CARACTERISTICAS GENERALES - PLATAFORMA DE GESTIÓN DE CONTENEDORES E INTEROPERABILIDAD		
Item	**Grupo**	**Requerimiento**
1	Microservicios	La solución debe implementar el habilitar la traducción de múltiples protocolos del consumidor a un protocolo especifico del microservicio ofrecido a través de un API Gateway
		La solución debe implementar la publicación de microservicios que generen múltiples API para plataformas y clientes específicos con las funciones especificas y protocolos requeridos por cada plataforma
		La solución debe implementar un mecanismo de hacer la trazabilidad, uso y registro de actividades de los microservicios.
2	Integración	Debe admitir la transformación y el enrutamiento hacia / desde SOAP / HTTP a los servicios REST.
		Debe poder convertir mensajes a / desde: XML, objetos Java, JSON, REST, CSV.
		Debe permitir la composicion y transformación de datos a partir de EIP (Patrones de integración empresarial - o patrones de integración corporativa)
		Debe permitir el uso de código Java, Javascript, Kotlin, Groovy o XML para realizar la asignación / transformación de datos.
		Debe proporcionar componentes para la transformación utilizando modelos predefinidos (plantillas).
		La infraestructura debe distribuirse de modo que las integraciones, construidas a partir de EIP (patrones de integración empresarial) y conectores predefinidos, se implementen en la infraestructura nativa del contenedor para adaptarse y escalar rápidamente
		La solución debería exponer métricas con integración nativa al software Prometheus
3	API Management	Debe ser posible gestionar la creacion de un token de acceso, eligiendo su alcance, permiso y otras cualidades a nivel de autenticacion
		La solución de administración de API debe admitir su instalación a través del operador de Kubernetes, siguiendo el patrón del operador.
		La solución de administración de API debe admitir la instalación de API Gateways en contenedores tanto dentro de una plataforma Kubernetes o utilizando solo un motor de contenedor aprobado por la especificación OCI.
		La línea de comando (CLI) de la solución de API Management debe tener la capacidad de: Importar API basadas en archivos en formato .csv Haga copias de las API de una cuenta a otra Cambiar la configuración de la API Crear API informando un archivo en el estándar de especificación OpenAPI Crear, enumerar, eliminar, exportar e importar planes de consumo para API Crear, enumerar, actualizar, eliminar API Crear, enumerar, actualizar y eliminar aplicaciones cliente, es decir, aplicaciones que consumen la API
		Debe permitir extraer datos de consumo de las API mediante el servicio REST o mediante la interfaz gráfica.
		Debe permitir ver las llamadas a la API y separar los códigos de retorno HTTP;
		La herramienta API Gateway debe proporcionar una interfaz de acceso API escalable;
		La herramienta API Gateway debe controlar la ejecución de llamadas, recopilar métricas, aplicar políticas y límites de ejecución;
		La solución de plataforma de administración de API debe cumplir con los siguientes requisitos mínimos con respecto a la definición, administración y documentación de las API: 1. Debe compartir, proteger, distribuir y controlar las API; 2. Tener gráficos dinámicos que contengan la información de uso de las API más utilizadas; 3. Tener gráficos dinámicos que contengan información sobre el uso de API para el día actual, las últimas 24 horas, los últimos 7 días y también en función del período a informar (parametrizable);
		Tener la capacidad de exportar datos consultados sobre estadísticas de uso de API, en el formato: CSV u hoja de cálculo Excel;
		La solución de plataforma de administración de API debe tener los siguientes requisitos mínimos para la administración de desarrolladores: 1. Tener la inclusión de desarrolladores a través de menús administrativos de la solución; 2. Realice la creación de flujos de aprobación de desarrolladores mediante la intervención manual de los administradores. 3. Tener interfaces administrativas gráficas para la gestión de desarrolladores;
4	General	La solución debe permitir certificados digitales
		Para permitir interoperabilidad debe habilitar transporte de mensajes y conectarse entre ellos. Los mecanismos de transporte deben incluir Java Messaging Service (JMS), Active MQ y asi mismo protocolos de comunicación tal como HTTP/HTTPS, SMTP, entre otros
		Integridad: Se debe garantizar mecanismos que permitan mantener la integridad de la información, expuesta a través de servicios de modificación de los datos.
		Confidencialidad: Se debe garantizar la protección frente a accesos no autorizados a información confidencial.
		Los desarrollos deben ser arquitectura a microservicios
		Soporte para la publicación de servicios como APIs a través de SOAP, REST, JSON y XML.
		Monitoreo y análisis de tráfico de las APIs
		Soporte para la gestión del ciclo de vida de las APIs: creación, publicación, bloqueo y retiro.
		Se debe contar con la característica de Single Sign On (SSO)
		Capacidad para el monitoreo y análisis de logs y errores