

**CURSO: DEVOPS SENIOR**

**Módulo 4: Observabilidad avanzada**

**Ejercicio Práctico 1**

## **Observabilidad Full Stack con Prometheus, Grafana y Exporters en un Entorno Kubernetes Simulado**

### **Objetivo:**

Instalar y configurar Prometheus y Grafana para observar el rendimiento de un clúster Kubernetes, capturar métricas clave, y visualizar dashboards personalizados con alertas básicas.

### **Resultado esperado:**

Comprender la instrumentación del clúster, el consumo de métricas y la creación de tableros útiles para tomar decisiones operativas.

### **Instrucciones:**

- **Entorno sugerido:**
  - Use Play with Docker o un entorno local con Docker Compose.
  - Clone este repositorio base gratuito:  
<https://github.com/open-telemetry/opentelemetry-demo>  
(o usar una app propia dividida en frontend, backend y DB).
- **Actividad guiada:**
  - Lance la aplicación demo con OpenTelemetry Collector, Jaeger y/o Tempo integrados (en modo Docker).
  - Simule peticiones HTTP desde el frontend.

- Visualice en Jaeger o Tempo:
  - Tiempos de respuesta por servicio.
  - Dependencias entre servicios.
  - Servicios con mayor latencia.
- Instrumentación adicional (manual):
  - Agregue trazas personalizadas en el backend usando el SDK de OpenTelemetry para Node.js o Python.
  - Use atributos (`span.setAttribute`) y eventos (`span.addEvent`) para marcar pasos relevantes (ej: acceso a DB, fallos).
- Feedback técnico esperado:
  - Mapas de servicios correctamente generados.
  - Líneas de tiempo de ejecución que evidencian el flujo real.
  - Capacidad de detectar “outliers” o tiempos muertos.
  - Opcional: integración con Grafana Tempo
  - Use Grafana como frontend para Tempo y correlaciona métricas y trazas.