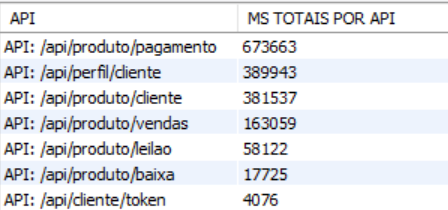
**-- 4. Gerar uma visão(dashboard/Indicador) das top10 APIS que tiveram maior tempo de resposta.**

SELECT CONCAT('API: ', API) AS 'API',

COUNT(TEMPO\_RESPOSTA) AS 'MS TOTAIS POR API'

FROM API GROUP BY API ORDER BY 2 DESC;



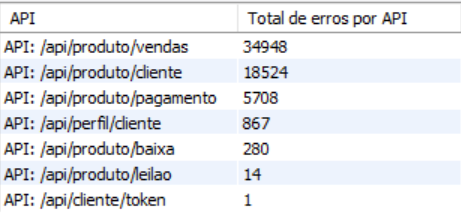
**DASHBOARD CRIADO NO GRAFANA**



**-- 5. Gerar uma visão(dashboard/Indicador) das top10 APIS que tiveram mais erros.**

SELECT CONCAT('API: ', API) AS 'API',

COUNT(API) AS 'Total de erros por API' FROM ERROS GROUP BY API ORDER BY 2 DESC;



**DASHBOARD CRIADO NO GRAFANA**



**--6. Gerar uma visão(dashboard/Indicador) por hora de uso das APIs.**

Para o item eu tive um problema com o MySQL pois a soma das horas é maior que **838:59:59** que é o máximo que um campo do tipo TIMESTAMP aceita.

Eu criei uma tabela de apoio só para demonstrar a lógica que eu usei mas com esse volume de dados não foi possível.

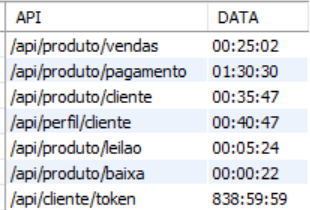
Criei uma tabela chamada TIME\_API com apenas 2000 registros e executei a query

SELECT API, TIME\_FORMAT(SEC\_TO\_TIME(SUM(time\_to\_sec(DATA\_OPERACAO))), '%H:%i:%s') AS 'DATA'

FROM TIME\_API GROUP BY API;

Onde converto as horas e minutos para SEGUNDOS e com base nisso faço a soma.

Depois retorno para o formato de horas, minutos e segundos.





Olhando o dashboard gerado com esses 2000 registros aleatórios da pra ver que uma api /api/client/token já ultrapassou as 838 horas por isso para de fazer a soma.

**--7. Qual API trafegou mais dados no período do log enviado e como você chegou a essa conclusão.**

SELECT concat('API: ', API) AS 'API', count(BYTES) as 'BYTES TRAFEGADOS'

FROM API GROUP BY API ORDER BY 2 DESC;



**DASHBOARD CRIADO NO GRAFANA**

****