



## **Sistema de Monitoreo de Almacén Automatizado con Kubernetes**

Juan Alberto Guillen Albarracin

Sistemas distribuidos

Cristian Timbi

## 1. Introducción

Este documento presenta el diseño, implementación y validación de un sistema de monitoreo en tiempo real para un almacén automatizado. El sistema se despliega sobre Kubernetes y demuestra persistencia de datos, seguridad básica y resiliencia ante fallos.

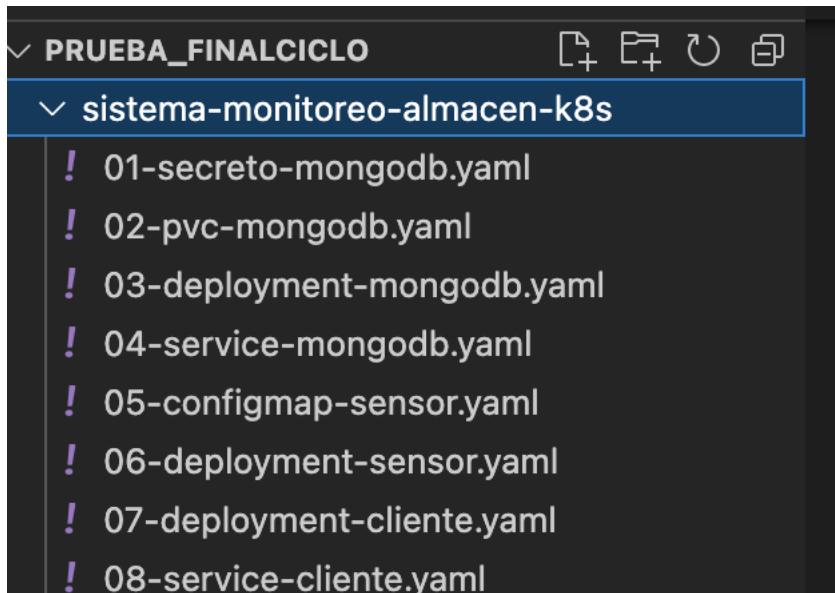
## 2. Tabla de Tecnologías Seleccionadas

| Componente     | Tecnología            | Justificación                          |
|----------------|-----------------------|--|
| Orquestación   | Kubernetes            | Gestión de contenedores y resiliencia  |
| Persistencia   | MongoDB               | Base NoSQL orientada a documentos JSON |
| Almacenamiento | PersistentVolumeClaim | Persistencia de datos ante fallos      |
| Productor      | Bash + mongosh        | Generación simple y continua de datos  |
| Visualización  | Mongo Express         | Interfaz web existente para MongoDB    |

## 3. Manifiestos de Kubernetes

Se implementaron manifiestos Kubernetes para Secrets, PersistentVolumeClaim, Deployments y Services. El script del microservicio productor se definió en un ConfigMap y se revisó especialmente la sintaxis de conexión autenticada a la base de datos.

(Los archivos YAML completos se incluyen en el repositorio Git)



## 4. Evidencias de Funcionamiento

### 4.1 Interfaz Web de Visualización

Captura de la interfaz web (Mongo Express) mostrando los datos almacenados con el formato JSON solicitado.

The screenshot shows the Mongo Express interface for the 'almacen\_db' database. At the top, there are buttons for 'View', 'Export', 'JSON', 'Import', and 'lecturas\_sensor'. Below this is a 'Database Stats' section with the following data:

| Collections (incl. system.namespaces) | 1          |
|---------------------------------------|------------|
| Data Size                             | 73.6 KB    |
| Storage Size                          | 69.6 KB    |
| Avg Obj Size #                        | 91.0 Bytes |
| Objects #                             | 809        |
| Indexes #                             | 1          |
| Index Size                            | 53.2 KB    |

The screenshot shows the Mongo Express interface for the 'lecturas\_sensor' collection within the 'almacen\_db' database. At the top, there is a search bar and a red banner stating 'Delete all 819 documents retrieved'. Below this is a table with the following data:

| _id                      | sensor_id | valor | timestamp            |
|--------------------------|-----------|-------|----------------------|
| 6983c79a73a0353c38ce5b0  | rbt-01    | 71    | 2026-02-04T22:28:08Z |
| 6983c7fd97a6540acfc5b0   | rbt-01    | 27    | 2026-02-04T22:28:12Z |
| 6983c801a668da6878ce5b0  | rbt-01    | 43    | 2026-02-04T22:28:17Z |
| 6983c8051abd997fc8ce5b0  | rbt-01    | 53    | 2026-02-04T22:28:20Z |
| 6983c809a5498aca38ce5b0  | rbt-01    | 0     | 2026-02-04T22:28:24Z |
| 6983c802d2fafccb7cfc5b0  | rbt-01    | 17    | 2026-02-04T22:28:28Z |
| 6983c8109e04c272698ce5b0 | rbt-01    | 0     | 2026-02-04T22:28:32Z |
| 6983c8144cded2b20fca5b0  | rbt-01    | 79    | 2026-02-04T22:28:35Z |
| 6983c815ea2fb2b5078ce5b0 | rbt-01    | 51    | 2026-02-04T22:28:39Z |
| 6983c81c1e5b4249538ce5b0 | rbt-01    | 56    | 2026-02-04T22:28:43Z |

## 4.2 Persistencia del Volumen

Evidencia de consola que muestra el estado del PersistentVolumeClaim en condición Bound.

```
● (base) juanguillenalbarracin@Juans-MacBook-Pro sistema-monitoreo-almacen-k8s % kubectl get pvc
  NAME      STATUS    VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS
  GE
  pvc-mongodb  Bound    pvc-bd8a91f9-0cf8-4605-adee-b94784230c7c  1Gi       RWO          standard      <unset>
  4m
● (base) juanguillenalbarracin@Juans-MacBook-Pro sistema-monitoreo-almacen-k8s %
```

## 5. Prueba de Resiliencia y Recuperación

Para validar la resiliencia del sistema, se eliminó manualmente el Pod de MongoDB. Kubernetes recreó automáticamente el Pod y los datos previamente almacenados permanecieron disponibles gracias al volumen persistente.

```
● (base) juanguillenalbarracin@Juans-MacBook-Pro sistema-monitoreo-almacen-k8s % kubectl delete pod -l app=bd-almacen  
pod "bd-almacen-564c9f49d5-jwhm9" deleted
```

MM75

```
④ (base) juanguillenalbarracin@Juans-MacBook-Pro sistema-monitoreo-almacen-k8s % kubectl get pods -w
```

| NAME                                   | READY | STATUS           | RESTARTS      | AGE   |
|--|-------|------------------|---------------|-------|
| bd-almacen-564c9f49d5-r8tmr            | 1/1   | Running          | 0             | 49s   |
| cliente-visualizacion-854c8bfc55-jrxrt | 1/1   | Running          | 0             | 5m19s |
| db-ordenes-f6d576db-7ztn8              | 1/1   | Running          | 1 (6m40s ago) | 8d    |
| db-productos-9f78f8dbf-r4h8d           | 1/1   | Running          | 1 (6m40s ago) | 8d    |
| gateway-api-7b77bdff5f-cd6jn           | 1/1   | Running          | 2             | 8d    |
| ms-envios-d59959749-ljz2x              | 1/1   | Running          | 0             | 8d    |
| ms-ordenes-bc85dfbc8-hzjqz             | 1/1   | Running          | 0             | 8d    |
| ms-productos-77d9f867dc-wmgqn          | 1/1   | Running          | 0             | 8d    |
| sensor-almacen-f8ff6878d-28m4s         | 0/1   | CrashLoopBackOff | 3 (48s ago)   | 5m19s |
| sensor-almacen-f8ff6878d-28m4s         | 1/1   | Running          | 4 (50s ago)   | 5m21s |

^C

```
④ (base) juanguillenalbarracin@Juans-MacBook-Pro sistema-monitoreo-almacen-k8s % kubectl get pods -w
```

| NAME                                   | READY | STATUS           | RESTARTS      | AGE   |
|--|-------|------------------|---------------|-------|
| bd-almacen-564c9f49d5-r8tmr            | 1/1   | Running          | 0             | 12s   |
| cliente-visualizacion-854c8bfc55-jrxrt | 1/1   | Running          | 0             | 4m42s |
| db-ordenes-f6d576db-7ztn8              | 1/1   | Running          | 1 (6m3s ago)  | 8d    |
| db-productos-9f78f8dbf-r4h8d           | 1/1   | Running          | 1 (6m3s ago)  | 8d    |
| gateway-api-7b77bdff5f-cd6jn           | 1/1   | Running          | 2             | 8d    |
| ms-envios-d59959749-ljz2x              | 1/1   | Running          | 0             | 8d    |
| ms-ordenes-bc85dfbc8-hzjqz             | 1/1   | Running          | 0             | 8d    |
| ms-productos-77d9f867dc-wmgqn          | 1/1   | Running          | 0             | 8d    |
| sensor-almacen-f8ff6878d-28m4s         | 0/1   | Error            | 3 (3m15s ago) | 4m42s |
| sensor-almacen-f8ff6878d-28m4s         | 0/1   | CrashLoopBackOff | 3 (12s ago)   | 4m43s |

^C

Kubernetes Dashboard

lecturas\_sensor - Mongo Express

127.0.0.1:61471/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=...

Gmail YouTube Twitch Netflix Maps Traducir WhatsApp Universidad Polité... AVAC

kubernetes default Buscar

Cargas de trabajo

Despliegues

Running: 9

Pods

Running: 9

Replica Sets

Running: 9

Cargas de trabajo

Despliegues

| Nombre                | Imagenes                         | Etiquetas                        | Pods  | Fecha de creación |
|-----------------------|----------------------------------|----------------------------------|-------|-------------------|
| cliente-visualizacion | mongo-express:1.0.2-20           | -                                | 1 / 1 | 55 minutes ago    |
| bd-almacen            | mongo:7                          | -                                | 1 / 1 | 55 minutes ago    |
| sensor-almacen        | mongo:7                          | -                                | 1 / 1 | 55 minutes ago    |
| ms-envios             | juanguillena/ms-envios:latest    | io.kompose.service: ms-envios    | 1 / 1 | 13 days ago       |
| ms-ordenes            | juanguillena/ms-ordenes:latest   | io.kompose.service: ms-ordenes   | 1 / 1 | 13 days ago       |
| ms-productos          | juanguillena/ms-productos:latest | io.kompose.service: ms-productos | 1 / 1 | 13 days ago       |
| db-ordenes            | postgres:16-alpine               | io.kompose.service: db-ordenes   | 1 / 1 | 13 days ago       |
| db-productos          | node:16-alpine                   | io.kompose.service: db-productos | 1 / 1 | 13 days ago       |

## **6. Conclusiones**

El proyecto cumple con todos los requerimientos establecidos, demostrando una arquitectura desacoplada, persistencia de datos, seguridad básica y recuperación automática ante fallos. La solución es adecuada para un entorno académico y permite comprender el funcionamiento de Kubernetes en sistemas distribuidos.

## 7. Tabla de Tecnologías Seleccionadas

| Componente     | Tecnología            | Justificación                          |
|----------------|-----------------------|--|
| Orquestación   | Kubernetes            | Gestión de contenedores y resiliencia  |
| Persistencia   | MongoDB               | Base NoSQL orientada a documentos JSON |
| Almacenamiento | PersistentVolumeClaim | Persistencia de datos ante fallos      |
| Productor      | Bash + mongosh        | Generación simple y continua de datos  |
| Visualización  | Mongo Express         | Interfaz web existente para MongoDB    |

