ACUERDO MARCO NUBE PÚBLICA FORMATO RFI
OPERACIÓN SECUNDARIA

Como primer paso en la adquisición de Servicios de Nube Pública, y de acuerdo con la necesidad que la Entidad Compradora posea, la Entidad Estatal debe registrar e identificar las siguientes características, lo anterior para reconocer claramente el requerimiento y de esta manera otorgar a los proveedores información necesaria para la correcta proyección de los servicios.

El registro de la información por parte de la Entidad garantiza que sean proporcionados suficientes datos al proveedor para dimensionar y proponer la solución que se adecúe a las necesidades de la Entidad, por lo tanto, Colombia Compra Eficiente recomienda que la Entidad registre la mayor cantidad de información posible (La Entidad podrá relacionar información en documentos adicionales):

1. Contexto

La Universidad Distrital Francisco José de Caldas se reconoce a sí misma como la institución de educación superior del Distrito Capital de Bogotá y de la Región Central de la República de Colombia, por consiguiente su visión de futuro está estrechamente ligada a los procesos de su entorno social. El proyecto educativo institucional encuentra sentido en el fortalecimiento estratégico de sus potencialidades académicas y en las posibilidades que ellas ofrecen al desarrollo de la región.

La Universidad Distrital Francisco José de Caldas deberá hacerse más competitiva ante pares del mundo académico. Por ello, con una visión estratégica ha decidido canalizar los esfuerzos y recursos en torno a cinco áreas académicas prioritarias: lo ambiental, la comunicación, la informatización, la educación y la producción.

La misión de la Universidad Distrital Francisco José de Caldas establece que la institución es un espacio social y una organización institucional, ente autónomo del orden distrital, que tiene entre sus finalidades la formación de profesionales especializados y de ciudadanos activos; la producción y reproducción del conocimiento científico, además de la innovación tecnológica y la creación artística. Impulsa el diálogo de saberes y promueve una pedagogía, capaz de animar la reflexión y la curiosidad de los estudiantes; además, fomenta un espíritu crítico en la búsqueda de verdades abiertas; en la promoción de la ciencia y la creación; asimismo, de la ciudadanía y la democracia; y alienta la deliberación, fundada en la argumentación y en el diálogo razonado.

A su vez la Oficina Asesora de Sistemas de la Universidad Distrital, encabeza todos los procesos relacionados con la planeación, proposición e implementación de la sistematización de actividades, procesos y tareas a nivel institucional. Esto, con el fin de facilitar y agilizar el funcionamiento administrativo, académico, operativo y de planeación de la Universidad, teniendo como premisa la satisfacción de las necesidades de la comunidad universitaria, bajo altos estándares de calidad.

En el año 2013 la Universidad Distrital a través de la Oficina Asesora de Sistemas comenzó a implementar soluciones en la nube de Amazon Web Services (AWS), durante este tiempo ha contado personal experto y certificado en diseño e implementación de soluciones en la nube. La Universidad Distrital ha sido un referente a nivel distrital, nacional y de Latinoamérica en lo referente a soluciones en la nube de AWS. La Universidad Distrital presentó la sesión DevOps: The Amazon Way en el evento AWS Public Sector Summit Bogotá. Además, en un esfuerzo colaborativo entre la Universidad Distrital y Amazon Web Services (AWS), cada semestre se imparte el curso Digital Government (DigiGov) diseñado para que los funcionarios públicos aprendan sobre conceptos de computación en la nube, propuesta de valor de la nube, casos de uso y entendimiento de los diferentes servicios disponibles a través de la nube de AWS, este curso cuenta con docentes acreditados AWS de la Universidad Distrital.

Actualmente la Universidad Distrital Francisco José de Caldas tiene gran parte de sus soluciones informáticas en la nube de AWS y cuenta con la experticia necesaria para administrar las cuentas AWS que soportan estas soluciones.

2. Objetivos principales (Entidad)

2.1. General

Se requiere para la presente vigencia la contratación de los servicios de la nube de Amazon Web Services (AWS) en modalidad autogestionada, para que soporten los sistemas de información académico, administrativo, financiero y de investigación como también contribuir con la ejecución de las actividades enmarcadas en los planes, programas y proyectos del plan de desarrollo vigente en la Universidad Distrital Francisco José de Caldas.

2.2. Específicos

- Adquirir Servicios de Computación en la Nube en la modalidad de la bolsa de recursos de acuerdo a la reglamentación provista por Colombia Compra Eficiente.
- Contratar los servicios de la nube de tal forma que la Universidad pueda autogestionar los recursos contratados sin intermediación del proveedor utilizando principios de diseño que incluyan excelencia operativa, seguridad, fiabilidad, eficacia de rendimiento y optimización de costos.
- Realizar un único pago inicial al proveedor por la contratación de los servicios de la nube adquiridos a través de Colombia Compra Eficiente para reducir la carga administrativa que se genera al hacer pagos mensuales, para flexibilizar el uso de los recursos de la nube y optimizar costos.

3. Puntos de contacto (Entidad)

Nombre de la entidad:	UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
Representante Legal Rector	Tarazona Bermúdez Giovanny Mauricio
NIT	899.999.230.7
Sede principal	Bogotá - Carrera 7 # 40B - 53
Página Web	www.udistrital.edu.co/inicio
Dependencia encargada	Oficina Asesora de Sistemas Plan Estratégico de Tecnologías de Información y Comunicaciones - Planestic-UD
Jefe Oficina Asesora de Sistemas	Alejandro Paolo Daza Corredor José Ignacio Palacios Osma
Correo dependencia	computo@udistrital.edu.co Planesticud@udistrital.edu.co
Teléfonos	323 9300 Ext: 1109, 1112, 1113 Planestic-UD 6338
Página Web	ti.udistrital.edu.co

4. Tipificación de la necesidad

La Entidad cuenta con diversas soluciones desplegadas en la nube de AWS y en particular las cargas de trabajo a las que se refiere este RFI han sido desplegadas y optimizadas en la nube de AWS desde hace 9 años. Hasta la fecha la tecnología ofrecida por AWS ha llenado los requisitos funcionales y no funcionales de las soluciones objeto de este RFI respondiendo adecuadamente a las expectativas de desempeño de la Entidad. Teniendo en cuenta lo anterior y el principio de uso eficiente de los recursos públicos la Entidad ha decido no incurrir en los costos adicionales que significa un proceso de migración o integración con otra tecnología diferente a la ofrecida por AWS. Es decir, este RFI corresponde a una renovación del sistema cloud actual (AWS) bajo la modalidad auto-gestionada, lo que consiste en que la administración, gestión y control de la cuenta sea administrada por las oficinas originadoras de la necesidad.

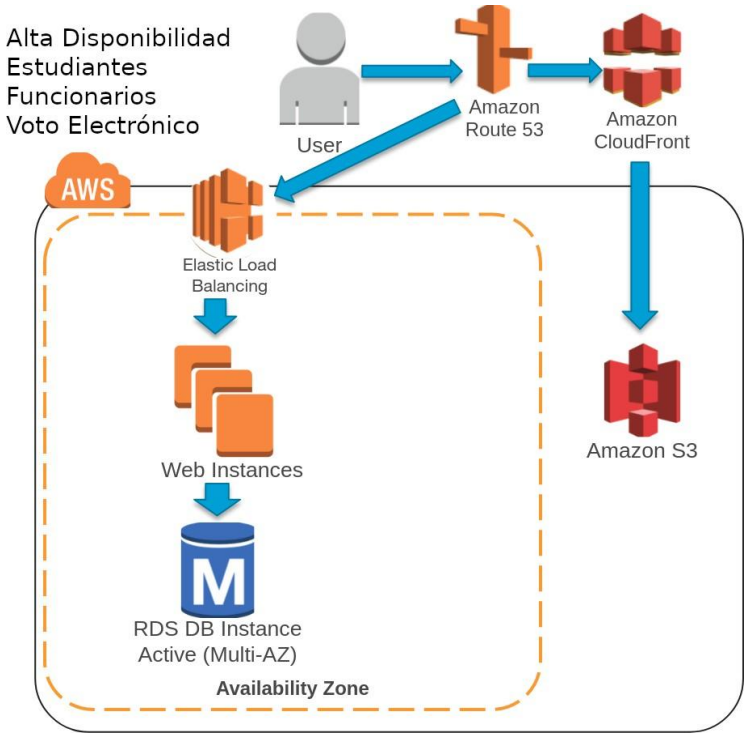
4.1. Descripción de la carga de trabajo.

La infraestructura en la nube implementada por la Universidad Distrital tiene tres implementaciones:

4.1.1. Arquitectura con alta disponibilidad sobre EC2 (Elastic Compute Cloud)

Están desplegados de forma independiente para cuatro servicios: Atención a estudiantes, Atención a funcionarios, atención al proceso de Voto electrónico y aulas virtuales.

Ilustración 1 Arquitectura de referencia



4.1.1.1. Atención a estudiantes

Para el servicio de atención a los estudiantes se mantienen en los tiempos valle 2 instancias web del tipo m5.large las cuales se escalan hasta 45 instancias por unas 20 horas cada semestre. Incluye ambiente de pruebas con 1 instancia t3.micro en funcionamiento la mayor parte del año. En cuanto a RDS, se cuenta con:

- 1 Instancia Oracle Standard Edition Two de producción db.r5.xlarge con licencia incluida en tiempos valle, para eventos como cierre e inicio de semestre se escala a db.r5.4xlarge, por un periodo total de aproximadamente 2 semanas; Se cuenta con 1 instancia para pruebas db.t3.micro Oracle Standard Edition Two con licencia incluida para todo el año.
- 1 instancia Aurora MySQL de producción db.r5.xlarge en tiempos valle, para eventos de cierre e inicio de semestre se escala entre db.r5.2xlarge a db.r5.12xlarge con un nodo adicional db.t3.micro, dependiendo de la demanda por aproximadamente 2 semanas; Se cuenta con 1 instancia para pruebas db.t3.micro para todo el año.

4.1.1.2. Atención a funcionarios

Para el servicio a los funcionarios se mantienen en los tiempos valle 2 instancias web del tipo m5.large las cuales se escalan hasta 4 instancias por unas 72 horas cada semestre. Incluye ambiente de pruebas con 1 instancia t3.micro. Incluye ambiente de pruebas con 1 instancia t3.micro en funcionamiento la mayor parte del año. En cuanto a RDS, se cuenta con:

- 1 Instancia Oracle Standard Edition Two de producción db.r5.large con licencia incluida para todo el año, hasta el momento no ha sido necesario escalar ; Se cuenta con 1 instancia para pruebas db.t3.micro Oracle Standard Edition Two con licencia incluida esporádicamente de acuerdo a solicitud de desarrollos nuevos que lo requieran.
- 1 Instancia MySQL Community de producción db.m4.large para todo el año, hasta el momento no ha sido necesario escalar; Se cuenta con 1 instancia para pruebas db.t3.micro para todo el año.
- 1 instancia PostgreSQL de producción db.m4.xlarge para todo el año, hasta el momento no ha sido necesario escalar; Se cuenta con 1 instancia para pruebas db.t3.micro para todo el año.

4.1.1.3. Servicio de voto electrónico

El servicio de voto electrónico utiliza instancias del tipo m4.large y se despliega durante 15 días, solamente cuando son programados procesos de elecciones en la Universidad, es decir dos o tres veces al año. Incluye ambiente de pruebas con 1 instancia t2.micro. Incluye ambiente de pruebas con 1 instancia t3.micro en funcionamiento la mayor parte del año.

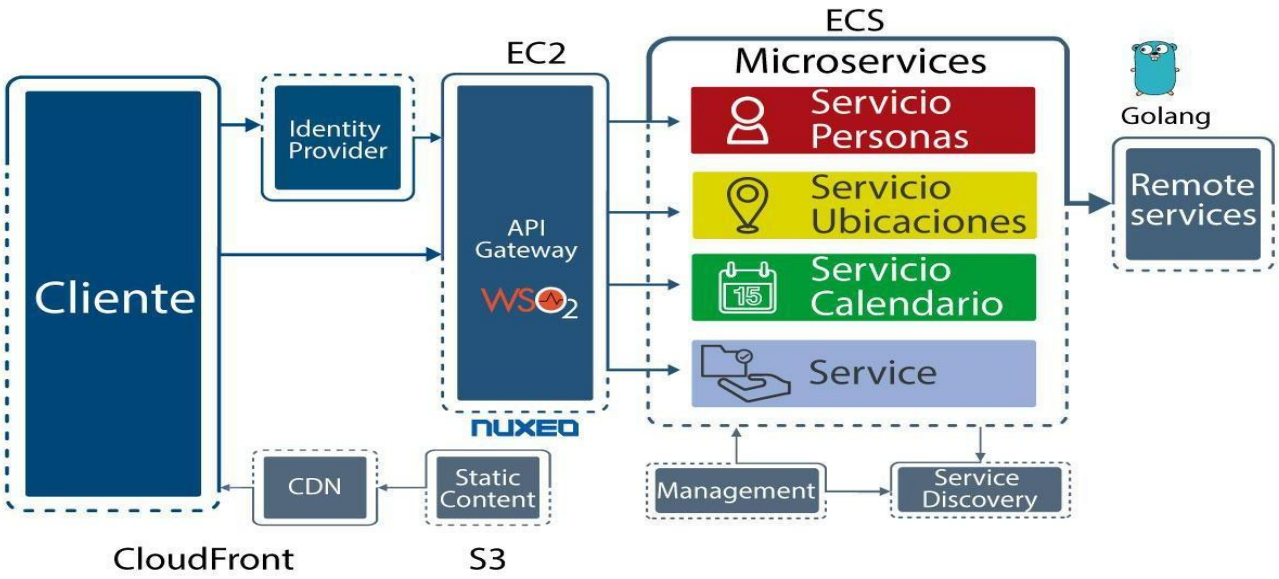
- 1 instancia Aurora MySQL de producción db.t3.small en un periodo aproximado de 2 meses por evento, para evento del dia de votación se escala a db.r5.4xlarge con un nodo adicional db.t3.micro; Se cuenta con 1 instancia para pruebas db.t3.micro para un periodo aproximado de 2 meses por evento.

4.1.1.4. Servicios de Aulas Virtuales

El servicio de aulas virtuales incluye las plataformas de Moodle y Edx desplegadas utilizando arquitecturas elásticas y con alta disponibilidad, durante la vigencia la universidad podrá mejorar la arquitectura en aras de reducir costos de consumo y ajustarse a las cargas cambiantes.

4.1.2. Arquitectura con alta disponibilidad basada en microservicios sobre ECS (Elastic Container service)

Ilustración 2 Arquitectura basada en microservicios



Los microservicios están en ECS que son soportados por dos clusters escalables cada uno 4 instancias EC2 m5.large. Aunque tiene implementado auto-escalamiento la carga de los servicios no lo ha requerido. El servidor de WSO2 es del tipo m5.xlarge no ha requerido escalamiento.

En cuanto al área de Base de datos se cuenta con:

- 1 instancia RDS PostgreSQL de producción db.t3.medium WSO2, para todo el año, hasta el momento no ha sido necesario escalar.

1 instancia EC2 de producción db.t3.small Mongo, para todo el año, hasta el momento no ha sido necesario escalar ; Se cuenta con 1 instancia para pruebas db.t2.micro para todo el año.

4.1.3. Servidores EC2 sin alta disponibilidad (no requerida)

Instancias de servidores para aplicaciones de uso interno o de poca demanda que no requieren auto-escalamiento.

- 10 Servidores m5.large
- 5 servidores t3.micro

4.2. Funcionalidades requeridas en cada servicio

4.2.1. Servicio de cómputo (Amazon EC2)

- El servicio debe contar con un auténtico entorno virtual de cómputo que permita utilizar interfaces de servicios web para lanzar instancias con distintos sistemas operativos, cargarlas con su entorno de aplicaciones personalizado, administrar los permisos de acceso a la red y ejecutar su imagen utilizando los sistemas que desee.
- El servicio debe permitir pausar y reanudar las instancias.
- El servicio debe contar con instancias de E/S de alto desempeño.
- El servicio debe contar con instancias de almacenamiento HDD denso.
- El servicio debe permitir configuraciones de CPU optimizadas
- El servicio debe contar con opciones de almacenamiento flexibles
- El servicio debe contar con la capacidad para lanzar / administrar un grupo de recursos de cómputo con una sola solicitud
- El servicio debe contar con la capacidad para desactivar Intel Hyper-Threading (HT) para cargas de trabajo de un solo subproceso
- El servicio debe permitir especificar en qué host dedicado se ejecutará un recurso de cómputo después de que se haya detenido y reiniciado.
- El servicio debe permitir hacer seguimiento de licencias para regular el uso y el cumplimiento
- El servicio debe permitir implementar funcionalidades de auto-escalamiento basadas en ML.
- El servicio debe contar con la capacidad de sincronización de tiempo para instancias cómputo.
- El servicio debe ser suministrado bajo un esquema de pago por uso
- El servicio debe ofrecer la posibilidad de colocar instancias en distintas regiones y zonas de disponibilidad
- El servicio debe permitir el uso de direcciones IP elásticas
- El servicio debe permitir ajustar la escala de la capacidad de las instancias automáticamente de acuerdo con las condiciones que se definan
- El servicio debe permitir el uso de redes mejoradas (Enhanced Networking) para lograr un desempeño de paquetes por segundo (PPS) significativamente superior, una reducción del ruido de red y latencias menores.

- El servicio debe permitir acceder de manera privada a la API de las instancias desde su red privada de nube o sobre conexión directa, sin utilizar IP públicas y sin que el tráfico deba atravesar la Internet.
- Debe ofrecer un servicio de origen de hora de alta precisión, fiabilidad y disponibilidad que pueda ser usado por los servicios de cómputo.
- El servicio debe contar con una disponibilidad mínima de 99,99%
- El servicio debe incluir sistemas de almacenamiento por bloques EBS y EFS

4.2.2. Servicio de escalado de aplicaciones para optimizar los costos y el nivel de desempeño (AWS Autoscaling)

- El servicio debe permitir lanzar y finalizar recursos de cómputo.
- El servicio debe poder escanear el entorno y descubrir automáticamente los recursos escalables en la nube subyacentes a la aplicación, por lo que no se requiere identificar manualmente estos recursos uno por uno a través de interfaces de servicio individuales.
- El servicio debe permitir seleccionar una de las tres estrategias de optimización: optimización del rendimiento, optimización de los costos o equilibrar las dos variables de optimización.
- El servicio debe permitir implementar estrategias de escalamiento predictivo a partir del tráfico futuro, incluidos los picos que ocurren regularmente, y aprovisiona el número correcto de recursos de cómputo antes de los cambios previstos. Utilizando algoritmos de aprendizaje automático
- El servicio debe permitir crear automáticamente políticas de escalado de seguimiento de destino para todos los recursos del plan de escalado
- El servicio debe poder calcular continuamente los ajustes de escala apropiados e inmediatamente debe agregar y eliminar capacidad según sea necesario para mantener las métricas en el objetivo.
- El servicio debe permitir que las políticas de escalado de seguimiento de objetivos se optimicen automáticamente y aprendan sus patrones de carga reales para minimizar las fluctuaciones en la capacidad de los recursos.
- El servicio debe tener la capacidad de mezclar diferentes tipos de recursos de cómputo en un grupo de escalado
- El servicio debe tener la capacidad de realizar acciones personalizadas en recursos de cómputo que se inician / finalizan
- El servicio debe ofrecer la posibilidad de personalizar las políticas de escalado.
- El servicio debe permitir habilitar y deshabilitar las políticas de escalado
- El servicio debe contar con notificaciones que proporcionen una guía de concientización y corrección para problemas de disponibilidad o rendimiento de recursos que pueden afectar las aplicaciones.
- El servicio debe permitir marcar un recurso de cómputo en mal estado para programar su reemplazo
- El servicio debe permitir utilizar AMI (o imágenes de la galería) para aprovisionar instancias / máquinas virtuales en un grupo de escalado.

4.2.3. Servicios de redes (Amazon VPC)

- El servicio debe ser escalable y debe permitir especificar un rango de direcciones IP privadas de que sean elegidas.
- El servicio debe permitir ampliar la nube privada virtual mediante la incorporación de intervalos IP secundarios.

- El servicio debe permitir dividir el rango privado de direcciones IP privadas de la nube privada virtual en una o varias subredes públicas o privadas para posibilitar la ejecución de aplicaciones y la prestación de servicios en la nube privada virtual.
- El servicio debe permitir controlar el acceso de entrada y salida desde y hacia subredes individuales por medio de listas de control de acceso.
- El servicio debe permitir almacenar datos y definir permisos de forma que el acceso a los datos sea posible exclusivamente desde el interior de la nube privada virtual.
- El servicio debe permitir asignar varias direcciones IP y asociar múltiples interfaces de red elásticas a instancias de la nube privada virtual.
- El servicio debe permitir asociar una o más direcciones IP elásticas a cualquier instancia de la nube privada virtual, de modo que puedan alcanzarse directamente desde Internet.
- El servicio debe permitir conectarse a la nube privada virtual con otras nubes privadas virtuales y obtener acceso a los recursos de otras nubes privadas virtuales a través de direcciones IP privadas mediante la interconexión de nube privada virtual.
- El servicio debe permitir conectarse de manera privada a los servicios del PSN sin usar una gateway de Internet, ni una NAT ni un proxy de firewall mediante un punto de enlace de la nube privada virtual.
- El servicio debe permitir conectarse de manera privada a sus propios servicios o soluciones de SaaS con tecnología de PrivateLink.
- El servicio debe permitir conectar la nube privada virtual y la infraestructura de TI local con la VPN del PSN de sitio a sitio.
- El servicio debe permitir asociar grupos de seguridad de la nube privada virtual con instancias en la plataforma.
- El servicio debe permitir registrar información sobre el tráfico de red que entra y sale de las interfaces de red de la nube privada virtual.
- El servicio debe permitir habilitar IPv4 e IPv6 en la nube privada virtual.
- El servicio debe permitir usar la replicación de tráfico de nube privada virtual a fin de capturar y replicar el tráfico de la red para las instancias.
- El servicio debe permitir interceptar y analizar el tráfico de entrada y salida, mediante un dispositivo de red y seguridad, incluidas las ofertas de terceros.
- El servicio debe tener la habilidad de mover direcciones entre instancias
- El servicio debe tener la capacidad de usar nuestra propia dirección IP pública dentro de la red virtual
- El servicio debe tener la capacidad de reflejar el tráfico y transmitirlo a un recopilador de paquetes de red.
- El servicio debe tener la capacidad de análisis para monitoreo de trafico de red.
- El servicio debe tener la capacidad de mover interfaces de red entre instancias
- El servicio debe permitir implementar conectividad de tránsito (modelo Hub-and-Spoke)
- El servicio debe permitir compartir una red virtual entre diferentes cuentas (modelo compartido)
- El servicio debe ofrecer resolución de DNS para entornos híbridos
- El servicio debe ofrecer resolución de DNS a nombres de host privados
- El servicio debe ofrecer resolución de DNS a nombres de host públicos
- El servicio debe contar con métricas de rendimiento de la red de instancias
- NIC: el servicio debe contar con la capacidad para configurar comprobaciones de origen / destino en interfaces de red
- El servicio debe permitir el cifrado de tráfico WAN (entre los centros de datos)

4.2.4. Servicio de Gestión de identidad y acceso (AWS Identity and Access Management (IAM))

- El servicio debe permitir controlar el acceso y los permisos a sus recursos y servicios de la nube
- El servicio debe permitir que se administren permisos para sus usuarios y aplicaciones
- El servicio debe permitir usar identidad federada para administrar accesos a una cuenta
- El servicio debe permitir analizar el acceso a recursos y servicios.
- El servicio debe garantizar que los usuarios no tendrán acceso a los recursos de la nube hasta que se concedan de forma explícita los permisos.
- El servicio debe permitir crear credenciales temporales
- El servicio debe permitir identificar recursos que puedan accederse desde fuera de la cuenta
- El servicio debe permitir identificar y eliminar fácilmente los permisos no utilizados
- El servicio debe permitir diferentes modos de autenticación de usuarios como contraseñas, pares de claves y autenticación multifactor
- El servicio debe soportar la federación desde sistemas corporativos como Microsoft Active Directory, así como proveedores de identidad basados en estándares.
- El servicio debe permitir bloquear el acceso a los puertos y generar listas blancas de direcciones IP a través políticas
- El servicio debe permitir identificar cuándo se utilizó por última vez una clave de acceso para rotar claves antiguas
- El servicio debe contar con un simulador de políticas para probar el funcionamiento de las políticas antes de usarlas en producción
- El servicio debe permitir validar las políticas para garantizar que sus políticas coinciden con la intención que se tiene en su definición
- El servicio debe permitir establecer credenciales de seguridad temporales al realizar solicitudes entre servicios

4.2.5. Almacenamiento de objetos (AWS S3)

- El servicio debe permitir que un solo objeto pueda tener un tamaño de hasta 5 terabytes
- El servicio debe contar con capacidades para anexar etiquetas de metadatos a los objetos, mover y almacenar datos entre los tipos de almacenamiento, configurar y aplicar controles de acceso a datos, proteger los datos frente a usuarios no autorizados, ejecutar análisis de big data y monitorear los datos en los niveles de objeto y carpetas que contienen objetos.
- El servicio debe permitir el acceso a los objetos a través de los puntos de acceso del servicio o directamente a través del nombre de host de la carpeta que los almacena.
- El servicio debe permitir anexar a cada objeto hasta 10 pares de clave-valor denominados etiquetas de objetos, que se pueden crear, actualizar y eliminar a lo largo de todo el ciclo de vida de los objetos.
- El servicio debe permitir utilizar un informe de inventario, donde se enumeran los objetos almacenados en una carpeta de objetos o con un prefijo específico, así como sus metadatos y estado de cifrado correspondientes.
- El servicio debe permitir copiar objetos entre carpetas, reemplazar conjuntos de etiquetas de objetos, modificar los controles de acceso y restaurar objetos archivados desde otros servicios de almacenamiento.

- El servicio debe admitir características que ayudan a mantener el control de versiones de los datos, impedir el borrado accidental y replicar datos en diversas ubicaciones del CSP.
- El servicio debe contar con control de versiones que permitan preservar, recuperar y restaurar fácilmente todas las versiones de un objeto almacenado, lo que debe permitir recuperarse fácilmente de acciones de usuarios involuntarias y de errores de aplicaciones.
- El servicio debe impedir el borrado accidental al contar con funcionalidades de eliminación Multi-Factor Authentication (MFA).
- El servicio debe permitir replicar objetos (así como sus metadatos y etiquetas de objeto respectivos) en otras regiones del CSP o en la misma ubicación para lograr una latencia reducida, conformidad, seguridad y recuperación de desastres.
- El servicio debe permitir aplicar políticas de escritura única y lectura múltiple (WORM)
- El servicio debe permitir aplicar etiquetas a las carpetas para asignar costos en múltiples dimensiones de negocio (por ejemplo, centros de costos, nombres de aplicación o propietarios) y, después, debe permitir utilizar los informes de asignación de costos para ver el uso y los costos que agregan las etiquetas de las carpetas.
- El servicio debe permitir hacer el seguimiento de las actividades de nivel de carpetas y objetos e informar sobre ellas.
- El servicio debe permitir configurar las notificaciones de eventos para activar flujos de trabajo y alertas.
- El servicio debe permitir crear usuarios y administrar su correspondiente acceso.
- El servicio debe conceder acceso a objetos individuales a los usuarios autorizados
- El servicio debe permitir configurar permisos para todos los objetos de una única carpeta.
- El servicio debe permitir simplificar la administración del acceso de datos a conjuntos de datos compartidos creando puntos de acceso con nombres y permisos específicos para cada aplicación o conjuntos de aplicaciones
- El servicio debe conceder acceso a otros usuarios por tiempo limitado con direcciones URL temporales.
- El servicio debe permitir enumerar las solicitudes realizadas a sus recursos para obtener total visibilidad de quién obtiene acceso a los distintos datos.
- El servicio debe ofrecer características de seguridad flexibles para bloquear el acceso de usuarios no autorizados a sus datos
- El servicio debe admitir el cifrado tanto del lado de servidor (con tres opciones de administración clave) como del lado de cliente para cargas de datos.
- El servicio debe permitir comprobar el estado de cifrado de los objetos
- El servicio debe contar con controles de seguridad que garantizan que las carpetas y objetos no tengan acceso público
- El servicio debe contar con clases de almacenamiento: clase / nivel de almacenamiento de movimiento de datos automático basado en patrones de acceso
- El servicio debe contar con la siguiente funcionalidad para protección de datos: sincronización de replicación bidireccional
- El servicio debe contar con la siguiente funcionalidad para protección de datos: cumplimiento de bloqueo WORM a nivel de carpeta/contenedor de objetos
- El servicio debe contar con la siguiente funcionalidad para protección de datos: Control de tiempo de replicación con ANS
- El servicio debe permitir ejecutar operaciones por lotes en etiquetas, incluidas eliminaciones
- El servicio debe permitir eliminar de varios objetos mediante una única llamada a la API
- El servicio debe permitir agregar etiquetas a los objetos
- El servicio debe permitir definir puntos de acceso para puntos de entrada seguros a datos compartidos
- El servicio debe contar con una función / herramienta / servicio para analizador el acceso
- El servicio debe permitir bloquear el acceso público a nivel de cuenta / suscripción
- El servicio debe permitir la auditoría continua de políticas de acceso y configuración de seguridad

4.2.6. Servicio de DNS (Route 53)

- El servicio debe ser escalable y debe proveer alta disponibilidad
- El servicio debe permitir crear reglas de reenvío condicional y puntos de enlace DNS para resolver nombres personalizados controlados en las zonas privadas alojadas en el servicio o en los servidores DNS que se encuentran en las instalaciones.
- El servicio debe permitir redirigir a los usuarios finales hacia los mejores puntos de enlace para la aplicación en función de la geo-proximidad, la latencia, el estado y otras consideraciones
- El servicio debe permitir remitir a los usuarios finales a un punto de enlace determinado que la Entidad especifique en función de la ubicación geográfica del usuario final.
- El servicio debe permitir administrar nombres de dominio personalizados para los recursos de la nube internos sin exponer datos de DNS en la web pública.
- El servicio debe permitir dirigir automáticamente a los visitantes del sitio web a una ubicación alternativa para evitar interrupciones del servicio.
- El servicio debe permitir dirigir automáticamente a los visitantes del sitio web a una ubicación alternativa para evitar interrupciones del servicio.
- El servicio debe ofrecer servicios de registro de nombres de dominio, donde sea posible buscar y registrar nombres de dominio disponibles o transferir nombres de dominio existentes para que se administren a través del servicio.
- El servicio debe contar con una sencilla interfaz de servicios web que permita ponerse en marcha en cuestión de minutos
- El servicio debe permitir transferir el dominio desde otro servicio DNS al servicio DNS en la nube
- El servicio debe ofrecer un conjunto sencillo de API que facilita la creación y la administración de registros DNS para los dominios
- El servicio debe incluir la funcionalidad de administración de nombres DNS para escalar hacia arriba o hacia abajo el microservicio.
- El servicio debe tener una disponibilidad del 100%

4.2.7. Red de Distribución de Contenido (AWS Cloudfront)

- El servicio debe permitir distribuir a clientes globalmente datos, vídeos, aplicaciones y API de forma segura, con baja latencia, altas velocidades de transferencia y dentro de un entorno fácil para desarrolladores
- El servicio permite entregar su contenido, sus API o sus aplicaciones a través de SSL/TLS y las características avanzadas de SSL se deben poder activar automáticamente
- El servicio debe permitir implementar la protección frente a ataques a la red y la capa de aplicación al integrarse con otros servicios
- El servicio debe soportar cifrados de SSL/TLS y HTTPS
- El servicio debe permitir utilizar SSD cifrados para ubicaciones de borde y volúmenes de almacenamiento elástico de bloques cifrados para cachés de borde regionales.
- El servicio debe permitir cifrar los datos en tránsito.

- El servicio debe soportar encriptación a nivel de campos.
- El servicio debe restringir el acceso a su contenido con una serie de funciones
- El servicio debe permitir la autenticación de tokens para restringir el acceso solo a los espectadores autenticados
- El servicio debe evitar que usuarios de ubicaciones geográficas específicas obtengan acceso a contenido
- El servicio debe permitir configurar varios orígenes para habilitar la redundancia en su arquitectura de backend
- El servicio debe permitir medir continuamente la conectividad a Internet, el rendimiento y la computación para encontrar la mejor manera de direccionar las solicitudes a nuestra red, teniendo en cuenta el rendimiento, la carga, el estado operativo y otros factores para ofrecer la mejor experiencia en tiempo real
- El servicio debe permitir la transmisión eficiente de solicitudes entre las ubicaciones
- El servicio debe permitir acelerar tanto el contenido estático como el dinámico para mejorar el rendimiento de los usuarios
- El servicio debe proporcionar una gran flexibilidad para optimizar el comportamiento de la caché, junto con optimizaciones de la capa de red para la latencia y el rendimiento
- El servicio debe admitir el protocolo WebSocket, así como el protocolo HTTP con los siguientes métodos HTTP: GET, HEAD, POST, PUT, DELETE, OPTIONS y PATCH
- El servicio debe permitir mejorar el rendimiento de los sitios web dinámicos que incluyen formularios web, espacio para comentarios, cuadros de inicio de sesión, botones "añadir a la cesta", aplicaciones basadas en WebSocket y otras características que cargan datos procedentes de los usuarios finales
- El servicio debe permitir utilizar un único nombre de dominio para la entrega de todo el sitio web y así acelerar tanto la descarga como la carga de partes de su sitio web.
- El servicio debe permitir técnicas como el almacenamiento en caché por capas y la optimización de la duplicación de objetos en caché para ayudar a maximizar la retención de caché.
- El servicio debe contar con registros de actividades en tiempo real.
- El servicio debe permitir crear / renovar certificados públicos / privados de forma gratuita.

4.2.8. Balanceador de carga (Elastic Load Balancing)

- El servicio debe distribuir automáticamente el tráfico de aplicaciones entrantes a través de varios destinos, tales como instancias y direcciones IP.
- El servicio debe estar en capacidad de detectar destinos que funcionen incorrectamente, dejar de enviar tráfico a ellos y, a continuación, distribuir la carga entre los destinos restantes que no presenten problemas.
- Se deben poder crear y administrar grupos de seguridad asociados con balanceadores de carga a fin de ofrecer opciones de seguridad y redes adicionales
- El servicio debe proporcionar la capacidad de administración integrada de certificados y descifrado SSL/TLS, lo que debe brindar la flexibilidad para administrar de manera centralizada los parámetros de SSL del balanceador de carga y eliminar el trabajo intensivo de la CPU de la aplicación.
- El servicio debe permitir equilibrar cargas de trabajo a nivel de capa 4 y capa 7.
- El servicio debe permitir equilibrar la carga en aplicaciones HTTP o HTTPS para características específicas de la capa 7.
- El servicio debe facilitar el monitoreo de rendimiento de las aplicaciones en tiempo real.

- El servicio debe proporcionar direccionamiento de solicitudes avanzado destinado a la entrega de arquitecturas de aplicaciones modernas, incluidos micro servicios y aplicaciones basadas en contenedores
- El servicio debe asegurar que se utilicen en todo momento los protocolos y cifradores SSL/TLS más recientes.
- El servicio debe poder presentar varios certificados mediante el mismo agente de escucha seguro, lo que le permite admitir varios sitios web seguros a través del uso de un único agente de escucha seguro
- El servicio debe ser compatible con el algoritmo de selección de certificados inteligentes con SIN (Indicación de nombre de servidor)
- Cuando el nombre de host indicado por un cliente coincide con varios certificados, el servicio debe estar en capacidad de establecer cuál es el certificado más adecuado en función de varios factores, entre ellos, las capacidades del cliente.
- El servicio debe permitir equilibrar cargas a un backend de aplicación alojado en cualquier dirección IP y con cualquier interfaz de una instancia
- El servicio debe permitir usar direcciones IP como destinos para equilibrar cargas de aplicaciones alojadas en ubicaciones locales.
- El servicio debe permitir distribuir el tráfico de entrada entre destinos en numerosas zonas de disponibilidad
- El servicio debe escalar automáticamente la capacidad de administración de solicitudes como respuesta al tráfico de aplicaciones entrante
- El servicio debe poder ser configurado para que se pueda obtener acceso a él desde Internet o crear un balanceador de carga sin direcciones IP públicas para que actúe como balanceador de carga interno (es decir, sin acceso a Internet)
- Si la aplicación se compone de varios servicios individuales, el balanceador de carga de aplicaciones debe poder direccionar una solicitud a un servicio en función del contenido de la solicitud
- El servicio debe soportar: direccionamiento basado en host, direccionamiento basado en ruta, direccionamiento basado en el encabezado HTTP, direccionamiento basado en el método HTTP, direccionamiento basado en parámetros de cadenas de consultas y direccionamiento basado en CIDR para direcciones IP de origen.
- El servicio debe poder direccionar una solicitud de cliente basada en el CIDR para direcciones IP de origen desde donde se origina la solicitud.
- El servicio debe ser compatible con HTTP/2
- El servicio debe ser compatible con WebSockets
- El servicio debe contar con compatibilidad IPv6 nativa
- El servicio debe ser compatible con las sesiones persistentes mediante el uso de cookies generadas por el balanceador de carga.
- El servicio debe direccionar el tráfico solamente a destinos que funcionan correctamente.
- El servicio debe facilitar el monitoreo de métricas tales como el recuento de solicitudes, el recuento de errores, los tipos de errores y la latencia de las solicitudes.
- El servicio debe permitir registrar todas las solicitudes enviadas al balanceador de carga.
- El servicio debe permitir monitorear una solicitud por un ID único a medida que esta se desplaza por los diversos servicios que componen sus sitios web y aplicaciones distribuidas.
- El servicio debe ser compatible con un algoritmo de equilibrio de cargas de turno rotativo.
- El servicio debe ser compatible con un modo de inicio lento con el algoritmo de turno rotativo que le permite añadir nuevos objetivos sin sobrecargarlos con un aluvión de solicitudes.

- El servicio debe permitir autenticar a los usuarios de manera segura a medida que obtengan acceso a las aplicaciones de la nube.
- El servicio debe permitir a los usuarios finales realizar autenticaciones mediante proveedores de identidades de redes sociales y mediante proveedores de identidades empresariales, como Microsoft Active Directory a través de SAML o cualquier proveedor de identidades que cumpla con OpenID Connect (IdP).
- El servicio debe permitir redirigir una solicitud entrante de una URL a otra distinta.
- El servicio debe tener la capacidad de redirigir las solicitudes de HTTP a las solicitudes de HTTPS.
- El servicio debe permitir usar los direccionamientos para enviar a los usuarios a diferentes sitios web, por ejemplo, al redirigir desde una versión antigua de una aplicación hacia una nueva versión.
- El servicio debe contar con la siguiente funcionalidad de compatibilidad: TCP + UDP en el mismo oyente / puerto para casos de uso de DNS.
- El servicio debe contar con capacidades de protección frente a borrado
- El servicio debe contar con la siguiente funcionalidad para enrutamiento: soporte nativo para respuesta fija
- El servicio debe soportar la siguiente funcionalidad para enrutamiento: admite cualquier enrutamiento basado en encabezado, URL e IP de origen.
- El servicio debe contar con backends sin servidor

4.2.9. Servicio de base de datos relacional (Amazon RDS)

- El servicio debe permitir automatizar las tareas administrativas, como el aprovisionamiento de hardware, la configuración de bases de datos, la implementación de parches y la creación de copias de seguridad.
- El servicio debe ofrecer varios tipos de recursos de cómputo: optimizados para memoria, rendimiento u operaciones de E/S
- El servicio debe permitir escoger entre los siguientes motores de bases de datos Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database y SQL Server.
- El servicio debe ser compatible con herramientas para migrar o replicar las bases de datos existentes.
- El servicio debe estar en capacidad de encargarse de tareas habituales de las bases de datos, como el aprovisionamiento, las revisiones, las copias de seguridad, la recuperación, la detección de errores y la reparación.
- El servicio se debe poder desplegar en múltiples ubicaciones.
- El servicio debe permitir aplicar de forma automática parches de software.
- El servicio debe contar con la opción de controlar si se deben aplicar parches a un recurso de cómputo de base de datos o no, y el momento en que se deben aplicar.
- El servicio debe ofrecer orientación sobre prácticas recomendadas mediante el análisis de las métricas de configuración y uso de los recursos de cómputo de bases de datos.
- El servicio debe ofrecer sugerencias sobre versiones de motores de base de datos, almacenamiento, tipos de recursos de cómputo y redes.
- El servicio debe permitir analizar las sugerencias disponibles y realizar una acción sugerida de inmediato, programarla para su próximo periodo de mantenimiento o descartarla por completo.
- El servicio debe contar con diversas opciones de almacenamiento en virtud del rendimiento requerido. Las opciones de almacenamiento deben incluir: Almacenamiento de uso general (SSD) y Almacenamiento de IOPS provisionadas (SSD).

- El servicio debe permitir escalar los recursos informáticos y de memoria para ampliar o reducir la implementación, hasta un máximo de 32 vCPU y 244 GiB de RAM.
- El servicio debe permitir aprovisionar almacenamiento adicional.
- El servicio debe permitir ampliar automáticamente el tamaño del volumen de la base de datos a medida que las necesidades de almacenamiento de la base de datos crecen, hasta un máximo de 64 TB o la cantidad máxima que establezca.
- El servicio debe permitir hacer réplicas de lectura.
- El servicio debe permitir crear una o varias réplicas de un recurso de cómputo de base de datos de origen determinada y abastecer el alto volumen de tráfico de lectura de la aplicación desde distintas copias de sus datos, lo cual aumenta el rendimiento de lectura total.
- El servicio debe permitir hacer copias de seguridad automatizadas.
- El servicio debe permitir realizar una copia de seguridad de los registros de base de datos y de transacciones y los debe poder almacenar durante un periodo de retención que puede especificar el usuario.
- El servicio debe permitir especificar el periodo de retención de copia de seguridad automática hasta un máximo de 35 días.
- El servicio debe permitir crear instantáneas de base de datos (copias de seguridad) que inicia el usuario de la instancia almacenada en el servicio de almacenamiento de objetos, y que se conservarán hasta que se eliminen explícitamente.
- El servicio debe permitir cifrar las bases de datos mediante las claves.
- El servicio debe permitir que los datos almacenados en reposo en el almacenamiento subyacente estén cifrados, al igual que las copias de seguridad automatizadas, las réplicas de lectura y las instantáneas.
- El servicio debe soportar la capacidad de aislar la base de datos en la propia red virtual y conectarse a su infraestructura de TI local mediante las VPN con IPsec cifradas estándar del sector.
- El servicio debe ofrecer la posibilidad de controlar las acciones que los usuarios y grupos.
- El servicio debe permitir controlar las acciones que pueden realizar los usuarios y grupos en grupos de recursos que tengan la misma etiqueta y valor asociado
- El servicio debe soportar herramientas de monitoreo que permitan monitorear métricas operativas clave, incluidos el uso de la capacidad de cómputo, memoria y almacenamiento, la actividad de E/S y las conexiones de instancias de bases de datos.
- El servicio debe soportar la capacidad de notificar eventos de la base de datos por email o SMS
- El servicio debe soportar el registro y auditoría de los cambios en la configuración de la instancia de base de datos, incluidos grupos de parámetros, grupos de subred, instantáneas, grupos de seguridad y suscripciones a eventos.
- El servicio debe soportar cifrado en reposo con claves administradas por la Entidad
- El servicio debe contar con capacidad de conmutación por error (automatizada)
- El servicio debe contar con capacidad de prueba de conmutación por error (manual)
- El servicio debe soportar leer réplicas, replicación retrasada (para configuraciones de recuperación ante desastres)
- El servicio debe soportar escalamiento horizontal

4.2.10. Servicio de gestión de contenedores (Amazon Elastic Container Service)

- El servicio debe permitir la orquestación de contenedores de forma completamente administrada.
- El servicio debe ser compatible con Docker y le permite ejecutar y administrar contenedores Docker
- El servicio debe permitir ejecutar, ajustar la escala y proteger aplicaciones en contenedores Docker
- Las aplicaciones empaquetadas como contenedores a nivel local se deben poder implementar y ejecutar de la misma forma que los contenedores administrados por el servicio
- El servicio debe estar en capacidad de ajustar la infraestructura de administración de clústeres y organización de contenedores
- El servicio debe permitir el escalamiento automático de los recursos
- El servicio debe permitir pasar de un solo contenedor a miles de ellos en cientos de recursos de cómputo sin incrementar el nivel de complejidad de la ejecución de la aplicación
- El servicio debe permitir ejecutar cualquier elemento: aplicaciones, trabajos por lotes o micro-servicios.
- El servicio debe permitir optar por tener una visibilidad y control completos del clúster de servidor subyacente, desde la creación y finalización de contenedores Docker a la visualización de información detallada acerca del estado de los clústeres.
- El servicio debe ser compatible con la administración de contenedores de Windows.
- El servicio debe contar con herramientas de código abierto que permitan definir y ejecutar aplicaciones multi-contenedor
- El servicio debe permitir usar cualquier repositorio de imágenes de Docker alojado por terceros o registro de Docker con acceso privado, como Docker Hub.
- El servicio debe permitir definir tareas a través de una plantilla JSON
- El servicio debe permitir acciones de API sencilla para crear y eliminar clústeres, registrar tareas y anular su registro, lanzar y finalizar contenedores de Docker y ofrecer información detallada sobre el estado del clúster y de sus instancias
- El servicio debe permitir actualizar con facilidad las versiones de los contenedores
- El servicio debe permitir iniciar contenedores nuevos automáticamente a partir de la imagen actualizada, y detener los contenedores que ejecuten la versión anterior.
- El servicio debe permitir recuperar automáticamente los contenedores con estado incorrecto para asegurarse de que disponga de la cantidad deseada de contenedores para su aplicación.
- El servicio debe incluir varias estrategias de programación que coloquen los contenedores en clústeres en función de las necesidades de recursos (por ejemplo, CPU o RAM) y los requisitos de disponibilidad.
- El servicio debe permitir programar tareas en lote, servicios y aplicaciones de ejecución prolongada, y procesos daemon.
- El servicio debe permitir personalizar la manera en la que se ubican las tareas en un clúster de recursos de cómputo en función de atributos integrados, como tipo de recurso de cómputo, su ubicación o atributos personalizados que se hayan definido.
- El servicio debe permitir usar estrategias de ubicación como empaquetamiento y distribución para definir con mayor precisión dónde se colocarán las tareas.
- El servicio debe poder brindar aislamiento a contenedores para controlar la manera en la que los contenedores se conectan con otros servicios y con tráfico externo.

- El servicio se debe poder integrar con el servicio de balanceo de carga para aplicaciones.
- El servicio debe permitir implementar la política de acceso de "privilegios mínimos"
- El servicio debe proporcionar funciones de monitorización para los contenedores y clústeres
- El servicio debe contar con métricas a nivel de recurso de cómputo y contenedor
- El servicio debe permitir configurar alarmas que avisen cuando sea necesario incrementar o reducir la capacidad de los contenedores o clústeres.
- Se deben poder registrar todas las llamadas a la API que haga el servicio. La información registrada debe incluir la identidad del intermediario de la API, la hora a la que se produce la llamada, la dirección IP de origen del intermediario, los parámetros de solicitud y los elementos de respuesta enviados por el servicio.
- El servicio debe permitir realizar análisis de seguridad, controlar cambios a recursos y llevar adelante auditorías de conformidad.
- El servicio debe tener la capacidad para establecer límites de CPU / memoria a nivel de tarea.
- El servicio debe contar con una vista gráfica del estado del clúster
- El servicio debe soportar la gestión de identidades y acceso
- El servicio debe contar con Métricas a nivel de instancia y contenedor
- El servicio debe soportar IP por tarea / Pod
- El servicio debe soportar funcionalidades de auditoría y registro
- El servicio debe soportar escalamiento automático
- El servicio debe soportar balanceo de carga
- El servicio debe contar con acceso y autenticación de tareas
- El servicio debe tener una disponibilidad de 99,99%.

4.2.11. Servicio de registro de contenedores (Amazon Elastic Container Registry)

- El servicio debe ser completamente administrado
- El servicio debe permitir almacenar fácilmente y ejecutar imágenes de contenedores para aplicación con cualquier organizador
- El servicio debe ser compatible con los estándares Open Container Initiative (OCI) y Docker Registry HTTP API V2
- El servicio debe permitir descubrir y utilizar software de contenedores que distribuidores y desarrolladores de proyectos de código abierto y comunidades compartan de manera pública
- El servicio debe permitir almacenar los artefactos e imágenes de contenedores con una durabilidad de los datos del 99,999999999 %
- El servicio debe permitir replicar de manera automática los datos en varias ubicaciones para las aplicaciones que requieren alta disponibilidad.
- El servicio debe contar con capacidad de definir y organizar repositorios en el registro mediante el uso de espacios de nombres
- El servicio debe permitir organizar los repositorios en función de los flujos de trabajo existentes del equipo.
- El servicio debe permitir definir qué acciones de la API pueden realizar otros usuarios en el repositorio (por ejemplo, crear, enumerar, describir, eliminar y obtener) mediante políticas de nivel de recursos

- El servicio debe permitir controlar y monitorear quién y qué pueden obtener acceso a las imágenes del contenedor
- El servicio debe permitir ajustar políticas y especificar permisos diferentes para usuarios y roles distintos, como, por ejemplo, acceso para inserción, extracción o acceso completo de administrador.
- El servicio debe permitir transferir las imágenes del contenedor mediante HTTPS
- El servicio debe permitir que las imágenes se cifran automáticamente en reposo mediante el cifrado del lado servidor.
- El servicio debe permitir elegir una clave propia administrada para cifrar imágenes en reposo.
- El servicio debe permitir gestionar políticas de ciclo de vida del registro de contenedores.
- El servicio debe contar con una disponibilidad mínima de 99,9%

4.2.12. Servicio de monitoreo (Amazon Cloudwatch)

- El servicio debe permitir monitorear aplicaciones y recursos de infraestructura locales, híbridos y de la nube.
- El servicio debe permitir recopilar y obtener acceso a todos los datos de rendimiento y operaciones en formato de registros y métricas a partir de una sola plataforma
- El servicio debe ofrecer visibilidad de hasta 1 segundo de las métricas y los datos de los registros, 15 meses de retención de datos (métricas) y la capacidad para realizar cálculos con las métricas
- El servicio debe permitir visualizar y analizar el estado, el rendimiento y la disponibilidad de sus aplicaciones en un solo lugar.
- El servicio debe tener la capacidad de contar una visión completa de las aplicaciones y sus dependencias.
- El servicio debe tener la capacidad de hacer monitoreo de las aplicaciones en tres dimensiones: monitoreo de infraestructura (con métricas y registros para comprender los recursos que respaldan sus aplicaciones), monitoreo de transacciones (con rastreos para comprender las dependencias entre sus recursos) y monitoreo de usuario final (para monitorear sus puntos de enlace y notificarle cuando su experiencia de usuario final se haya degradado)
- El servicio debe permitir monitorear puntos de enlace de la aplicación
- El servicio debe permitir escribir reglas para indicar los eventos de interés para la aplicación y las acciones automatizadas que se deben desencadenar cuando una regla concuerde con un evento.
- El servicio debe facilitar el diagnóstico, aislamiento y corrección de problemas
- El servicio debe permitir realizar análisis históricos para optimizar costos y obtener información en tiempo real sobre los recursos de la infraestructura y la optimización de las aplicaciones.
- El servicio debe permitir recopilar hasta 50 métricas predeterminadas de servicios de la nube
- El servicio debe permitir crear gráficos reutilizables y ver las aplicaciones y los recursos de la nube en una vista unificada
- El servicio debe permitir monitorear contenedores
- El servicio debe contar con granularidad configurable de monitoreo/alerta
- El servicio debe permitir correlacionar el patrón de registros de una métrica específica y definir alarmas para que avisen de manera proactiva acerca de problemas operativos y de rendimiento
- La funcionalidad de alarmas debe permitir definir un umbral de métricas y activar una acción.
- El servicio debe permitir hacer correlaciones entre registros y métricas

- El servicio debe contar con mecanismos de búsqueda
- El servicio debe admitir el uso de percentiles
- El servicio debe permitir monitorear el rendimiento operativo, resolver errores y detectar tendencias
- El servicio debe permitir minimizar los tiempos de inactividad y el potencial impacto en el desempeño de la solución
- El servicio aplica algoritmos de aprendizaje automático para analizar los datos de las métricas de manera permanente y detectar los comportamientos anormales
- El servicio debe permitir controlar qué usuarios y recursos tienen permiso para obtener acceso a sus datos y de qué manera lo hacen
- El servicio debe permitir cifrar los datos en tránsito y en reposo.

4.2.13. Servicio de auditoría y registro (AWS Cloudtrail)

- El servicio debe permitir visualizar y registrar actividades en la cuenta de nube
- El servicio debe permitir obtener registros agregados de varias cuentas de la nube
- El servicio debe permitir visualizar y descargar registros con hasta 90 días de antigüedad
- El servicio debe permitir comprimir los archivos de registros
- El servicio debe permitir visualizar, buscar y descargar registros de actividades de las cuentas
- El servicio debe permitir establecer si los archivos de registro no han sido alterados, tienen algún cambio o han sido borrados
- El servicio debe garantizar que los registros son encriptados usando server-side encryption (SSE)

4.2.14. Servicio de gestión de llaves (AWS KMS)

- El servicio debe permitir crear y administrar con facilidad las claves y controlar el uso del cifrado en una amplia variedad de servicios de la nube y en las aplicaciones.
- El servicio debe contar con módulos de seguridad de hardware que sirvan para proteger las claves y que han sido validados según las normas FIPS 140-2, o están en proceso de validación.
- El servicio debe permitir agregar de manera sencilla funcionalidades de cifrado y firma digital en el código de la aplicación.
- El servicio debe tener la capacidad de ejercer un control centralizado del ciclo de vida y los permisos de las claves.
- El servicio debe permitir importar claves desde una infraestructura de administración de claves propia o utilizar las claves almacenadas.
- El servicio debe permitir seleccionar la rotación automática anual de claves maestras generadas para no tener que volver a cifrar datos que ya lo estaban.
- El servicio debe conservar de manera automática versiones anteriores de la clave maestra para descifrar datos cifrados con antelación.

- El servicio debe permitir administrar las claves maestras y auditar su uso desde la consola de administración de la nube o desde una interfaz de línea de comandos.
- El servicio de gestión de llaves se debe poder integrar con otros servicios de la nube.
- El servicio debe soportar la ejecución de auditorías; es decir, cada solicitud que se haga en el servicio se debe anotar en un registro. La información registrada debe incluir los detalles del usuario, la hora, la fecha, la acción de API y, cuando corresponda, la clave utilizada.
- El servicio debe ser totalmente administrado.
- El servicio debe poder escalar automáticamente según se requiera.
- El servicio debe almacenar varias copias de las versiones cifradas de las claves en sistemas diseñados para ofrecer una durabilidad del 99,999999999 %, a fin de garantizar que la disponibilidad de las claves y los datos sea alta.
- El servicio debe soportar la creación automáticamente copias de seguridad de las copias cifradas de las claves con el fin de mantener control total sobre el proceso de recuperación.
- El servicio debe permitir crear un almacenamiento de claves propio con HSM.
- El servicio debe ofrecer la posibilidad de crear y usar Customer Master Keys (CMK) asimétricas.
- El servicio debe permitir generar un par de claves de datos asimétricas. La operación debe devolver una copia con texto no cifrado de la clave pública y la clave privada, así como también una copia de la clave privada cifrada con una CMK simétrica que se especifique.
- El servicio debe permitir usar clave privada o pública con texto no cifrado en la aplicación local y almacenar la copia cifrada de la clave privada para un uso futuro.

4.2.15. Servicio de administración de costos en AWS

- El servicio debe permitir visualizar, comprender y administrar sus costos y uso de la nube a lo largo del tiempo.
- El servicio debe permitir generar reportes personalizados que analizan los datos de costos y uso.
- El Partner debe brindar acceso a los reportes de costo que pueda generar esta herramienta.
- El servicio debe permitir identificar tendencias, determinar los factores de costo y detectar anomalías.
- El servicio debe permitir predecir el uso y costos futuros de los servicios de nube teniendo en cuenta información histórica.
- El servicio debe permitir establecer períodos de tiempo personalizados y determinar si se desea ver los datos en un nivel de granularidad mensual o diario.
- El servicio debe permitir profundizar en los datos aprovechando la funcionalidad de filtrado y agrupación
- El servicio debe permitir guardar el progreso como un nuevo informe para volver a consultarlo en el futuro.
- El servicio debe permitir ajustar el intervalo de tiempo que abarcan los informes para ver los datos históricos pertenecientes a los últimos doce meses, con el fin de comprender las tendencias de los costos.
- El servicio debe permitir profundizar mediante dimensiones de facturación granular como tipo de uso y etiquetas.

4.2.16. Servicio de optimización de costos y rendimiento

- El servicio debe ofrecer recomendaciones para que la Entidad pueda seguir las mejores prácticas de la nube.
- El servicio debe permitir identificar formas de optimizar su infraestructura de la nube, mejorar la seguridad y el rendimiento, reducir los costos y monitorear las cuotas de servicios.
- El servicio debe ayudar a la entidad a ahorrar, pues debe sugerir eliminar recursos sin usar o inactivos, o usar capacidad bajo ofertas económicas, entre otras cosas.
- El servicio permite detectar recursos de cómputo que se estén utilizando por encima de su capacidad para mejorar el rendimiento de los servicios.
- El servicio debe identificar oportunidades de uso de la funcionalidad de escalado automático.
- El servicio debe permitir revisar los permisos para mejorar la seguridad de la aplicación.

4.2.17. Otros servicios

Se deben incluir los servicios del portafolio ofrecido por AWS con todas las funcionalidades que nos no superen el monto contratado

4.3. Consideraciones de seguridad

4.3.1. Seguridad de la infraestructura

La nube debe ofrecer varias capacidades y servicios de seguridad para aumentar la privacidad y controlar el acceso a la red. Estos incluyen:

- La nube debe permitir crear redes privadas y controlar el acceso a sus recursos de cómputo o aplicaciones.
- La Entidad debe poder controlar el cifrado en tránsito con TLS a través de los servicios de la nube.
- La nube debe ofrecer opciones de conectividad que permitan conexiones privadas o dedicadas desde la Entidad.
- La nube debe contar con tecnologías de mitigación de DDoS que se aplican en la capa 3 o 4, así como en la capa 7.
- La nube debe contar con cifrado automático de todo el tráfico en las redes globales y regionales

4.3.2. Gestión de inventario y configuración

La nube debe ofrecer una gama de herramientas que permitan cumplir con los estándares y las mejores prácticas de la Entidad. Estos incluyen:

- La nube debe contar con herramientas de implementación para administrar la creación y desmantelamiento de recursos
- La nube debe contar con herramientas de administración de inventario y configuración para identificar recursos y luego rastrear y administrar los cambios en esos recursos a lo largo del tiempo.
- La nube debe contar con la capacidad de definir plantillas y herramientas de administración para crear recursos de cómputo estándar y pre-configurados.

4.3.3. Cifrado de datos

La nube debe ofrecer la posibilidad de agregar una capa de seguridad a los datos en reposo, proporcionando funciones de cifrado escalables y eficientes. Estos incluyen:

- La nube debe contar con capacidades de cifrado de datos en reposo en los servicios requeridos por la Entidad
- La nube debe contar con opciones flexibles de administración de claves, incluido un servicio de administración de claves propio de la nube
- La nube debe contar con almacenamiento de claves criptográficas dedicado y basado en hardware

4.3.4. Control de identidad y acceso

La nube debe ofrecer capacidades para definir, hacer cumplir y administrar las políticas de acceso de los usuarios en todos los servicios. Estos incluyen:

- La nube debe contar con un servicio que le permita a la Entidad definir cuentas de usuario individuales con permisos en todos los recursos de la nube.
- La nube debe contar con autenticación multi-factor para cuentas privilegiadas, incluidas opciones para autenticadores basados en software y hardware.
- La nube debe contar con un servicio que permita otorgar a los funcionarios dla Entidad y a las aplicaciones acceso federado a la consola de administración.
- La nube debe contar con inicio de sesión único (SSO) que le permita a la Entidad administrar el acceso SSO y los permisos de usuario a todas sus cuentas de la nube de manera centralizada

4.3.5. Monitoreo y registro

La nube debe proporcionar herramientas y características que le permitan a la Entidad ver lo que sucede en su entorno de nube. Estos incluyen:

- La nube debe contar con un servicio que permita monitorear las implementaciones en la nube al obtener un historial de llamadas API para la cuenta de la Entidad
- La nube debe permitir identificar qué usuarios y cuentas llamaron API de la nube para los servicios que admiten esta funcionalidad. En particular debe permitir hacer seguimiento de la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron las llamadas.
- La nube debe contar con una solución de monitoreo confiable, escalable y flexible
- La nube debe contar con un servicio de detección de amenazas que monitorea continuamente la actividad maliciosa y el comportamiento no autorizado para proteger la cuenta de la Entidad y cargas de trabajo que este ejecutando por la Entidad en la nube.
- La nube debe exponer las notificaciones para que pueda activar una respuesta automatizada o notificar a la persona designada por la Entidad.
- La nube debe contar con herramientas y características que den visibilidad a la Entidad para detectar problemas antes de que afecten el desarrollo de las actividades soportadas por la nube y permitan mejorar la postura de seguridad y reducir el perfil de riesgo.

5. Responsabilidades del proponente

5.1. Administración de cuentas

- Aprovisionar y eliminar cuentas de la nube mediante programación con las API para garantizar la uniformidad
- Permitir que las cuentas de gestión de acceso e identidad externas habiliten y deshabiliten usuarios
- Proporcionar inicio de sesión único a la consola de administración de la nube para que los usuarios de cuentas de la nube administren los recursos de la nube.
- Implementación de esquemas de integración con proveedores de gestión de identidad y acceso externos como Active Directory
- Admitir la gestión de tokens MFA para los usuarios ROOT de las cuentas
- Asociar cuentas de la nube con una o más cuentas de facturación maestras
- Asociar usuarios con políticas de gestión de acceso e identidad para controlar el acceso
- Apoyar la jerarquía organizativa de varios niveles
- Proporcionar un flujo de trabajo de autoservicio que permita a los usuarios unirse a proyectos
- Proporcionar un flujo de trabajo de autoservicio que permita a los usuarios crear nuevos proyectos.
- Proporcionar un flujo de trabajo de autoservicio que permita a los usuarios conectar una o más cuentas.
- Controlar el acceso a imágenes de máquinas personalizadas
- Permitir que los usuarios accedan a la API de la nube, la Consola de administración de la nube y los SDK.

5.2. Gestión de costes y presupuesto

- Generar reportes de la acumulación de gastos actuales de las cuentas de la nube
- Generar reportes del gasto agregado de las cuentas de la nube según la estructura y el propósito de la Entidad
- Aplicar restricciones de costos a las cuentas de la nube (por ejemplo, forzar el uso de ciertos tipos de instancias, restringir el uso de instancias a instancias de menos de $ x / h, etc.)
- Establecer reglas para definir acciones de cumplimiento (incluida la notificación, limitar la creación de nuevos recursos en la nube, archivar recursos en la nube y finalizar los recursos en la nube) cuando se alcanzan los umbrales financieros para cada cuenta de la nube
- Enviar alertas a las partes interesadas cuando se cumplan los límites y umbrales predefinidos

5.3. Soporte del Proveedor de Servicios de Nube

Los servicios directos del Proveedor con el plan de soporte AWS Business Support

5.4. Transición del servicio al término de la orden de compra

El Proponente antes de finalizar la orden de compra, está obligado a colaborar con la transferencia de los servicios prestados durante la vigencia del mismo, así como entregar las configuraciones de los equipos y/o servicios que soportan la solución, proporcionando a la Entidad de forma enunciativa más no limitativa lo siguiente: Usuarios, contraseñas, e información o cualquier otro recurso relacionado a la de seguridad de la solución y los servicios prestados.

El periodo de transición iniciará 30 días hábiles previos a la terminación del contrato con el Proponente y deberá concluir al término de la vigencia de la orden de compra.

Adicionalmente, el Proponente debe entregar respaldos de toda la solución a la Entidad al cabo de este contrato en los dispositivos físicos o virtuales que se determinen (máquinas virtuales, bases de datos y almacenamiento, así como claves o cualquier elemento necesario para el buen funcionamiento de dicha solución en otro entorno físico o nube).

El Proponente también debe transferir la cuenta maestra de gestión de la infraestructura de la Universidad ya sea a la Entidad o a quien la Universidad designe al finalizar el presente contrato. Este acto de entrega representa el fin de las responsabilidades financieras del Proponente con el Proveedor de Servicios de Nube. En ese momento, ya sea la Entidad o quien ésta designe, pasará a ser responsable por cualquier elemento técnico relacionado al servicio de nube, así como a los costos asociados con la misma.

5.5. Bolsa de recursos de servicios de nube

La cláusula 4 de la minuta del Acuerdo Marco de Nube Pública establece que: “Los Servicios del Catálogo de Servicios de Computación en la Nube aparecen en la Orden de Compra como una sola línea correspondiente a la bolsa de recursos para adquirir Servicios de Computación en la Nube.” En virtud de lo anterior la Entidad

gestionará los recursos correspondientes a los Servicios de Computación en la Nube como una bolsa de recursos. Es decir, durante la ejecución de la orden de compra la Entidad podrá optimizar la arquitectura inicialmente planteada y hacer cambios respetando el valor por el que fue pactada la orden de compra o haciendo uso de las alternativas legalmente disponibles para ampliar el valor y duración del contrato.

5.6. Enfoque Y metodología de la solución propuesta (Proponente)

i. Alcance: descripción de la necesidad que se está resolviendo

ii. Supuestos

iii. Solución y arquitectura

iv. Metodología utilizada

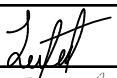
v. Servicios profesionales

vi. Cronograma

vii. Definición de responsabilidades

viii. Seguridad y privacidad en el manejo de la información

ix. Términos y condiciones de la propuesta

	NOMBRE	CARGO	FECHA	FIRMA
Revisó y Aprobó	Alejandro Paolo Daza Corredor	Jefe Oficina Asesora de Sistemas	enero 2022	
	José Ignacio Palacios Osma	Coordinador de Planestic-UD	enero 2022	
Proyectó	Luis Fernando Torres Romero	Asesor - Oficina Asesora de Sistemas	enero 2022	
	Diana Paola Guayará Castro	Asesor - Oficina Asesora de Sistemas	enero 2022	
	Jhoan Eduardo Villa Lombana	CPS - Planestic-UD	enero 2022	
	Sandra Milena Silva Ávila	CPS - Planestic-UD	enero 2022	
	Hernán Darío Lozano Rojas	CPS - Planestic-UD	enero 2022	

✓