

Apêndice: Métodos e Implementações — Framework CMR v1.0

1. Introdução

Este apêndice descreve, de forma detalhada e operacional, os experimentos, códigos e métodos utilizados no contexto do *Framework CMR (Campo de Materialização Relacional)* disponível em:

<https://github.com/BondfinderBR/CMR-v1.0-Operational-Release>

O objetivo é permitir que qualquer pesquisador técnico reproduza, entenda e integre os experimentos em seu próprio ambiente.

As implementações foram desenvolvidas em Python, sem dependência de frameworks pesados, e todos os parâmetros operacionais estão explícitos nos códigos. O foco é a **reprodutibilidade funcional**, não a simulação física de sistemas quânticos.

2. Organização do Repositório

Estrutura de pastas

```
CMR-v1.0-Operational-Release/
├── README.md
├── LICENSE
├── paper/
│   ├── CMR_Framework_pt.pdf
│   ├── CMR_Framework_en.pdf
│   └── appendix_methods.pdf
├── experiments/
│   ├── operational/
│   ├── empirical/
│   └── conceptual/
├── configs/
└── .gitignore
```

- **operational/**: experimentos que testam a lógica operacional do CMR.
 - **empirical/**: testes com ruído físico real (não simulado).
 - **conceptual/**: modelos conceituais que exploram lógica observacional (ex.: dupla fenda operacional).
-

3. Experimentos Operacionais

Todos os scripts abaixo estão em `experiments/operational/` e testam diferentes aspectos do CMR conforme discutido no paper.

3.1 `bell_cmr.py`

- **Objetivo:** testar correlações entre pares de observadores variando ângulos, inspirado em desigualdades de Bell.
- **Critério:** calcular correlação empírica e estimar valor CHSH S .
- **Saída típica:** valores de $E(a,b)$ e S .

3.2 `bell_cmr_meio_fraco.py`

- **Objetivo:** testar o efeito de “meio fraco”, onde a coerência do meio é reduzida.
- **Critério:** observar como o valor de S muda com intensidade de meio fraco.

3.3 `bell_cmr_mult_obs.py`

- **Objetivo:** estender o teste de Bell com múltiplos observadores no mesmo meio.
- **Critério:** medir divergência entre observadores que experimentam o mesmo evento sob condição de memória compartilhada.

3.4 `asymmetric_reality_cmr.py`

- **Objetivo:** explorar divergência de realidade percebida quando observadores têm estados de memória diferentes.
- **Critério:** taxa de divergência observacional.

3.5 `cmr_realidade_objetiva.py`

- **Objetivo:** testar a hipótese da “falácia da objetividade global”.
- **Critério:** taxa de divergência em regime misto.

3.6 `cmr_observador_memoria.py`

- **Objetivo:** testar a influência da memória temporal na estabilidade observacional.
- **Critério:** mudanças percebidas com e sem memória.

3.7 `cmr_memorias_incompatibilizs.py`

- **Objetivo:** testar a compatibilidade/incompatibilidade de histórias observacionais.
- **Critério:** discordância entre observadores com memórias incompatíveis.

3.8 `cmr_saturacao_consenso.py`

- **Objetivo:** testar saturação de consenso em múltiplos observadores.
- **Critério:** divergência média em função do número de observadores.

3.9 `cmr_convergencia_forcada.py`

- **Objetivo:** testar a convergência observacional forçada por critérios externos.
- **Critério:** taxa de convergência.

3.10 `cmr_autoridade_narrativa.py`

- **Objetivo:** explorar regimes narrativos onde um observador influencia outro.
- **Critério:** taxa de alinhamento e mudanças observadas.

3.11 `cmr_trauma.py`

- **Objetivo:** investigar mudanças observacionais após eventos extremos.
- **Critério:** instabilidades percebidas.

4. Teste Empírico com Ruído Físico

`experiments/empirical/cmr_contradicao_8B_jitter.py`

- **Objetivo:** atacar diretamente objeções referentes a artefatos de simulação, usando ruído físico real (RTT jitter).
- **Procedimento:**
 - coletar RTT entre nós
 - medir instabilidade percebida com e sem memória
- **Critério:**
 - o meio não se torna mais estável com memória
 - a redução de instabilidade ocorre no observador

5. Modelos Conceituais

Estes scripts ilustram a lógica operacional da materialização de fatos, sem pretensão de simular física real:

`materializacao_cmr.py`

- **Objetivo:** testar operação de materialização de assinatura de estados entre dois nós.
- **Critério:** comparação de assinaturas A e B por média, desvio e entropia.

`campo.py`, `run.py`

- **Objetivo:** modelo conceitual de comunicação dependente de meio.
- **Critério:** inferência de estado latente a partir de padrões observados.

Scripts de Dupla Fenda Conceitual

- `emissor.py`

- `receptor.py`
- `observer_nomem.py`
- `observer_log.py`

Estes scripts exploram a lógica de observação dependente de método, usando uma metáfora da dupla fenda.

Importante: Estes não simulam interferência física. Eles testam **lógica de inferência observacional**, e isso está declarado no README.

6. Reprodutibilidade e Parâmetros

Configurações

Parâmetros de experimento encontram-se organizados em:

`configs/`

Se houver arquivos `.yaml` ou similares, eles podem ser carregados pelos scripts para padronizar regimes de meio / memória.

7. Executando os Experimentos

Pré-requisitos

- Python 3.8+
- Pacotes mínimos: `numpy`, `scipy`, `matplotlib` (quando aplicável)

Instalação (exemplo):

```
pip install numpy scipy matplotlib
```

8. Observações Importantes

Sobre a Dupla Fenda Conceitual

Este modelo **não é uma simulação física do experimento de dupla fenda real**. Ele captura a **lógica de materialização do dado observacional**, ou seja:

- o papel do método na definição do que é observado
- como inferências são feitas a partir de padrões
- como um observador interpreta estados latentes

Essa distinção é declarada no apêndice e no README.

Sobre o uso de memória

“Memória” significa **integração temporal estatística**.

Não implica consciência, psicologia, nem mecânica de gpu/estado quântico.

9. Saídas de Logs

Todos os scripts imprimem no console estatísticas-chave, tais como:

- taxa de divergência
 - correlações
 - medidas de instabilidade
 - médias e desvios padrão
 - inferências de estado
-

10. Limitações

- Não são modelos físicos de alta fidelidade.
 - Experimentos conceituais não garantem comportamento microscópico real.
 - A conexão entre modelos e realidade física está nos **regimes operacionais**, não nas interações fundamentais.
-

11. Citação

Ao utilizar estes métodos em publicações:

Flávio Oliveira, CMR Framework Operational Release, GitHub, 2026.