

# Sistema Centralizado de Logs y Monitoreo

Manual de Instalación

Versión 1.0

Septiembre 2023

## INTRODUCCIÓN

El presente documento tiene como finalidad mostrar el proceso que se debe seguir para poner en marcha los productos que componen el sistema de logs y monitoreo.

Este sistema consta de tres productos software diferentes:

1. wsLogs. – Web Service que se encarga de recibir los logs de los diferentes sistemas de la Cooperativa y de registrar tanto los logs en archivos de texto plano, como en las bases de datos de MongoDB, también es el encargado de obtener la información desde MongoDB para el Frontend.
2. ApiGatewayLogs. – Este es el enrutador de las peticiones desde el Frontend hacia el web Service, además es el que da validez a las peticiones haciendo comprobaciones de seguridad.
3. app\_web\_logs. — Esta aplicación web se usa para revisar los logs de los sistemas y monitorear la ejecución de los mismos, es el Frontend.
4. AccesoDatosGrpcMongo. – Este es el servicio que permite acceder a MongoDB para obtener las bases de datos, colecciones y registros.

***Nota:*** *El enlace de acceso al sistema es:* [*http://192.168.55.63:5223/*](http://192.168.55.63:5223/)

## PARAMETRIZACIONES Y CONFIGURACIONES

Se debe parametrizar la matriz:

“Matriz\_parametros.xlsx”

## REFERENCIAS AMBIENTE

Las ramas se encuentran con las referencias al servidor de desarrollo 192.168.55.62, en el puerto 6021. Actualmente no se tiene un ambiente de Pruebas o QA, por lo que las pruebas se realizan directamente en el servidor de desarrollo.

## SITIOS (EJECUTABLES)

Las ramas se encuentran con las referencias al servidor de desarrollo 192.168.55.62, en el puerto 6021 (wsLogs) y al 192.168.55.63 en el puerto: 6023 (Api Gateway). Actualmente no se tiene un ambiente de Pruebas o QA, por lo que las pruebas se realizan directamente en el servidor de desarrollo.

1. wsLogs:
   1. El sistema está desarrollado usando las características DevOps que brinda Jenkins, por lo que cualquier cambio que se realice en la rama sincronizada con git se verá reflejado en el producto final de forma inmediata.
   2. Para poner en producción se debería solicitar a quien corresponda la creación de un Docker que contendrá el producto y la correspondiente sincronización entre Jenkins y la rama máster, para luego realizar una solicitud de fusión de la rama de desarrollo con la máster para que el sistema se ponga en línea.
   3. Se debe tener los parámetros ya configurados antes de correr el sistema.
   4. Este sistema debe estar configurado como servicio usando Docker Swarm.
2. apiGatewayLogs:
   1. El sistema está desarrollado usando las características DevOps que brinda Jenkins, por lo que cualquier cambio que se realice en la rama sincronizada con git se verá reflejado en el producto final de forma inmediata.
   2. Para poner en producción se debería solicitar a quien corresponda la creación de un Docker que contendrá el producto y la correspondiente sincronización entre Jenkins y la rama máster, para luego realizar una solicitud de fusión de la rama de desarrollo con la máster para que el sistema se ponga en línea.
   3. Se debe tener los parámetros ya configurados antes de correr el sistema.
   4. El archivo “ocelot.json” ya apunta al servidor y puerto asignado para el proyecto. Si las direcciones y los puertos de los servicios cambian, se debe modificar el archivo ocelot para que apunte a las nuevas direcciones.
3. app\_web\_logs:
   1. El sistema está desarrollado usando las características DevOps que brinda Jenkins, por lo que cualquier cambio que se realice en la rama sincronizada con Git se verá reflejado en el producto final de forma inmediata.
   2. Para poner en producción se debería solicitar a quien corresponda la creación de un Docker que contendrá el producto y la correspondiente sincronización entre Jenkins y la rama máster, para luego realizar una solicitud de fusión de la rama de desarrollo con la máster para que el sistema se ponga en línea.
   3. Se debe tener los parámetros ya configurados antes de correr el sistema.
4. AccesoDatosGrpcMongo:
   1. El sistema está desarrollado usando las características DevOps que brinda Jenkins, por lo que cualquier cambio que se realice en la rama sincronizada con Git se verá reflejado en el producto final de forma inmediata.
   2. Para poner en producción se debe solicitar el Merge Request en el Git para que los cambios realizados en el sistema se actualicen en producción.
   3. Este sistema debe estar configurado como servicio usando Docker Swarm.

## BASE DE DATOS

1. Se debe tener acceso al servidor GRPC de Mongo desde el wsLogs, también, este último debe tener configurado el acceso a su propia base de datos para realizar el registro de las conexiones.

## LOGS

* El sistema registra logs de las peticiones realizadas.
* Cada sistema tiene su propio directorio de logs de texto.
* Cada sistema tiene su base de datos de logs en MongoDB.
* Se debe configurar un volumen de memoria que este vinculado a la carpeta de registro de logs del wsLogs para que el contenido de los directorios no se pierda en las actualizaciones.

## LIBRERIAS

Todas las librerías usadas en el proyecto se pueden descargar desde NuGet. Se recomienda que si se actualizan las librerías se deberá comprobar que no haya afectado alguna funcionalidad.

## NOTIFICACIONES

El sistema en la primera versión no realiza notificaciones.

## VERSIONADO

Para la puesta en producción se debe realizar una solicitud de fusión con la rama máster y se debe versionar las ramas especificadas en el correo con su descripción de etiqueta correspondiente.

## ESTRATEGIA DE PUESTA EN PRODUCCIÓN

Previo a la puesta en producción se debe registrar el sistema en el Sistema de Administración, con la siguiente información:

|  |  |
| --- | --- |
| **Campo** | **Valor** |
| Nombre del Sistema | MONITOREO |
| Tipo de sistema | INTERNO |
| URL del Sistema | http://192.168.55.63:6223/ |
| Versión Actual | 1.0.0 |
| Descripción | Sistema web para monitoreo de logs, inquietudes, incidencias, soporte etc. |
| Sistema Aplicación | app\_web\_logs |
| Base de Datos | MongoDB - ws\_logs |
| Ultimas modificaciones | Registrar conexiones a MongoDB.|Obtener lista de bases de datos (se obtienen las bases asignadas a las credenciales de conexión).|Obtener logs de Mongo y de Texto (se puede descargar).|Realizar seguimiento de transacciones. |
| Programador Principal | JHON ALEXANDER CARRIÓN PIEDRA |
| Dueño del producto | Director de Tecnología |
| Documentación Usuario | Manual en PDF, actualizado, accesible desde el sistema |
| Documentación Técnica | Propuesta del sistema, Documento de Especificación de Requisitos - IEEE830, Diagramas de flujo de las funcionalidades del sistema, Diagrama de arquitectura |
| Plataforma | Web |
| Lenguaje de Programación | C# + ReactJS |
| Porcentaje de migración estándares | 100 |

Luego se deben crear las siguientes funcionalidades desde el Sistema de Administración al nuevo sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Funcionalidad** | **Tipo Funcionalidad** | **URL** | **Descripción** |
| LOGS | MENU PRINCIPAL | /logs | Acceder a la funcionalidad de la lista de Logs |
| SEGUIMIENTO | MENU PRINCIPAL | /seguimientotran | Acceder a la funcionalidad de dar seguimiento a una transacción |
| VER\_LOGS\_DE\_MONGO | PERMISO ACCIÓN | /logsMongo | Habilitar la funcionalidad de ver logs de MongoDB |
| VER\_LOGS\_DE\_TEXTO | PERMISO ACCIÓN | /logsTexto | Habilitar la funcionalidad de ver los archivos de texto |
| VER\_CONTENIDO\_LOGS\_DE\_TEXTO | PERMISO ACCIÓN | /contenidoLog | Habilitar la funcionalidad de ver los logs en los archivos de texto |
| DESCARGAR\_LOGS\_DE\_TEXTO | PERMISO ACCIÓN | /logsTexto | Habilitar la funcionalidad de descargar un archivo de logs de texto |

Para que los usuarios puedan acceder, se debe agregar el **PERFIL DE CONSULTAS** al sistema creado; una vez agregado el perfil, se debe proceder a asignarle los permisos antes creados; y para finalizar se procede a agregar los usuarios a dicho perfil.

Para la puesta en producción del sistema, se debe primero realizar una evaluación en un ambiente controlado (Transacciones Controladas), en donde se puede evaluar que los componentes del sistema funcionan correctamente y se integran a los demás sistemas. Además, al centralizar el registro de logs de todos los sistemas, se debe garantizar que el servicio no represente un cuello de botella para las transacciones, por lo que se aconseja generar varias instancias del servidor para balancear la carga.

## NOTAS ADICIONALES

* Se debe tener en cuenta lo indicado en el punto 9 para que todo esté listo antes de correr el sistema.
* Las configuraciones de parámetros y de base de datos deben realizarse antes de cualquier otra tarea.
* Se debe primero correr el GRPC de Mongo, segundo el wsLogs, tercero el Api Gateway y finalmente la Aplicación Web.
* Se debe verificar que se haya creado y configurado el volumen para las carpetas de logs.
* Se debe verificar que los sistemas tengan conexión entre ellos.
* Se debe verificar que todos los sistemas de la cooperativa tengan acceso al wsLogs para registrar sus logs.
* Si el sistema, los perfiles o los permisos no han sido registrados, los usuarios no podrán acceder al sistema.
* Se debe actualizar el archivo appsettings.json tanto de la app web logs como del api Gateway, así mismo de este último se debe actualizar también el archivo ocelot.json. si no se actualizan estos archivos correctamente en los volúmenes, el sistema presentará errores.