

Contrato Interadministrativo Nro. 3021063 de 2021
Suscrito Entre la Empresa de Telecomunicaciones de Bogotá y el Fondo Financiero
Distrital de Salud - Secretaría Distrital de Salud

Fase 3 del Proyecto
Grupo de Entregables E20 – Infraestructura, Soporte, Mantenimiento y Operación

Despliegue y Configuración de Infraestructura de Ambiente de Producción de Nuevos
Desarrollos

Código:
SISEM-3-E20-6

Bogotá, 25 de agosto de 2025

FICHA DE IDENTIFICACIÓN DEL DOCUMENTO

Título del Documento	“Despliegue y configuración de infraestructura de ambiente de producción de nuevos desarrollos”	
Resumen	Este entregable presenta el plan de despliegue y configuración del Sistema de Emergencias Médicas. El entregable forma parte de las actividades del paso a producción.	
Código	SISEM-3-E20-6	
Autor	ETB	
Estado	Final	25/08/2025
Revisado por		

CONTROL DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
V1	Versión inicial	José Daniel Peña	22/11/2023
Final	Versión final	Carlos Caicedo	25/08/2025

DERECHOS DE AUTOR

Derechos de autor
ETB. Todo el material de este entregable puede ser utilizado libremente por la SDS y las personas a quienes ésta asigne en el marco del convenio citado.

CRÉDITOS

Créditos
Grupos de Diseño de Arquitecturas de ETB.

Tabla de Contenido

1.	INTRODUCCIÓN	5
2.	OBJETIVOS	6
3.	ALCANCE	7
4.	AUDIENCIA.....	8
5.	COMPONENTES	9
6.	GENERACIÓN DE CONTENEDORES.....	10
7.	DESPLIEGUE DEL PRODUCTO.....	12
8.	INSTALAR REALM DE KEYCLOAK.....	13
9.	CONECTARSE A LAS BASES DE DATOS.....	14
10.	VISUALIZACIÓN DE LOS LOGS	14
11.	SERVICIO NOTIFICACIONES.....	15

Tabla de Ilustraciones

ILUSTRACIÓN 1 ARQUITECTURA PROYECTO SISEM	9
ILUSTRACIÓN 2 SCRIPT DE DOCKER FILES.....	10
ILUSTRACIÓN 3 INICIO DEL REALM.....	13
ILUSTRACIÓN 4 VISUALIZACIÓN DE LOGS DE LOS PODS	14

1. Introducción

Este documento ha sido elaborado como una guía detallada que aborda el despliegue y la configuración del entorno de producción del proyecto SISEM. Su objetivo es proporcionar un manual completo que cubra desde los elementos esenciales de la infraestructura hasta los procesos clave de despliegue y configuración.

En estas páginas, se presentan el proceso para llevar a cabo la generación de contenedores, el despliegue del producto en el ambiente de producción y la implementación de componentes esenciales, como la instalación de Realm de Keycloak y la correcta conexión con las bases de datos. Asimismo, se aborda la visualización de registros (logs), ofreciendo un enfoque holístico para lograr un ambiente operativo y funcional del sistema.

2. Objetivos

Este documento se centra en exponer de manera los principales aspectos del proceso de contenerización, generación de imágenes y despliegue de aplicaciones en el ambiente de producción del proyecto SISEM.

Se abordan puntos como la explicación detallada de las variables de entorno utilizadas, la configuración del chart Helm y el proceso de despliegue de aplicaciones, incluyendo la visualización efectiva de registros (logs).

3. Alcance

Este documento está diseñado para servir como guía exhaustiva para el despliegue y la configuración del entorno de producción del proyecto SISEM. Su objetivo principal es ofrecer instrucciones sobre los procesos esenciales que abarcan desde la contenerización y generación de imágenes hasta la instalación y configuración de componentes críticos como Realm de Keycloak, la conexión con bases de datos, y la gestión y visualización de logs.

Este manual pretende ser una referencia detallada para profesionales de TI, desarrolladores y equipos involucrados en el proyecto, proporcionándoles una guía estructurada y completa para llevar a cabo con éxito el despliegue y la configuración del entorno de producción del proyecto SISEM.

4. Audiencia

Este documento está dirigido a los usuarios técnicos del sistema, más específicamente a aquellos que se desempeñan en la Dirección de Tecnologías de la Información y la Comunicación - TIC - , en su calidad de solicitantes de servicios, procesos, datos, información y demás herramientas funcionales y no funcionales que les debe proveer la solución contratada, en el marco del SISEM.

5. Componentes

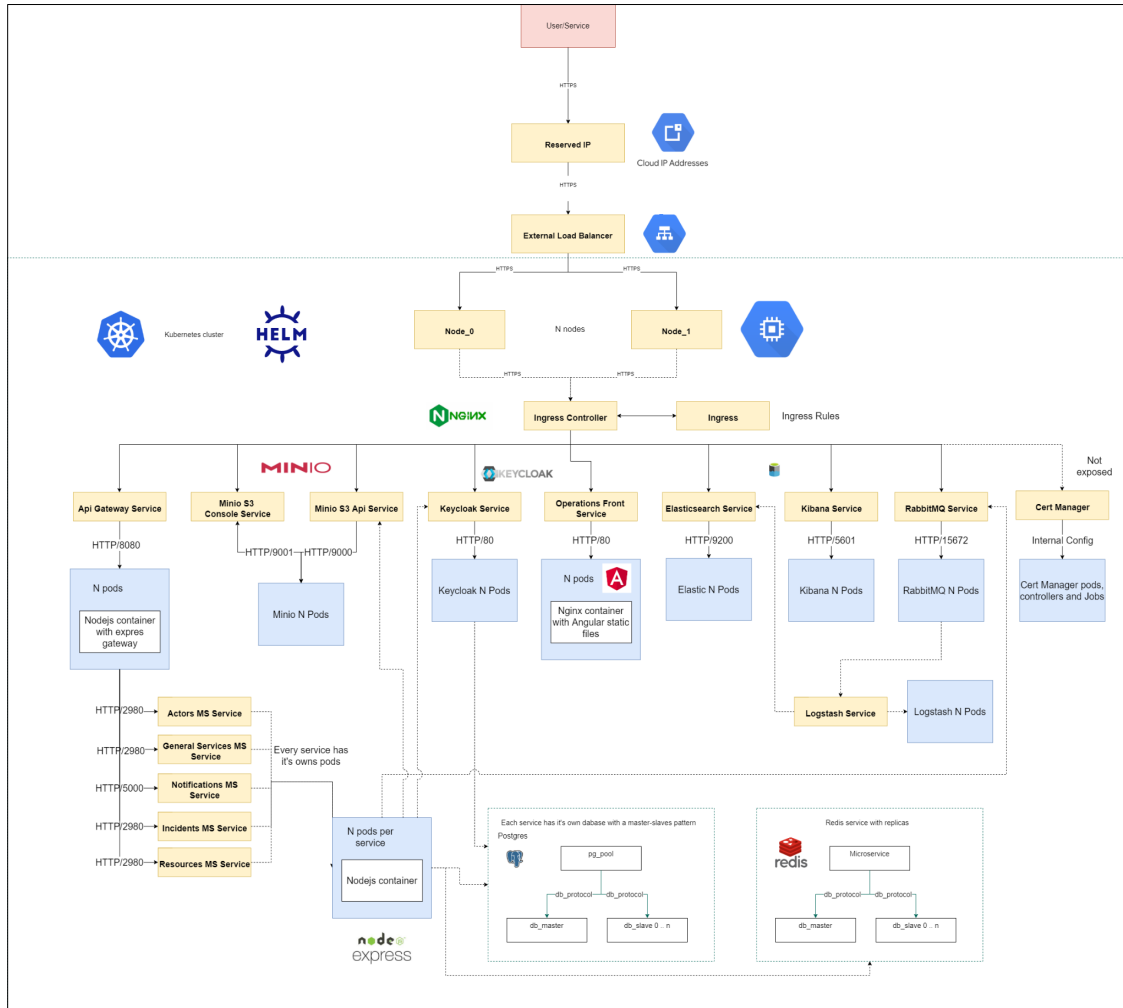


Ilustración 1 Arquitectura proyecto SISEM

El diagrama anterior ilustra la vista de componentes y conectores de los componentes que conforman el proyecto, varios de ellos desarrollados y otros como proyectos externos open source. Al mismo tiempo se ilustra la configuración del sistema bajo kubernetes, explicando la configuración de acceso y despliegue de los servicios dentro y fuera del cluster.

6. Generación de Contenedores

Los servicios desarrollados en el entorno de SISEM están basados en NodeJS. Por consiguiente, los siguientes pasos son aplicables a todos los servicios diseñados para este entorno.

Los contenedores se generan mediante imágenes de Docker, utilizando los archivos Dockerfile. Estos archivos, entre otras funcionalidades, se encargan de compilar los proyectos de Typescript a Javascript en dos fases de ejecución: una fase que utiliza las dependencias de desarrollo y otra que no. Esta técnica permite reducir significativamente el tamaño de las imágenes generadas, optimizando así el despliegue de los contenedores.

```
FROM node:16-alpine3.11 AS build
ARG npm_cred
ARG node_prune="true"
ARG docker_image_tag
ENV APP_VERSION=$docker_image_tag

WORKDIR /usr/src/app
COPY .npmrc package*.json ./
COPY . .
RUN npm config set //verdaccio.smartsoftlabs.com/:_authToken=$npm_cred
RUN npm config set @smartsoft:registry https://verdaccio.smartsoftlabs.com
RUN npm install
RUN npm run tsc

# remove development dependencies
RUN npm prune --production

# run node prune
#RUN chmod +x node-prune.sh
#RUN sh node-prune.sh

## this is stage two , where the app actually runs

FROM node:16-alpine3.11 AS pack
ARG npm_cred
ARG docker_image_tag
ENV APP_VERSION=$docker_image_tag

WORKDIR /usr/src/app
COPY .npmrc package*.json ./
RUN npm config set //verdaccio.smartsoftlabs.com/:_authToken=$npm_cred
RUN npm config set @smartsoft:registry https://verdaccio.smartsoftlabs.com
RUN npm install --only=prod
COPY --from=build /usr/src/app/build ./build
COPY --from=build /usr/src/app/src/static ./build/static

EXPOSE 5000

CMD ["npm", "run", "start"]
```

Ilustración 2 Script de docker files

La imagen anterior muestra el proceso detallado a seguir:

1. **Copia del código fuente y dependencias:** Incluye el archivo package.json o cualquier valor necesario para cargar datos predeterminados
2. **Configuración de variables de autenticación:** Esto permite la instalación de dependencias provenientes de un repositorio privado
3. **Instalación de dependencias:** Realizada con npm o pnpm
4. **Transpilación de código Typescript a Javascript:** Esta fase convierte el código Typescript a Javascript
5. **Fase de compilación (fase 2):** Repite los pasos 1 y 2 en un contenedor separado. En el paso 3, instala únicamente las dependencias de producción, excluyendo las de desarrollo, como Typescript
6. **Transferencia del código generado:** En el contenedor de la fase 2, se copia el código generado en la fase 1, asegurando así el código compilado sin las dependencias de desarrollo
7. **Definición de comandos de inicio del contenedor:** Se establecen los comandos básicos necesarios para iniciar el contenedor
8. **Exportación de la imagen generada:** Este paso se realiza desde la consola, exportando la imagen a un repositorio privado

Esta ilustración es aplicable a todos los backends desarrollados en el proyecto SISEM. En el caso del frontend, el proceso es similar, pero se diferencia en que el contenedor de la fase 2 utiliza una imagen de Nginx como base para exponer el proyecto Angular generado a través del servidor web mencionado.

Cada contenedor generado se exporta con una etiqueta que representa su versión correspondiente. Las variables de entorno se cargan directamente mediante docker-compose o se incluyen en un archivo de configuración de Kubernetes. Estas variables no forman parte del proceso de compilación para evitar dependencias innecesarias.

7. Despliegue del Producto

El sistema originalmente se desplegó en Docker Compose, lo que significa que cuenta con soporte para esa forma de despliegue. Sin embargo, el enfoque actual se centra en la configuración y despliegue mediante Kubernetes. Por lo tanto, a continuación, se detalla específicamente la configuración de Kubernetes y cómo está estructurada. Este extracto del readme.md se encuentra en la carpeta que contiene la configuración de Kubernetes para una mayor información detallada.

Consideraciones importantes:

- Se requiere una cuenta en GCP para llevar a cabo las siguientes acciones:
 - Crear un clúster de Kubernetes (GKE)
 - Establecer un balanceador de carga
 - Obtener una IP pública estática para el balanceador de carga
- Se utiliza Helm para crear los "charts" de Kubernetes, simplificando así el proceso de despliegue:
 - Todas las variables modificables están unificadas en un archivo values.yaml; se recomienda no realizar cambios directos dentro de la carpeta "templates"
 - Hasta el momento de la redacción de este documento, no se ha implementado el escalado automático, ya que se requieren datos operativos reales para una aplicación eficiente

El archivo values.yaml contiene todas las configuraciones necesarias para el funcionamiento del sistema, tanto para la configuración inicial como para la operación normal. Entre las configuraciones disponibles, las más destacables son:

- Cantidad de réplicas.
- Imágenes.
- Secretos.
- Variables de entorno de los servicios.
- Dominios.

El proceso de despliegue de la aplicación se resume en los siguientes pasos:

1. Resolver los requisitos de despliegue
2. Instalar las dependencias de Helm
3. Instalar el cert manager en el clúster (si se manejan certificados automáticos)
4. Actualizar values.yaml con la configuración necesaria, como la IP del balanceador de carga
5. Instalar el "chart", lo que generará también trabajos adicionales para configurar las bases de datos
6. Instalar el Realm de Keycloak (consultar detalles en el siguiente punto)
7. Actualizar values.yaml con las credenciales de los clientes de Keycloak, así como las credenciales de Minio
8. Actualizar los servicios (re-despliegue)

9. Crear el índice de Elastic/Kibana.

Para obtener detalles adicionales, consultar el readme.md en la carpeta de Kubernetes, donde se encuentra toda la configuración de despliegue.

8. Instalar Realm de keycloak

Para instalar el realm de Keycloak, se puede exportar e instalar como un archivo JSON. Sin embargo, su instalación inicial se ejecuta únicamente la primera vez que el servicio se pone en marcha. Cualquier ajuste adicional requerirá intervención manual. Para abordar este problema, se ha desarrollado un Script que contiene la configuración completa de Keycloak para actualizar el realm, clientes, mappers, roles, grupos y medidas de seguridad, entre otras configuraciones.

A continuación, se detallan los pasos para instalar el realm:

1. Diríjase a la carpeta "setup-keycloak-realm".
2. Instale las dependencias ejecutando npm install.
3. Valide la configuración que se va a utilizar. Para ello, vaya a la carpeta "config-data" y abra el archivo "keycloak.realm.data.ts".
4. Encontrará dos variables esenciales al principio del archivo: "AvailableConfigs" y "EnvironmentData".
5. La primera variable especifica la configuración de los entornos existentes para conectarse y ejecutar la configuración. La segunda variable indica qué entorno se seleccionará para ejecutar el Script.
6. Asegúrese de que la configuración del entorno sea correcta y esté seleccionada.
7. Ejecute el script "startup" dentro del archivo package.json en esa carpeta.
8. En la primera ejecución del script, pueden aparecer errores. Ejecute el Script varias veces hasta obtener el resultado deseado (consulte la imagen para referencia).

```
rolesPerClient []
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_general_services_ms' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_operations_front' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_resources_ms' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_incidents_ms' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_notifications-service' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] DELETING FOUND ROLES FOR CLIENT 'sisem_real_actors_ms' AND CREATING NEW ONES
[validateOrCreateKeycloakRealmAndConfig] Setting groups config
[validateOrCreateKeycloakRealmAndConfig] Setting service accounts config
[validateOrCreateKeycloakRealmAndConfig] DONE

Process finished with exit code 0
```

Ilustración 3 Inicio del realm

9. Puede ejecutar el script tantas veces como sea necesario, siempre y cuando se complete el proceso hasta el final, tal como se muestra en la imagen anterior.

Es importante destacar que solo se debe modificar la configuración de los entornos. No se debe alterar ningún aspecto propio de Keycloak, ya que esto podría resultar en errores inesperados relacionados con el rechazo de peticiones de usuarios o servicios debido a problemas de autenticación y autorización. Si se desean agregar nuevos roles o permisos, se recomienda consultar con el desarrollador para evitar inconvenientes.

9. Conectarse a las Bases de Datos

Como se mencionó previamente, las bases de datos están segmentadas y no son accesibles de forma pública. Para establecer una conexión, se debe crear un túnel entre el cluster y la máquina local utilizando el siguiente comando:

```
kubectrl port-forward -n sisem-pre-prod svc/{nombre-servicio-pgpool} puerto-local:5432
```

Reemplace '{nombre-servicio-pgpool}' por el nombre del servicio de pg-pool correspondiente a la base de datos a la que desea acceder. Además, especifique un 'puerto-local' disponible en su máquina local para la conexión. Tenga en cuenta que este puerto debe estar libre durante la conexión.

10. Visualización de los Logs

Gracias a la facilidad que proporciona GCP, para visualizar los registros de los servicios, acceda al servicio o directamente al pod que desea revisar y haga clic en 'Registros'.

Nombre	Estado	Tipo	Extremos	Pods	Espacio de nombres	Clústeres
sisem-actors-ms-service	OK	IP del clúster	10.40.8.248	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-api-gateway-service	OK	IP del clúster	10.40.10.78	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-cert-manager	OK	IP del clúster	10.40.12.218	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-cert-manager-webhook	OK	IP del clúster	10.40.5.199	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-elasticsearch-service	OK	IP del clúster	10.40.4.95	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-elasticsearch-service-headless	OK	IP del clúster	Ninguno	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-general-services-ms-service	OK	IP del clúster	10.40.6.9	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-incidents-ms-service	OK	IP del clúster	10.40.4.158	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-keycloak	OK	IP del clúster	10.40.12.6	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-keycloak-headless	OK	IP del clúster	Ninguno	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-kibana-service	OK	IP del clúster	10.40.3.32	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-logstash-service-headless	OK	IP del clúster	Ninguno	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-minio	OK	IP del clúster	10.40.1.51	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-minio-console	OK	IP del clúster	10.40.5.209	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-minio-svc	OK	IP del clúster	Ninguno	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-nginx-ingress-controller	OK	Balanceador de cargas externo	35.196.154.63.80	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-nginx-ingress-controller-admission	OK	IP del clúster	10.40.8.175	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-notifications-ms-service	OK	IP del clúster	10.40.11.206	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-operations-service	OK	IP del clúster	10.40.1.136	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-pg-actors-ms-pgpool	OK	IP del clúster	10.40.7.9	1/1	sisem-pre-prod	sisem-pre-prod-cluster-1
sisem-pg-actors-ms-postgresql	OK	IP del clúster	10.40.5.29	3/3	sisem-pre-prod	sisem-pre-prod-cluster-1

Ilustración 4 Visualización de logs de los pods

11. Servicio Notificaciones

El servicio de notificaciones recibe archivos de entrada que contienen credenciales para la integración con los proveedores de servicios, así como plantillas para los correos, en caso de ser utilizadas.

Fin del documento.