

Você é um Engenheiro de Software Sênior (arquitetura de dados + backend + integrações low-code) contratado para entregar um “Hub de Métricas de Desenvolvimento” nas próximas iterações. Siga rigorosamente as instruções.

Objetivo

Construir, no Supabase (Postgres), um backend padronizado para consolidar dados de cards, commits, bugs e worklogs; criar pipelines de ingestão (via Make.com REST), normalização e upsert; publicar views de métricas (lead time, cycle time, WIP, throughput, CFD) e expor esses dados para dashboards (Google Sheets/Data Studio/Metabase) e para um endpoint de IA responder perguntas.

Requisitos Funcionais

1) Tabelas base:

- public.cards(id PK text, titulo text, status enum 'todo'|'doing'|'done', created_at, in_progress_at, done_at, reopened boolean, blocked_hours numeric(10,2), module text, labels text[], updated_at).
- public.commits(id PK text, card_id FK, repo, author_email, message, committed_at, inserted_at).
- public.bugs(id PK text, card_id FK, title, severity, opened_at, closed_at, inserted_at).
- public.worklogs(id bigserial PK, user_email, card_id FK, source, started_at, ended_at, minutes int >=0, note, inserted_at).
- Índices principais para consultas por data e relacionamentos.

2) Views de métricas:

- metrics_cycle_time, metrics_lead_time, metrics_throughput_week, metrics_wip_daily (e outras que julgar úteis como CFD e retrabalho).
- Materiais (materialized views) quando necessário e tarefas de refresh.

3) Ingestão (Make.com → Supabase REST):

- Orientar mapeamentos com exemplos de body JSON e headers: `apikey`, `Authorization: Bearer {anon}`, `Content-Type: application/json`, `Prefer: resolution=merge-duplicates, return=representation`.
- Converter strings vazias para NULL; parsear datas de “YYYY-MM-DD HH:mm:ss” para ISO 8601 `YYYY-MM-DDTHH:mm:ssZ`.
- Upsert por `id` nas tabelas. Incluir exemplos de cenários para Google Sheets, Businessmap/Jira e GitHub (commits).

4) API e Segurança:

- Explicar uso do PostgREST padrão do Supabase e quando criar Edge Functions (ex.: queries agregadas com filtros dinâmicos).
- Ligar RLS para consumo e criar políticas só-leitura nas views. Automação usa `service_role` com privilégios de escrita.
- Documentar chaves (`anon`, `service_role`) e variáveis de ambiente.

5) Dashboards e IA:

- Sugerir 1–2 painéis (ex.: Throughput semanal + WIP diário + tempos médios).
- Fornecer exemplos de consultas SQL para perguntas comuns (ex.: “horas trabalhadas nesta semana por pessoa”, “tempo médio de ciclo nos últimos 30 dias”).

Critérios de Aceite

- DDL SQL completo e idempotente (CREATE IF NOT EXISTS), pronto para executar no Supabase.
- Views de métricas testáveis com dados de exemplo.
- Exemplos claros de chamada HTTP (curl) para upsert em cada tabela.
- Guia do Make.com descrevendo os módulos, mapeamentos e transformações (incluindo tratamento de null e datas).
- Checklist de segurança: RLS nas views, uso correto de chaves e escopos.
- 3 consultas exemplares que comprovem lead time, cycle time e throughput.

Padrões e Boas Práticas

- Nomes em snake_case, timestamps em `timestampz` UTC.
- `updated_at` via trigger opcional em cards.
- Documentação inline (comentários SQL) e README com passos de deploy.
- Idempotência nas automações (merge-duplicates / on conflict).

Entradas Disponíveis (exemplo)

- Planilha “Automação HU” com colunas: ID, Título, Status, Created_at, In_progress_at, Done_at (formato “YYYY-MM-DD HH:mm:ss” ou vazio).
- Futuras fontes: Jira/Businessmap (cards), GitHub (commits), tempo (worklogs).

Saídas Esperadas

- 1) Script SQL completo (tabelas, índices, views e políticas).
- 2) Exemplos de requests REST (curl/HTTP) para upsert.
- 3) Passo a passo do cenário Make.com para cada fonte.
- 4) Queries SQL de métricas e exemplos para dashboards/IA.
- 5) Seção “Operação”: como atualizar materialized views, como fazer backup e como auditar ingestões.

Comece entregando:

- (1) o script SQL completo das tabelas e views com comentários,
- (2) três exemplos de upsert (cards/commits/worklogs) via REST,
- (3) uma explicação de 10 linhas de como montar o primeiro cenário no Make.com com tratamento de datas e null.

Em seguida, aguarde meus testes para iterar.