



## RESUMO

Este trabalho apresenta um estudo das operações de produto matriz-vetor e fatoração LU incompleta com nível de preenchimento ( $ILU(p)$ ) no contexto de armazenamento de matrizes esparsas. Foram analisados quatro formatos de armazenamento: COO (*Coordinate Storage*), CSC (*Compressed Sparse Column*), CSR (*Compressed Sparse Row*) e *Lista Encadeada*. O método de *Lista Encadeada* apresentou o pior desempenho, devido ao uso ineficiente da memória cache, enquanto o CSR foi o mais eficiente. Logo, o formato CSR é a melhor escolha para matrizes esparsas de grande porte nas operações avaliadas.

## ARMAZENAMENTOS

As quatro estruturas de dados utilizadas para armazenar matrizes esparsas neste subprojeto foram:

- COO (*Coordinate Storage*)
- CSC (*Compressed Sparse Column*)
- CSR (*Compressed Sparse Row*)
- *Lista Encadeada*

## PRODUTO MATRIZ-VETOR

Dada uma matriz esparsa  $A$  de ordem  $N$  e um vetor  $v$  de ordem  $N$ , a multiplicação matriz-vetor é definida por  $p = Av$ , onde  $p$  é o vetor resultante do produto. Cada elemento  $p_i$  do vetor é calculado por:

$$p_i = \sum_{j=1}^N a_{ij} * v_j \quad (1)$$

## FATORAÇÃO $ILU(p)$

A fatoração  $ILU(p)$  consiste em achar dois fatores  $\tilde{L}$  e  $\tilde{U}$ , tais que  $A \approx \tilde{L}\tilde{U}$ , sendo  $\tilde{L}$  uma matriz triangular inferior com diagonal unitária e  $\tilde{U}$  uma matriz triangular superior. De forma geral os coeficientes da matriz  $L$  e  $U$ , na etapa  $k + 1$  do processo de fatoração podem ser representados por:

$$l_{ik} \leftarrow a_{ik} = \frac{a_{k+1,k}}{a_{kk}} \quad (2)$$

$$u_{ij} \leftarrow a_{ij} = a_{ij} - l_{ik} * a_{kj} \quad (3)$$

O parâmetro  $p$  determina o nível de preenchimento permitido, sendo considerado preenchimento até a etapa  $p + 1$  do processo de fatoração.

## MATRIZES UTILIZADAS

As matrizes do repositório *SuiteSparse Matrix Collection*: Hamrle3, StocF-1465, Geo\_1438, Long\_Coup\_dt6, Flan\_1565, F1, Fault\_639, cage14, CurlCurl\_4, FullChip, thermal2, CurlCurl\_3, dielFilterV2real, dielFilterV3real, nd24k, raefsky3, venkat01, power9, language, majorbasis, xenon2 e PR02R.

## REFERÊNCIAS

- [1] Miguel V. M. Pim. Estruturas de dados otimizadas para armazenar e manipular matrizes esparsas de grande porte. Technical report, UFES, 09 2024.
- [2] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

## RESULTADOS DO PRODUTO MATRIZ-VETOR

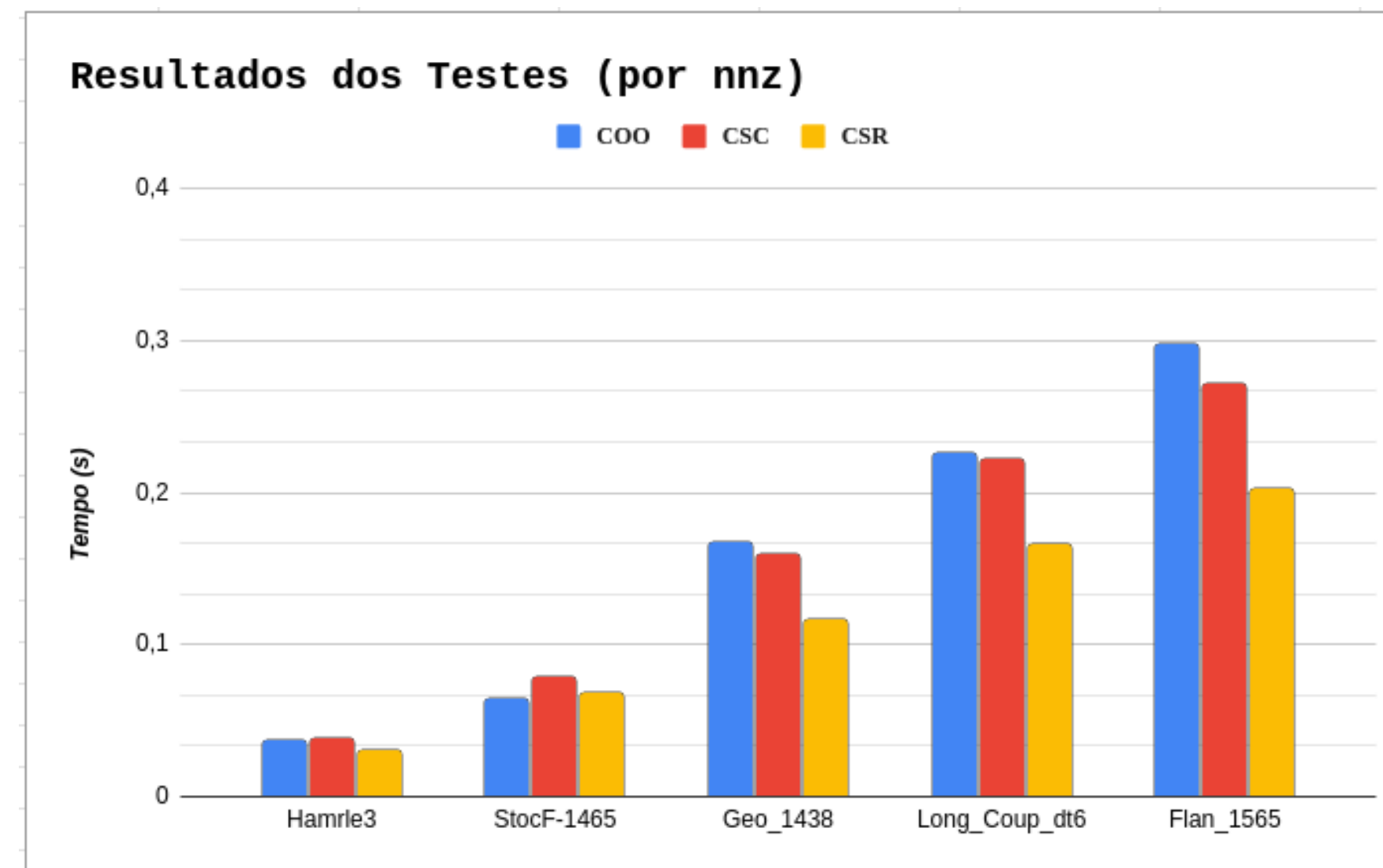


Figura 1: Resultados dos testes por nnz

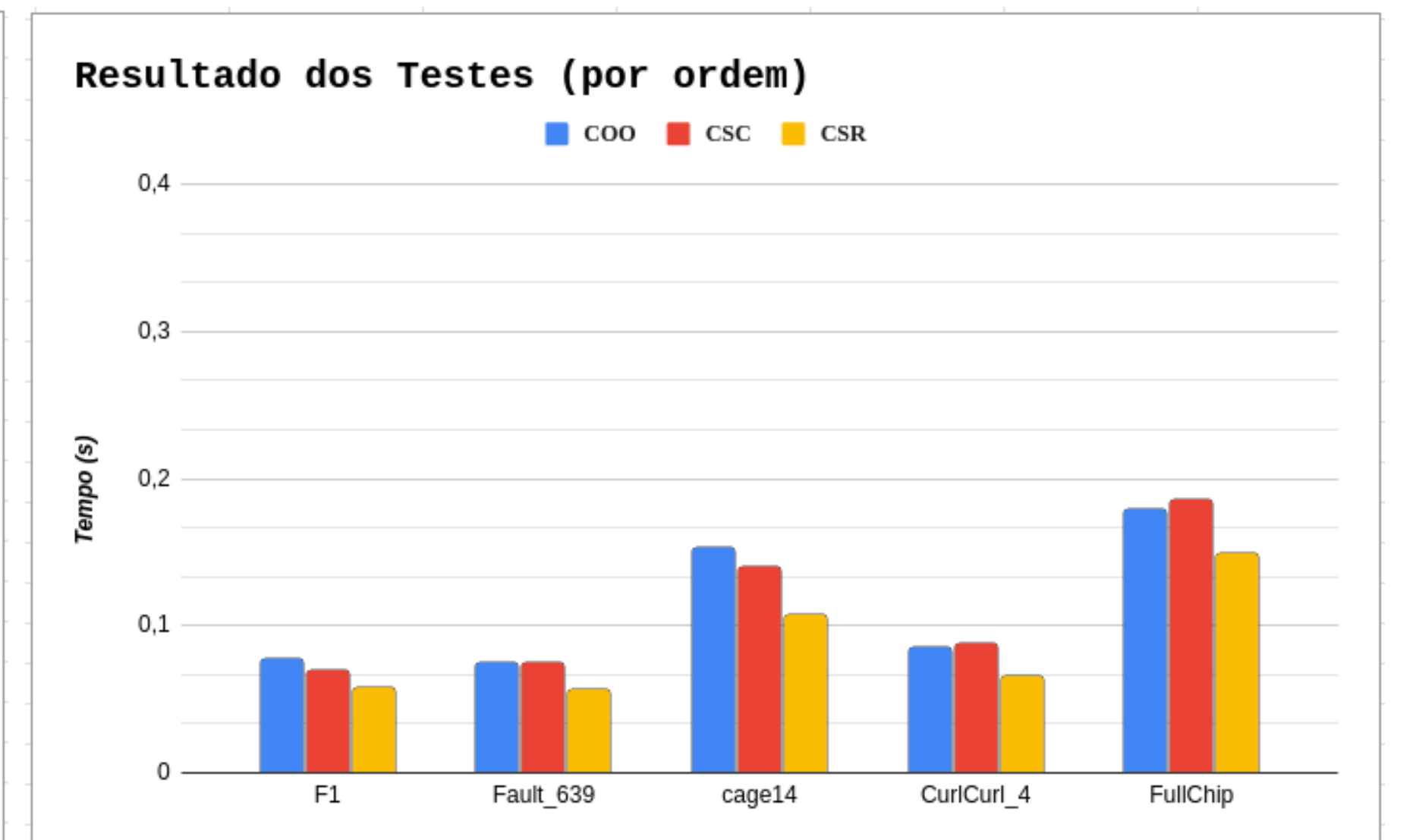


Figura 2: Resultados dos testes por ordem

Matrizes	COO	LinkedList
Hamrle3	0,037	0,262
StocF-1465	0,064	0,632
Geo_1438	0,167	2,856
Long_Coup_dt6	0,226	3,574
Flan_1565	0,297	6,626

Tabela 1: Tempo de Execução do Produto matriz-vetor - Comparação COO com *Lista Encadeada*

## RESULTADOS DA FATORAÇÃO $ILU(p)$

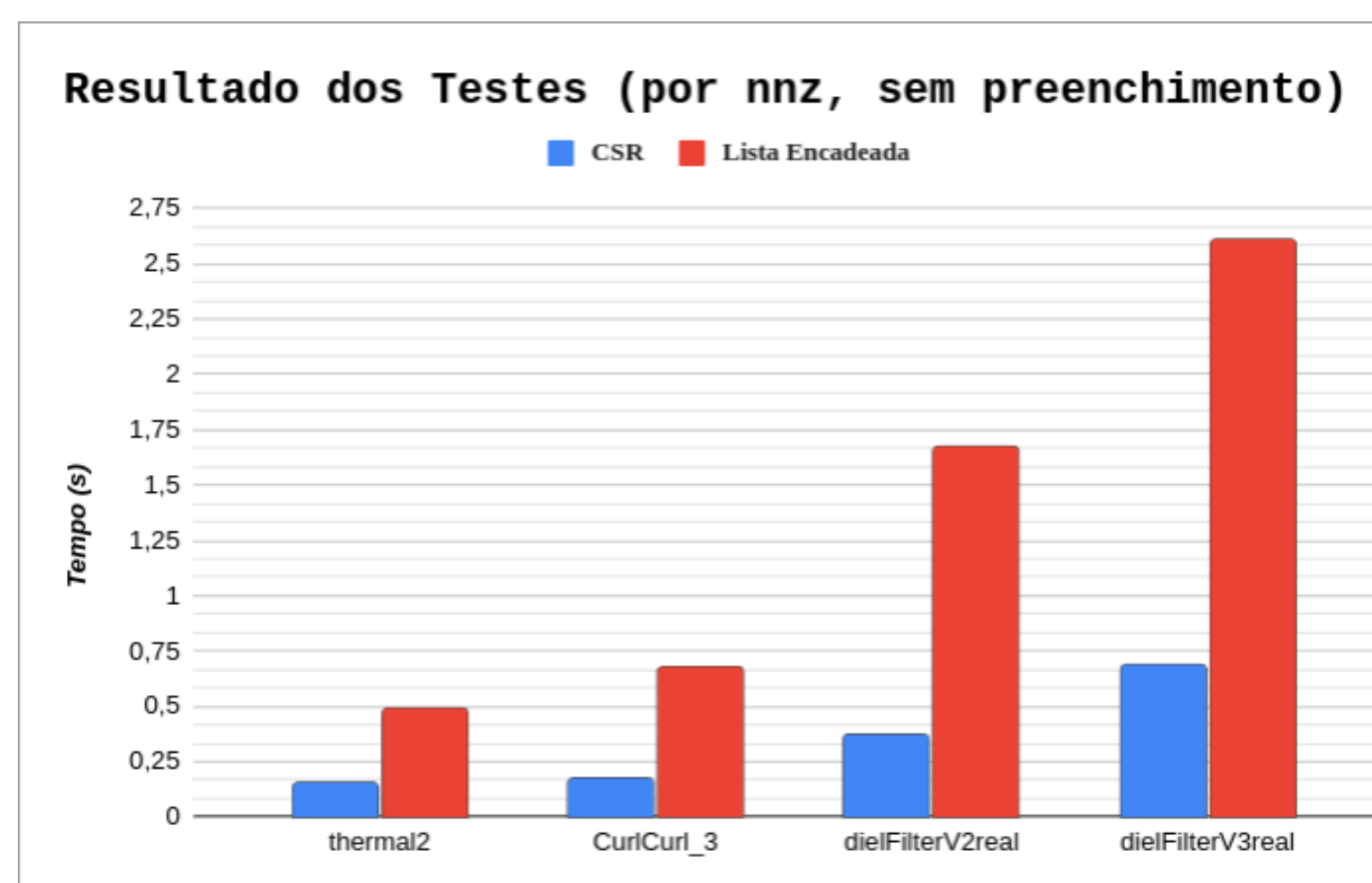


Figura 3: Resultados dos testes por nnz, sem preenchimento

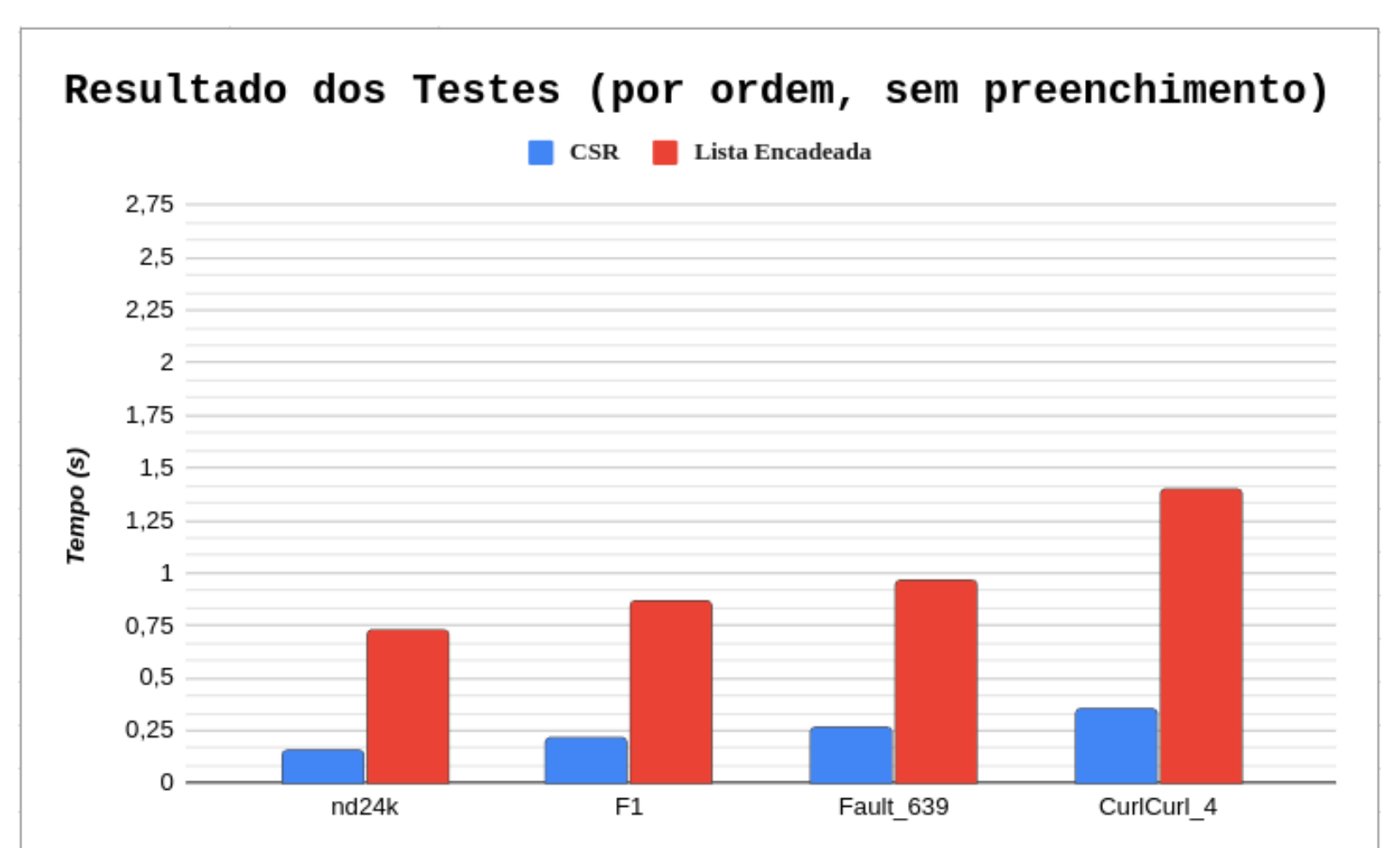


Figura 4: Resultados dos testes por ordem, sem preenchimento

Matrizes	matriz CSR				matriz LinkedList			
	0	1	2	3	0	1	2	3
raefsky3	0,079	0,180	0,298	0,451	0,179	0,483	0,985	1,709
venkat01	0,057	0,108	0,215	0,356	0,147	0,347	1,0299	3,052
power9	0,098	24,895	59,095	122,229	0,317	89,380	124,532	466,382
language	0,139	0,231	0,645	1,636	0,200	0,554	1,904	5,946
majorbasis	0,056	0,165	0,532	1,540	0,142	0,685	2,104	12,422
xenon2	0,168	0,547	1,211	2,657	0,389	1,374	4,355	34,980
PR02R	0,388	1,367	4,316	8,751	1,052	5,257	25,266	865,812

Tabela 2: Resultados dos testes por nnz e ordem, com preenchimento

PR02R e power9 são as duas matrizes com maior quantidade de preenchimento em cada um dos casos de teste. Por isso, o tempo de execução aumenta numa taxa muito maior que as outras matrizes.

## AGRADECIMENTOS

Agradecimentos à professora e doutora Lucia Catabriga por me orientar durante todo o projeto, e em especial ao CNPq pelo apoio financeiro.