1. LLM-as-a-Judge — Detalhamento

O serviço `tim-ai-agnt-llm-judge` é um batch job que executa como **CronJob Kubernetes**. Não expõe HTTP nem gRPC.

1.1. Fluxo de execução

- Lê config dos agentes em configs/judge/agents.yaml (agentId, daysBack, percentage).
- Define a janela temporal BRT: D − daysBack (default D-1), das 00:00 às 23:59 (America/Sao_Paulo).
- Busca sessions no Langfuse — credenciais por agente via env LANGFUSE_{AGENT}_*.
- Amostragem aleatória determinística — seed = agentId:reference_date (reproduzível). Pega percentage das sessions.
- Coleta todos os traces das sessions amostradas.
- Avalia com LLM — uma chamada por trace e uma por session.
- Calcula métrica determinística local: loop (streak máximo de metadata.intent repetido consecutivo entre traces).
- Gera arquivo .TXT.GZ — 12 colunas, separador |;, todos os campos entre aspas duplas. Linha final: "TOTAL"|;"<n>".
- Upload do arquivo para o Google Cloud Storage no caminho do agente.
- Auditoria via trace Langfuse do próprio job (judge-job-{agentId}, status created/completed/failed).
- Arquivo local apagado após upload. Falha de upload → agente marcado como falho.

Agentes mapeados: `billing`, `sales`, `collection` e `backoffice-anatel`.

1.2. Buckets por ambiente

Ambiente	Bucket
dev	tim-bigdata-dev-ca1f-raw-standard-agnt-ai-atendimento
fqa	tim-bigdata-fqa-60ce-raw-standard-agnt-ai-atendimento
prd	tim-bigdata-prod-e305-raw-standard-agnt-ai-atendimento

1.3. Prefixos por agente

agentId	Prefixo no bucket
billing	agnt_ai_contas
sales	agnt_ai_ofertas
collection	agnt_ai_cobranca
backoffice-anatel	agnt_ai_bko

Padrão do nome do arquivo:
`AGENTE_{agentId}_LLM_JUDGE_{YYYYMMDD}_{HHMMSS}.TXT.GZ`

1.4. Métricas geradas

1.4.1. Trace-level (LLM-judged)

Uma chamada por trace, prompt `trace_metrics.yaml`. Contexto = últimos N turnos (context_turn_limit). Falha de parse → 2ª tentativa com retry-hint; persistindo, métrica vira `None` e é serializada como `"0.00"`.

Métrica	Range	Sentido
judgeScore	0.0–1.0	Qualidade global (clareza, utilidade, tom). Alto = bom.
accuracyScore	0.0–1.0	Factualidade em relação ao contexto. Alto = bom.
alucinationScore	0.0–1.0	Afirmações não suportadas pelo contexto. Alto = ruim (inverso).

1.4.2. Trace-level (determinística)

Métrica	Tipo	Origem
promptLength	int	len(trace.input) — contagem de caracteres bruta.

1.4.3. Session-level (LLM-judged, replicada em cada trace da session)

Uma chamada por session, prompt `session_metrics.yaml`. O valor da session é replicado em cada linha de trace daquela session.

Métrica	Range	Sentido
inferredCsiScore	0.0–1.0	Sentimento inferido (0 negativo, 0.5 neutro, 1 positivo).
resolution	{0, 1}	Problema foi resolvido?
conversationPrecision	{0, 1}	Agente manteve foco? 0 se divagou, repetiu ou entrou em loop.

1.4.4. Session-level (determinística, replicada)

Métrica	Tipo	Descrição
loop	int ≥ 0	Indica se o agente entrou em loop naquela sessão.

1.4.5. Identificadores (não-métrica)

Compõem as quatro últimas colunas do arquivo: `uraCallId`, `channelId`, `sessionId` e `messageId`. Extraídos do metadata (trace primeiro, session como fallback).

1.5. Layout do arquivo de saída

Arquivo `.TXT.GZ` com 12 colunas, separador `|;`, todos os campos entre aspas duplas. Uma linha por trace; métricas de session replicadas em todas as linhas da mesma session.

Cabeçalho:

```
"judgeScore"|;"accuracyScore"|;"alucinationScore"|;"promptLength"|;"loop"|;"inferredCsiScore"|;"resolution"|;"conversationPrecision"|;"uraCallId"|;"channelId"|;"sessionId"|;"messageId"
```

Linha final: `"TOTAL"|;"<n_traces>"`.