



Monitoreo EN TIEMPO REAL



Prometheus



- Sistema de monitoreo y base de datos de series temporales.
- Recolecta métricas vía pull de exporters.
- Lenguaje de consultas: PromQL (lenguaje propio).



Grafana

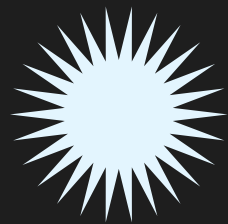


- Plataforma de visualización de métricas.
- Compatible con múltiples fuentes (Prometheus, Loki, etc.).
- Paneles personalizables y alertas.

Utilidades de grafana y prometheus



Grafana visualiza estas métricas en dashboards interactivos, permitiendo a los administradores detectar cuellos de botella, sobrecarga de recursos o fallas en tiempo real.



Grafana puede mostrar series históricas y resaltar anomalías. Al dispararse una alerta, se puede enviar un mensaje a Slack, correo, PagerDuty, etc.



Permite hacer consultas sobre métricas usando su lenguaje PromQL que es muy potente para análisis y detección de patrones

¿Cómo implementar Grafana y Prometheus?

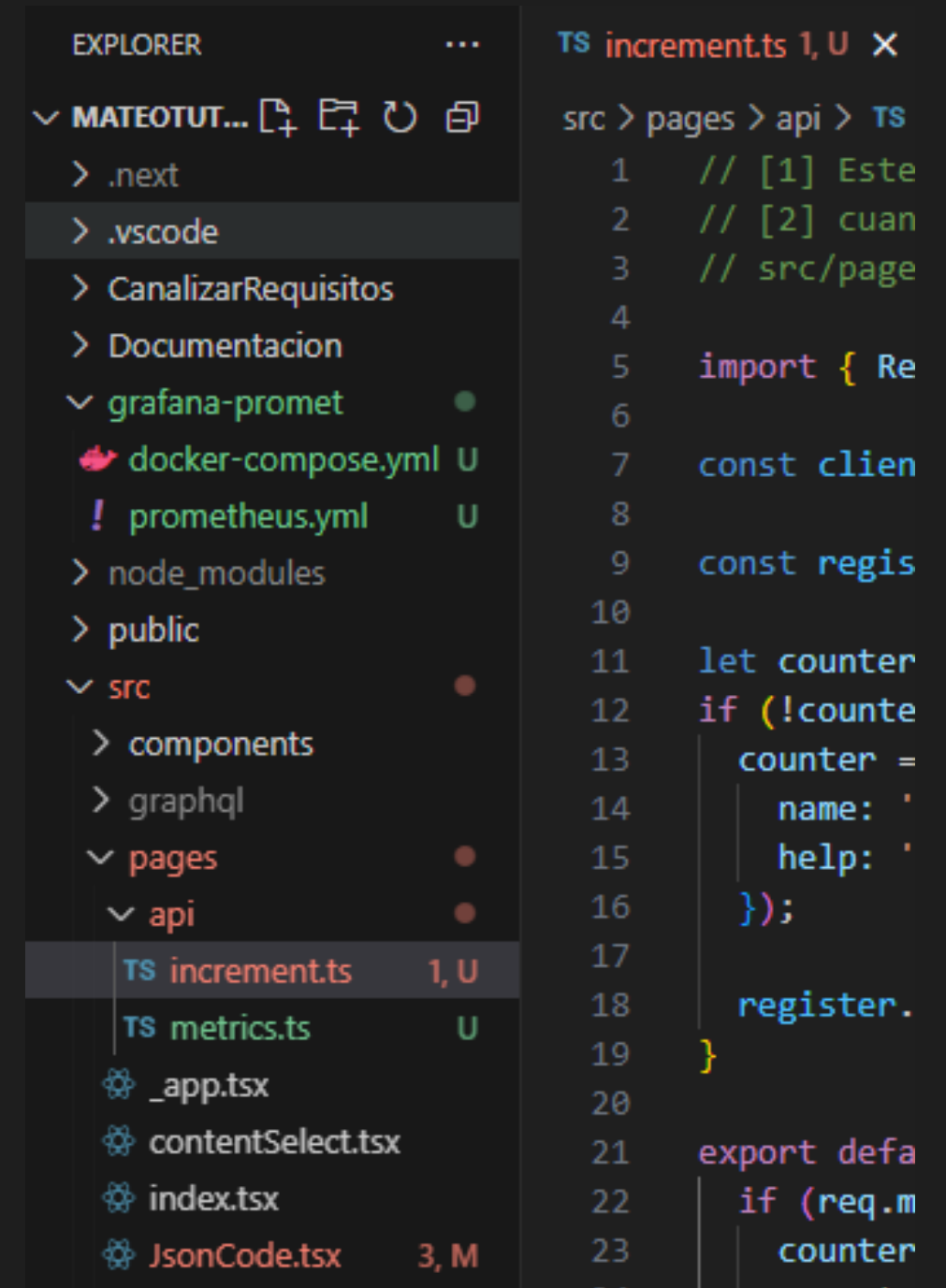
[1] Instalar las librerías necesarias para exponer las métricas en formato prometheus. Se ejecuta en consola “pnpm add prom-client”.

[2] Crear el archivo “metrics.ts” el cual sirve de endpoint para nuestra APP, y crear otro archivo “increment.ts” el cual hace uso del archivo “metrics.ts” para registrar determinadas acciones en nuestra APP.

[3] Implementar en nuestra pagina la comunicación con el archivo “increment.ts”, el cual a su vez se comunica con “metrics.ts”.

[4] Crear el archivo “docker-compose.yml” el cual sirve para levantar los servicios de Grafana y Prometheus usando Docker Compose.

[5] Crear el archivo “prometheus.yml” el cual sirve como configuración específica para "Prometheus".



```
EXPLORER
MATEOTUT...
  .next
  .vscode
  CanalizarRequisitos
  Documentacion
  grafana-promet
  docker-compose.yml
  ! prometheus.yml
  node_modules
  public
  src
    components
    graphql
    pages
      api
        TS increment.ts
        TS metrics.ts
        _app.tsx
        contentSelect.tsx
        index.tsx
        JsonCode.tsx

TS increment.ts 1, U X
src > pages > api > TS
1 // [1] Este
2 // [2] cuan
3 // src/page
4
5 import { Re
6
7 const clien
8
9 const regis
10
11 let counter
12 if (!counte
13   counter =
14     name: '
15     help: '
16   });
17
18 register.
19 }
20
21 export defa
22   if (req.m
23     counter
```

Demostración de nuestra implementación

**MUCHAS
GRACIAS**