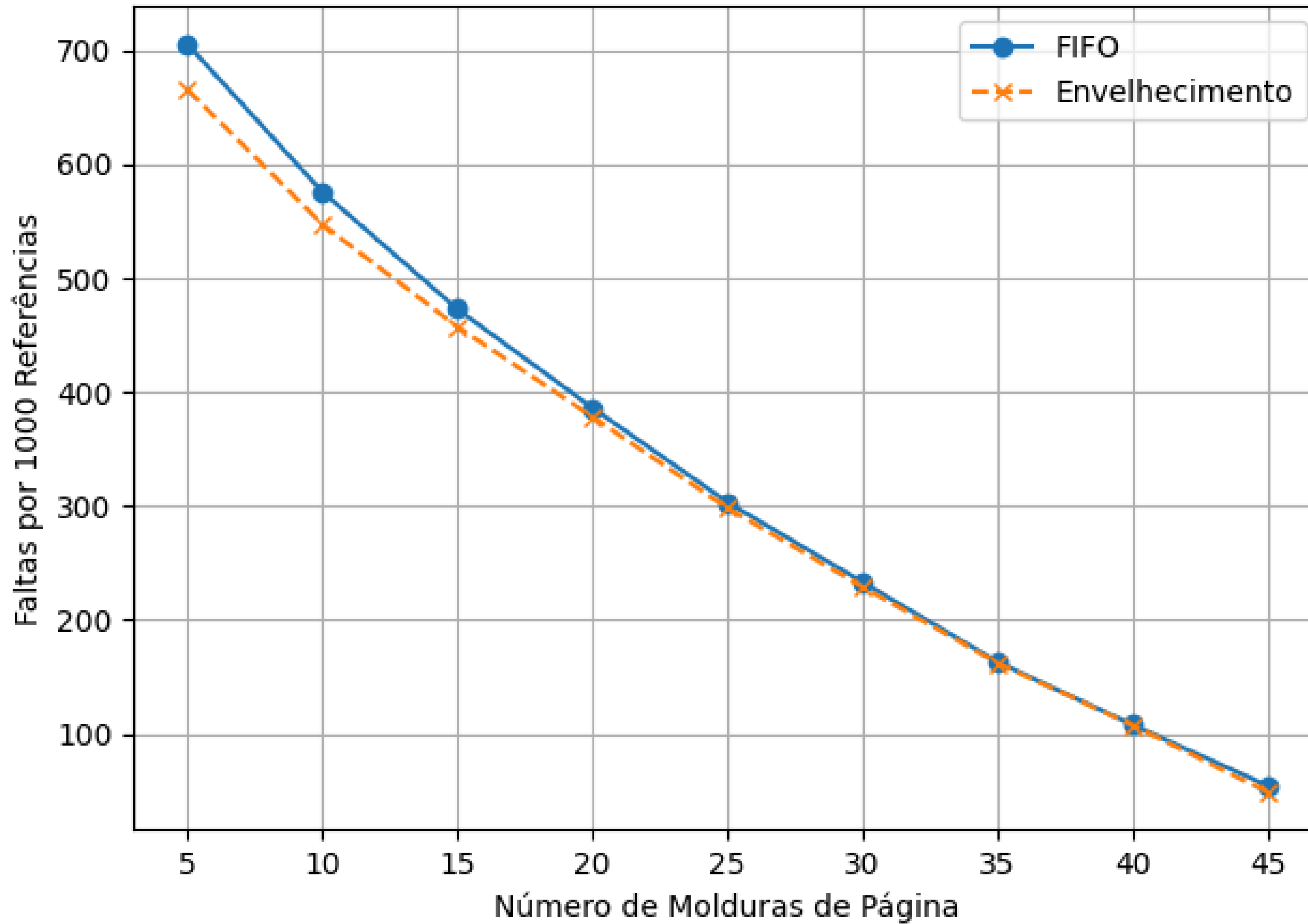
```
if __name__ == "__main__":
    simular()
```



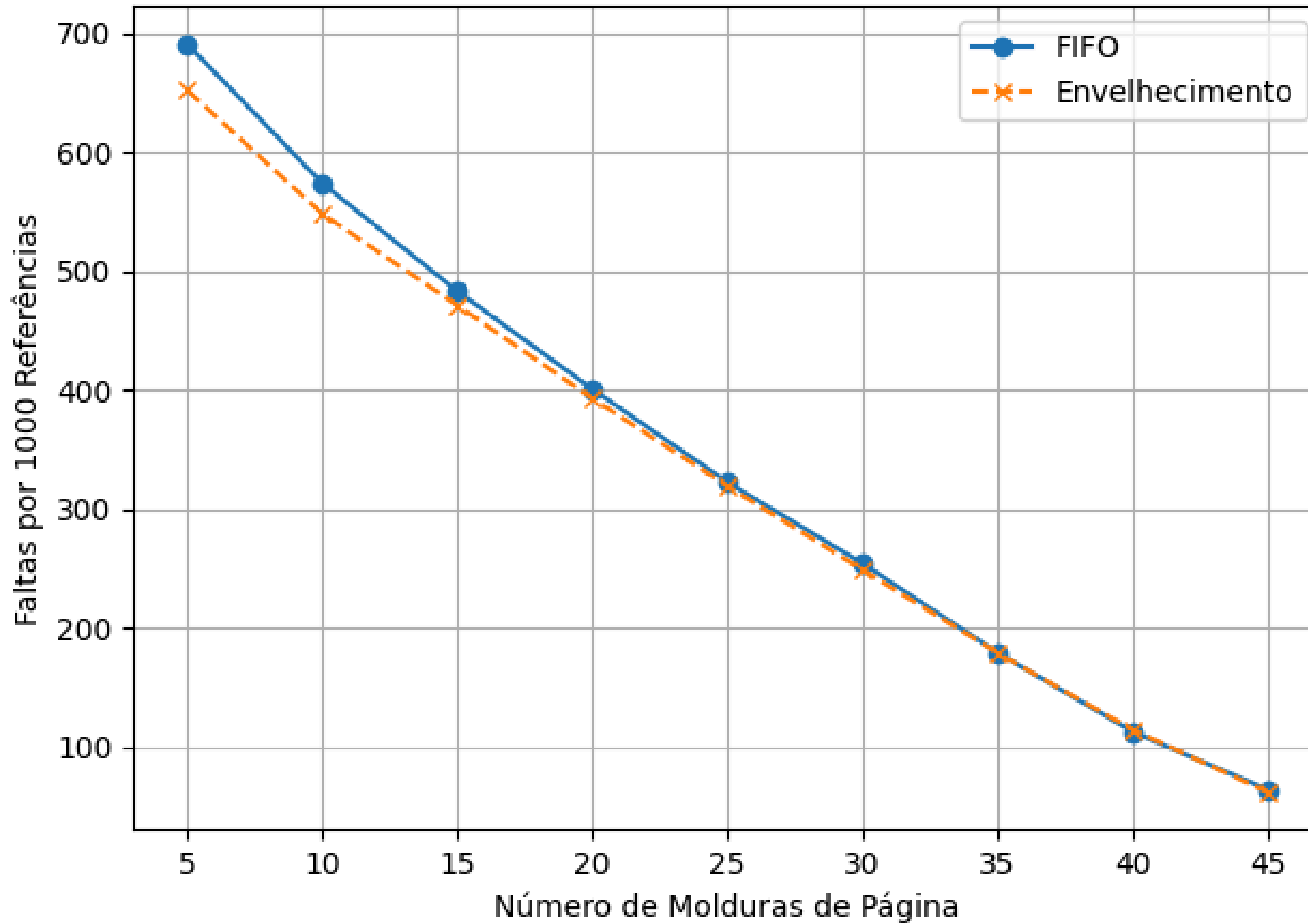
PARÂMETROS DE SIMULAÇÃO

- Processos: 3
- Páginas virtuais: 50
- Sequência total: 10.000 referências
- Molduras testadas: 5, 10, 15, ..., 45

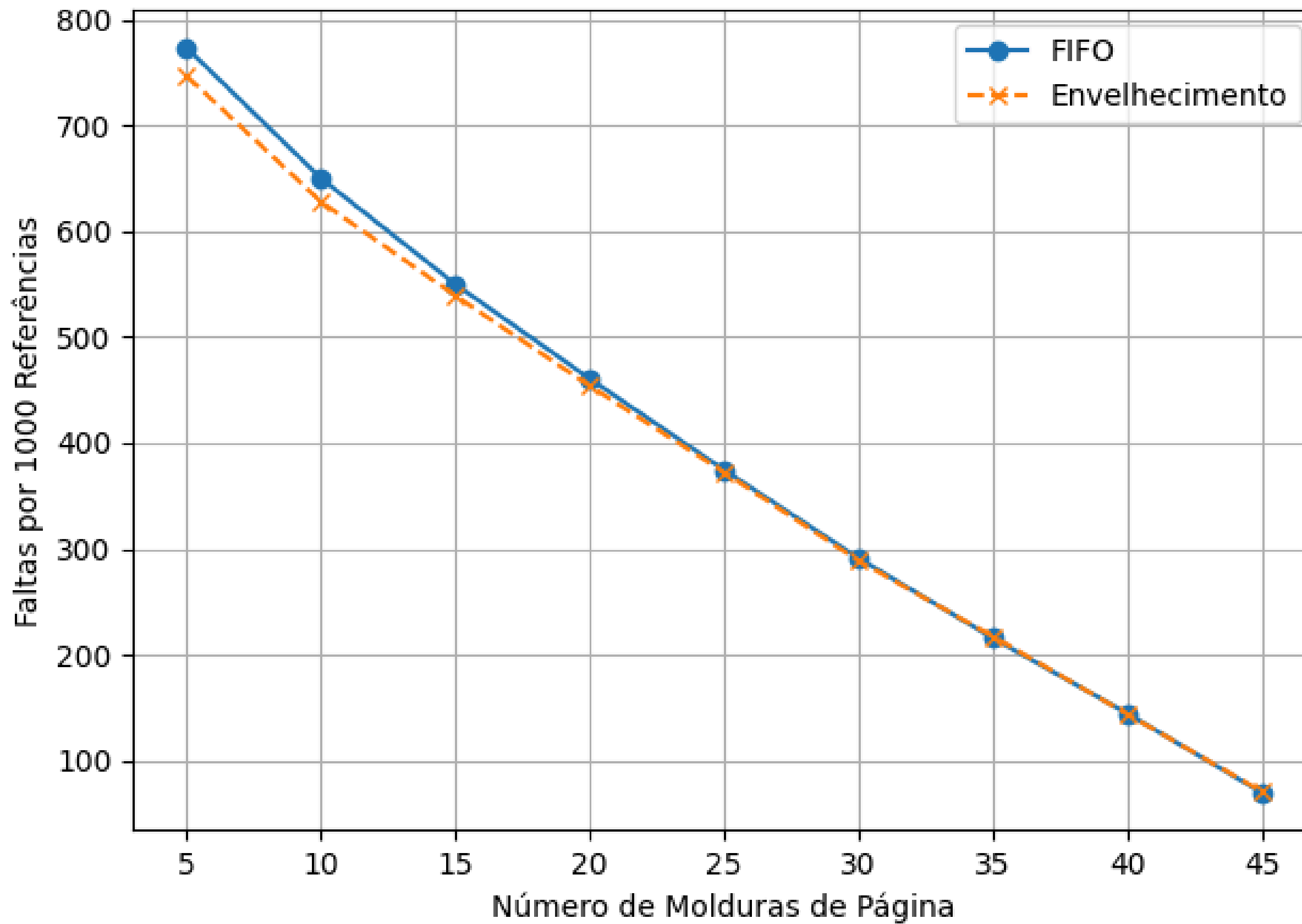
Processo 1: FIFO x Envelhecimento



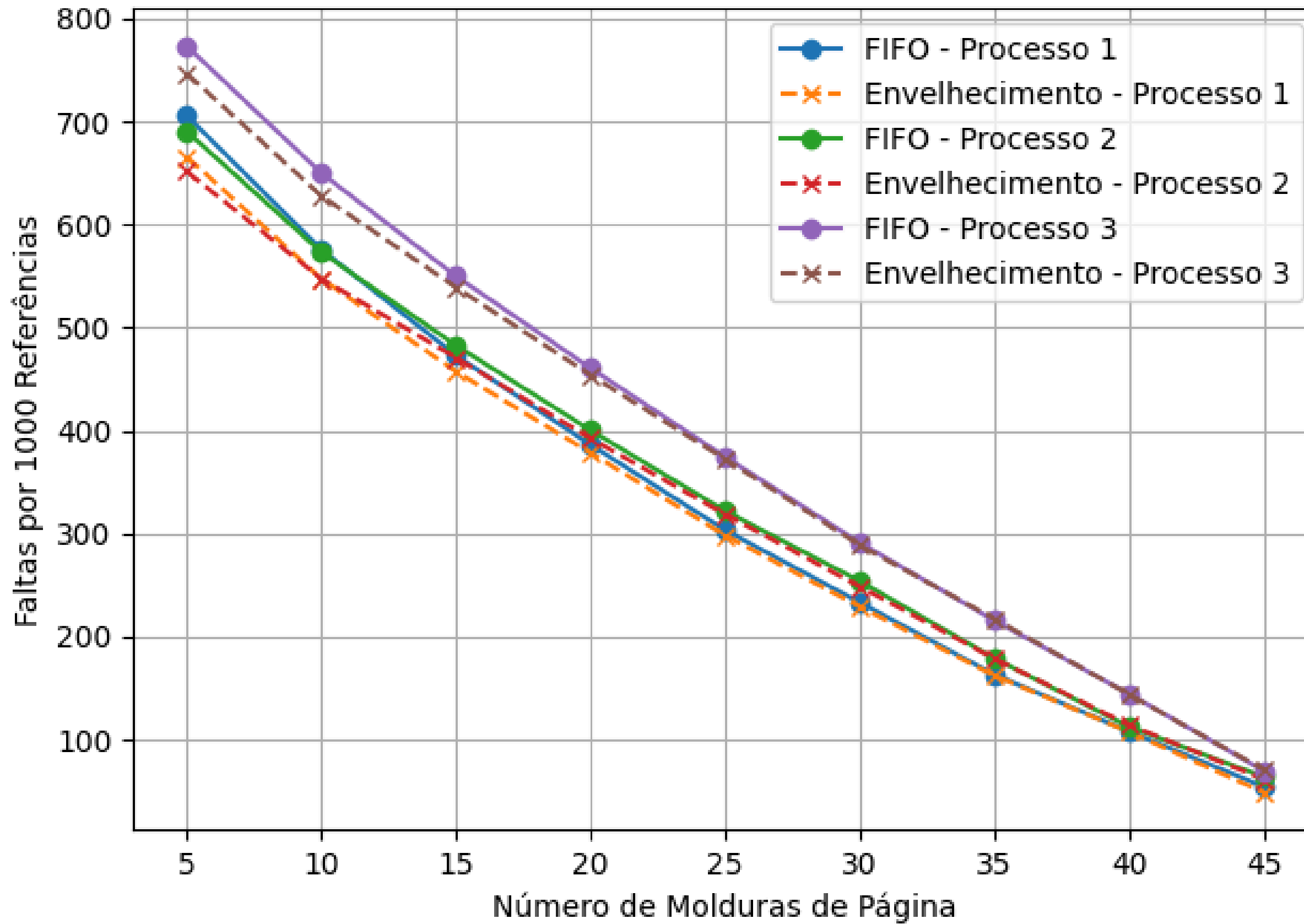
Processo 2: FIFO x Envelhecimento

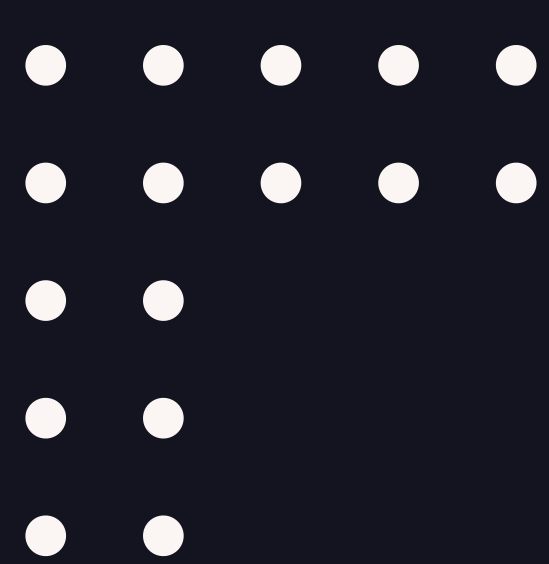


Processo 3: FIFO x Envelhecimento



Comparação Geral - FIFO x Envelhecimento





CONCLUSÃO

- Envelhecimento se sobressai em ambientes com poucas molduras
- Se Molduras \geq working set, as curvas convergem
- Maior $\sigma \rightarrow$ mais dispersão de acesso \rightarrow envelhecimento ganha vantagem