LINEAMIENTOS DE DESARROLLO DE SOFTWARE

Subdirección de Soluciones y Desarrollo

Dirección de Gestión de Innovación y Tecnología

Versión 1.3

Año 2025

Tabla de Contenido

Versión	Vigencia		Descripción de Cambios
	Desde	Hasta	
1	25/04/2025	23/05/2025	Versión inicial
1.1.	28/06/2025	01/07/2025	Se incluye numeral para requerimientos no funcionales. Se detalla lo relacionado con pruebas Se incluye arquitectura transición para facture electronica Se actualiza anexo “Lineamientos de codificación para proyectos basados en c Sharp” Se incluyen los anexos RFC de factura electronica
1.3.	2/07/2025		Se actualiza vista de interoperabilidad para incluir a BAMOE Se incluye el análisis de dependencias en el numeral 9.10 Se incluye Revisión de Complejidad y Patrones de Diseño.

Elaboró:	Equipo de Arquitectura Equipo de Seguridad		Subdirección de Soluciones y Desarrollo	
Revisó:	Climaco Alberto Llamas Caamano	Inspector IV	Subdirección de Soluciones y Desarrollo	
Aprobó:	Antonio Jose Barrios Hoyos	Subdirector	Subdirección de Soluciones y Desarrollo	

1. Introducción

El presente documento tiene como propósito establecer los lineamientos y buenas prácticas para el desarrollo de software en la DIAN, busca garantizar la calidad, seguridad, mantenibilidad y alineación con las políticas institucionales y los lineamientos de Gobierno Digital. De igual manera pretende estandarizar los procesos de desarrollo, facilitar la interoperabilidad entre sistemas, proteger la información y promover el uso de tecnologías definidas en la arquitectura objetivo para soportar las necesidades de la entidad.

Con el fin de garantizar los aspectos mencionados, este documento es de obligatorio cumplimiento para todos los proyectos de desarrollo de software adelantados en la DIAN y está dirigido a todas las partes interesadas que participan en los procesos de planificación, diseño, desarrollo, pruebas, despliegue, operación y mantenimiento de software dentro de la DIAN.

2. Alcance

Este documento aplica tanto para desarrollos de software tanto internos como externos (Fábricas de software)

Este documento tiene revisiones periódicas y que puede ser modificado de acuerdo con la evolución de la tecnología y las buenas prácticas vigentes.

3. Normatividad y Referencias

La DIAN, como entidad estatal, busca dar cumplimiento y adoptar las buenas prácticas establecidas en las normas, estándares y lineamientos que sirven como base para el desarrollo de software en la organización, para de esta forma garantizar el cumplimiento legal, técnico y buenas prácticas nacionales e internacionales.

A continuación, se listan normas y documentos, tanto a nivel general (Estado colombiano) como internos de la entidad, que son de cumplimiento obligatorio en el desarrollo de software para la entidad.

3.1. Normas Generales

- Decreto 767 de 2022 Política de Gobierno Digital MinTIC.
- Manual de gobierno digital MinTIC última versión.
- Marco de referencia de arquitectura empresarial MinTIC última versión.
- Ley 1581 de 2012: Protección de datos personales y su reglamentación.
- Resolución MinTIC 1519 del 2020 Anexo 1 “Directrices de accesibilidad web”.
- Resolución MinTIC 1519 del 2020 Anexo 3 “Condiciones mínimas técnicas y de seguridad digital”
- Normas NTC-ISO aplicables: ISO 27001 para seguridad de la información.
- ISO 25010 para calidad de software.
- Decreto 88 de 2022 MinTIC
- Guía de Desarrollo de Software Seguro para Entidades Públicas Anexo I Cloud v.1 - MINTIC.
- Guía de Desarrollo de Software Seguro para Entidades Públicas Anexo Web.3.0 v.1 – MINTIC
- Guía de Desarrollo de Software Seguro V1.0 - MINTIC

3.2. Lineamientos internos DIAN

Los siguientes documentos son de obligatorio cumplimento:

- Arquitectura de Referencia DIAN.
- Lineamientos Codificación Mejores Prácticas Equipos Desarrollo.
- Estándares Construcción Código Java.
- Arquitectura y guía de desarrollo usando buenas prácticas para aplicaciones Angular Versión 3.0.
- Guía de desarrollo usando librerías DIAN (Angular).
- Anexo Lineamientos de codificación para proyectos basados en c sharp.
- Lineamientos Adopción Vistas Arquitectónicas.
- Vistas Arquitectónicas Desarrollo Software.
- IN-IIT-Repositorios Pruebas.
- Matriz de Riesgos Proyecto Tecnológico.
- Lineamientos soluciones SaaS.
- Anexo RFC Factura Electrónica. RFC_ Factura_ Electrónica.xlsx
- Anexo Documento de Caracterización Despliegues en Azure Factura Electrónica - RFC. Documento de Caracterización Despliegues en Factura Electrónica - RFC.docx
- Manual de políticas y lineamientos de seguridad de la información. MN-IIT-0072 (DIAN)
- Manual de políticas y lineamientos de protección de datos personales MN-IIT-0062 (DIAN)
- Lineamientos de desarrollo seguro fábricas de software.

Los documentos internos de la DIAN en su proceso de mejora continua pueden ser ajustados y las nuevas versiones rigen desde las fechas de sus publicaciones.

4. Principios

Dentro de los principios de calidad de software según la norma ISO 25010, encontramos:

La DIAN establece los siguientes principios para regir la calidad en el desarrollo de software.

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.
La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad.

Adecuación Funcional: Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas de los usuarios cuando el producto se usa en las condiciones especificadas.

Eficiencia de Desempeño: Esta característica representa el desempeño de un producto en la realización de sus funciones dentro de unos parámetros de tiempo y rendimiento especificados y con un uso eficiente de recursos (CPU, memoria, almacenamiento, energía...) utilizados bajo determinadas condiciones.

Compatibilidad: Capacidad de un producto de intercambiar información con otros productos y/o llevar a cabo sus funciones requeridas cuando comparten un mismo entorno y recursos.

Capacidad de Interacción: Capacidad del producto software para que el usuario interactúe mediante su interfaz intercambiando información para completar determinadas tareas.

Fiabilidad: Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados sin interrupciones o fallos.

Seguridad: Capacidad de protección de la información y los datos de manera que las personas u otros productos tengan el grado de acceso a los datos adecuado a sus tipos y niveles de autorización, y para defenderse de los patrones de ataque de agentes malintencionados

Mantenibilidad: Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Flexibilidad: Capacidad del producto para adaptarse a cambios en sus requisitos, contextos de uso o entorno del sistema.

Protección: Esta característica representa la capacidad del producto, en condiciones definidas, de evitar un estado en el que se ponga en peligro la vida humana, la salud, la propiedad o el medio ambiente.

5. Arquitectura de referencia

La arquitectura de referencia es un marco conceptual que proporciona una estructura común para diseñar y desarrollar sistemas dentro de un dominio de negocio específico. Actúa como una "plantilla" o modelo base, reutilizable, que guía la construcción de soluciones, garantizando consistencia, interoperabilidad y buenas prácticas, siendo abstracto, reusable, estandarizado y modular.

Tomando como base los siguientes requerimientos se elaboró la primera versión de arquitectura de referencia:

- Infraestructura existente y proyectos de modernización en curso (proyecto multinube que permite a la DIAN utilizar Azure y AWS, infraestructura on-premise, proyecto DataR).
- Portabilidad: permite que las aplicaciones y los datos se muevan fácilmente entre diferentes entornos sin requerir cambios significativos. El propósito no es tener una portabilidad al 100%, pero debe ser posible lograrlo en caso de requerirlo.
- Se busca estandarización para facilitar el desarrollo y la administración de las soluciones.
- Uso de estándares y buenas prácticas de la industria.
- Conocimiento previo en desarrollo (el software misional que actualmente esta producción fue desarrollado al interior de la DIAN).

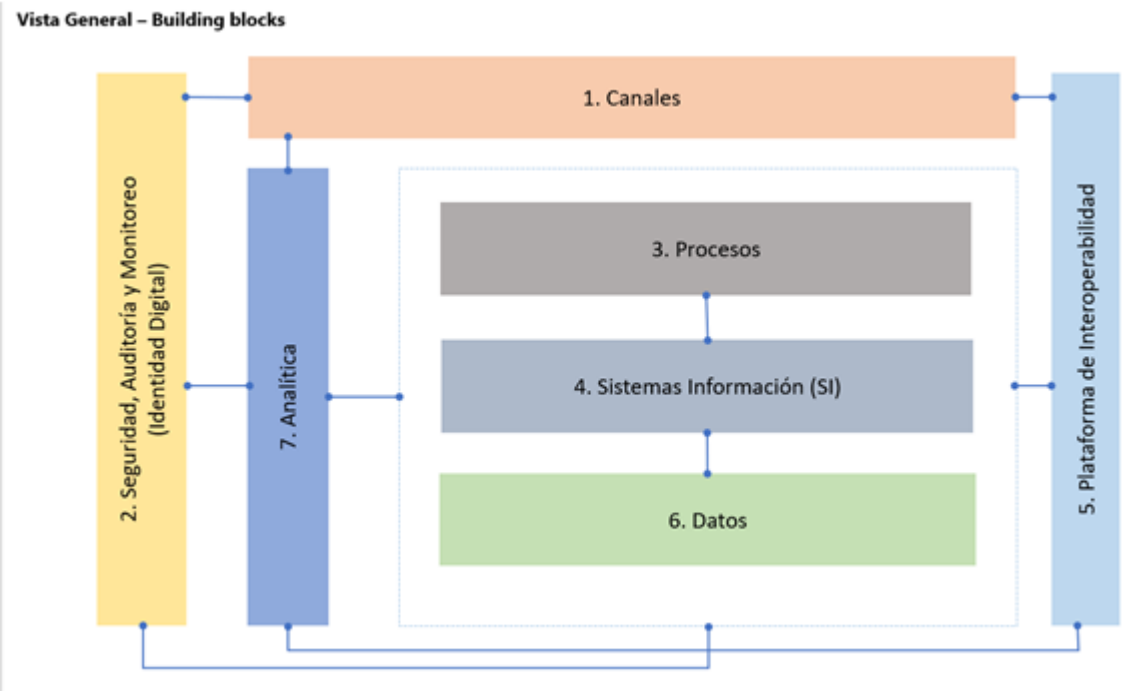
A continuación, se relacionan algunos lineamientos tenidos en cuenta para garantizar la portabilidad:

- Contenerización.
- En lo posible utilizar herramientas y servicios agnósticos que no dependan de características específicas de un proveedor de nube asegura la independencia y portabilidad.
- Desacoplamiento de servicios: microservicios, micro-frontend.
- Uso de formatos de datos estándar como JSON, XML, o Parquet.

- Infraestructura como código (IaC).
- Observabilidad y trazabilidad: monitoreo centralizado.
- Implementar políticas de seguridad consistentes en todos los entornos, incluyendo autenticación, autorización y cifrado.
- Utilizar administración de identidades y acceso de manera uniforme para gestionar accesos y permisos en diferentes plataformas, garantizando la seguridad y portabilidad.
- Integración y entrega continuas (CI/CD).

La metodología utilizada para definir la Arquitectura de Referencia fue la siguiente:

- Identificar y establecer los Buildings Blocks (ABB – Architecture Building Blocks) para la entidad.
- Identificar las capacidades requeridas para soportar los Buildings Blocks, atendiendo los lineamientos y requerimientos establecidos.
- Evaluar las alternativas de los servicios para suplir las capacidades en la multinube híbrida de la DIAN (sobre un modelo técnico y de costos), garantizando un alto nivel de portabilidad.
- Establecer los servicios finales a utilizar y generar las vistas correspondientes para su socialización.

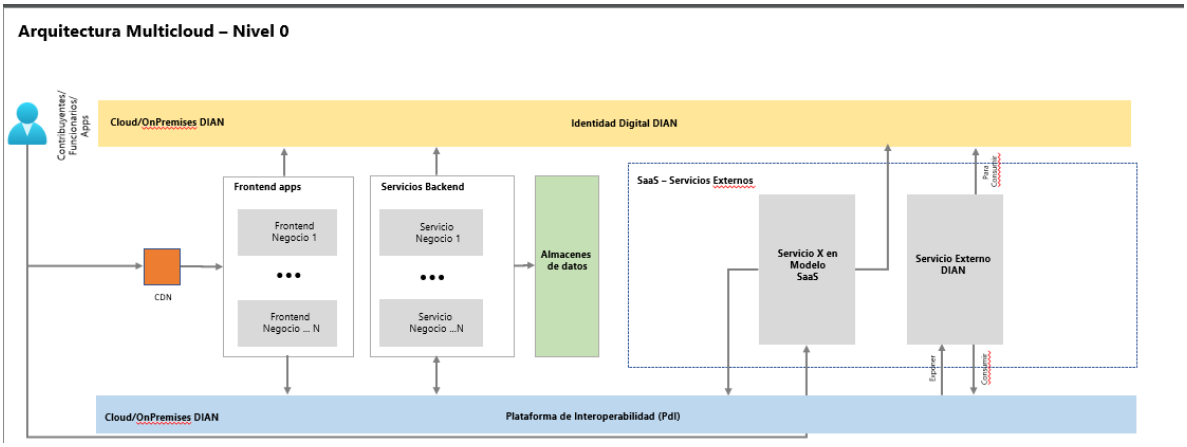


Vista General - Arquitectura de Referencia.
Fuente: Equipo de Arquitectura Digital

Generalidades arquitectura de referencia

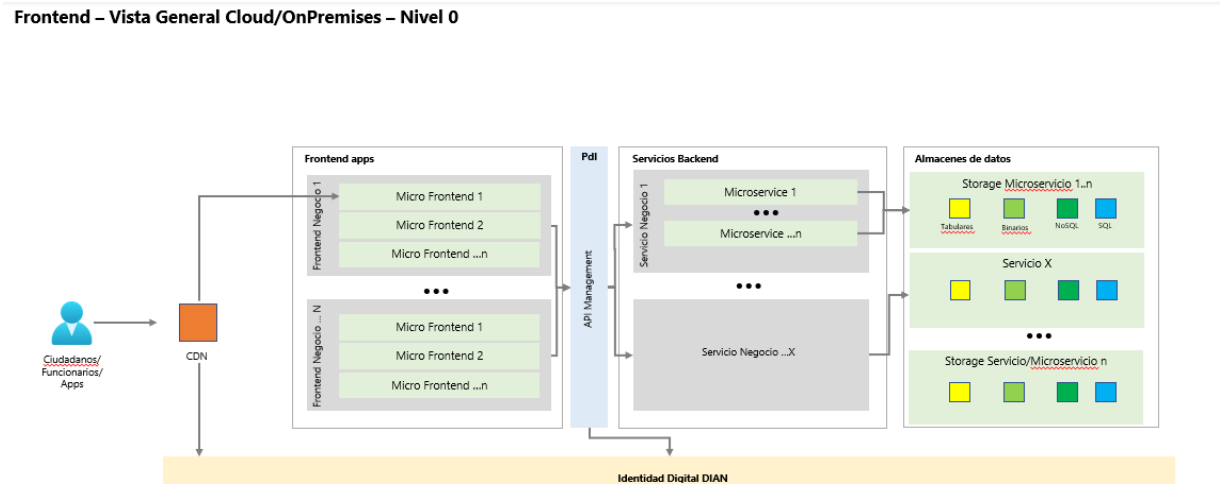
- La Arquitectura de referencia debe ser utilizada para las soluciones que se desplieguen total o parcialmente en la nube (pública o privada) u on-premises.

- El componente de Identidad Digital (IAM)[1] se encarga de brindar capacidades Single Sing On (SSO) a todos los servicios.
- La nube de los terceros solo es válida para soluciones en modelo SaaS que sean previamente autorizadas por la DIAN.
- La plataforma de Interoperabilidad (PdI) es transversal a todos los servicios y está asegurada contra el componente de Identidad Digital.
- La comunicación entre el Frontend y el Backend solo puede realizarse mediante la Plataforma de Interoperabilidad (PdI).



Nivel 0 - Arquitectura de Referencia.
Fuente: Equipo de Arquitectura Digital

CDN (Content Delivery Network)
SaaS (Software as a Service): Software como servicio, las soluciones operan en la infraestructura del proveedor del servicio



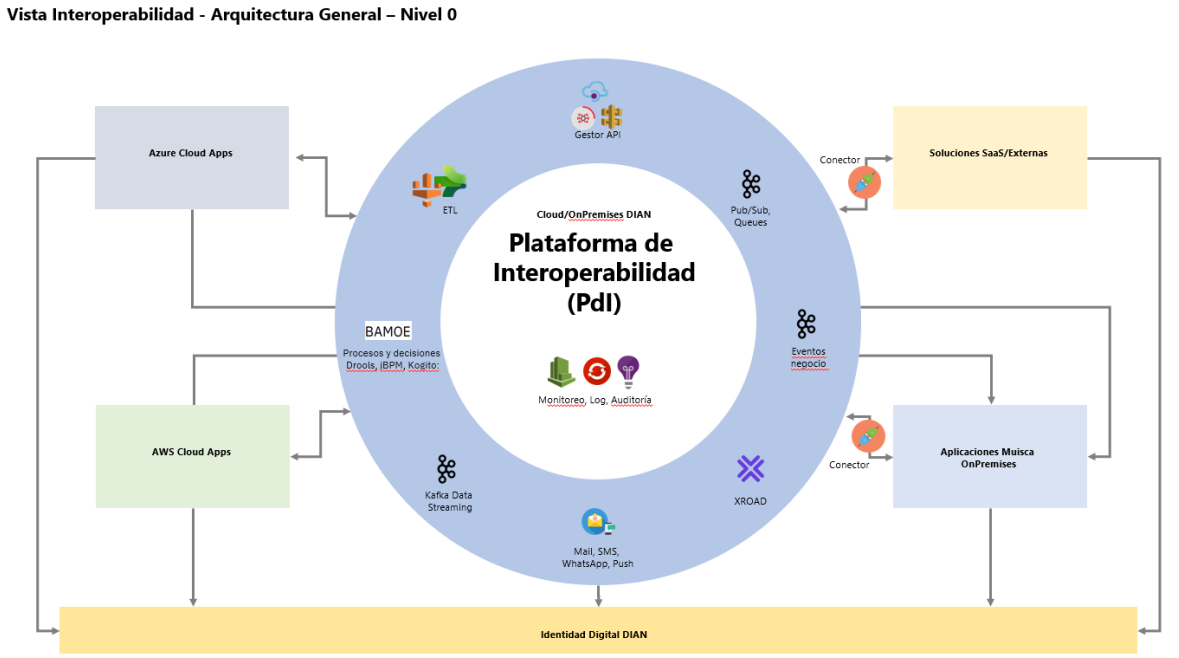
FrontEnd - Arquitectura de Referencia.

[1] El componente de identidad aquí mencionado es propio de la DIAN y no corresponde al componente de identidad de servicios ciudadanos digitales de MinTIC

Fuente: Equipo de Arquitectura Digital

Vista General Cloud/On-Premises – Nivel 0

- Los microfrontends generalmente consumen uno o más microservicios.
- En almacenes de datos encontramos varios tipos: No-SQL, SQL, Binarios, Key/Value, Cache.
- Los microservicios deben contenerizarse.



Vista Interoperabilidad - Arquitectura de Referencia.
Fuente: Equipo de Arquitectura Digital

Generalidades plataforma de interoperabilidad

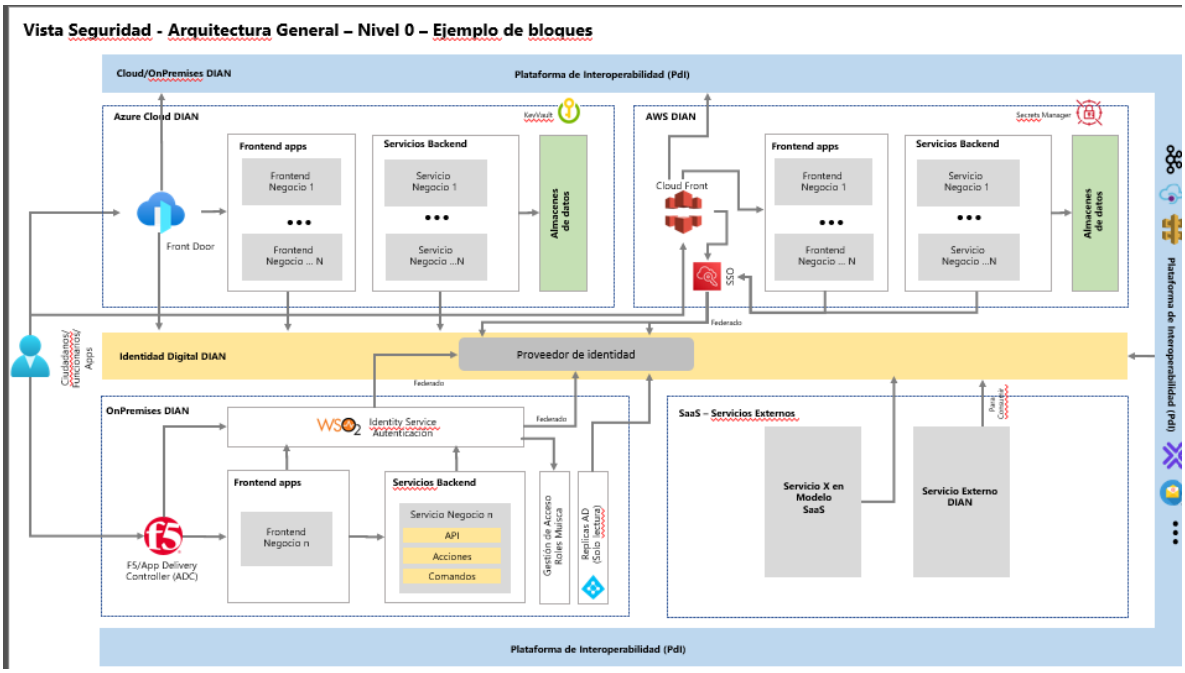
La Plataforma de Interoperabilidad cubre los escenarios permitidos para interoperabilidad:

- Publicación y Consumo de API.
- Publicación y consumo de eventos.
- Colas de mensajes, publicación y suscripción de mensajes.
- Soporte de interoperabilidad con XROAD de MinTic.
- Uso de servicios de correo, SMS, WhatsApp, Pushing.
- Streaming de datos y IoT.
- Orquestación de servicios mediante la automatización.
- Capacidades de transformación, carga y extracción de datos.
- Las aplicaciones dispuestas en nube u on-premises utilizan la PdI para comunicarse con sus backends en nube u on-premises.
- Las aplicaciones de terceros pueden consumir servicios disponibles en el ecosistema de la DIAN.
- Para la automatización y orquestación integrada de procesos de negocio y decisiones, incluyendo la gestión de casos y reglas complejas, se debe usar BAMOE. Business Automation

Manager Open Edition. Plataforma de automatización basada en estándares abiertos (BPMN, DMN, DRL) con soporte empresarial (Red Hat).

Las aplicaciones de terceros deben publicar sus API (las que pueden ser consumidos) en la PdI de la DIAN.
Los protocolos de seguridad utilizados son OpenID Connect (OIDC) con token JWT.

Se usa Kafka. Kafka es una plataforma distribuida de gestión de eventos que permite almacenar, transmitir y procesar grandes volúmenes de datos en tiempo real.



Vista Seguridad - Arquitectura de Referencia.
Fuente: Equipo de Arquitectura Digital

- Los roles, grupos y atributos son gobernados de forma centralizada.
- Esquemas de acceso a los usuarios soportados: RBAC (Role-based Access Control) y ABAC (Attributes-based Access Control).
- Los servicios externos deberán integrarse con el proveedor de identidad de la DIAN para el consumo de servicios DIAN.
- Las API expuestas por servicios externos deberán utilizar esquemas que se adapten al esquema de seguridad de la DIAN para ser expuestas en la PdI.
- El estándar de seguridad para todo el ecosistema DIAN es OIDC y JWT.

Para mayor detalle de la arquitectura de referencia se debe consultar con el equipo de arquitectura de la Subdirección de Soluciones y Desarrollo.

6. Stack tecnológico elegido por la DIAN

A continuación, se enuncia el stack tecnológico propuesto para el desarrollo de soluciones a la medida para la DIAN, el mismo puede tener cambios que se comunicaran oportunamente al proveedor.

Capa	Productos
Orquestación y Gestión de Microservicios	OpenShift
Comunicación y Orientación a eventos	Kafka
Microfront ends	Angular 21
Almacenamiento: Data Grid	Infinispan
Almacenamiento: Bases de datos NoSQL	MongoDB, CosmosDB, Oracle (capacidades no sql)
Almacenamiento: Bases de datos SQL	Oracle, SQL Server
Procesos de Negocio y Reglas	IBM Business Automation Manager Open Edition (IBM BAMOE)
Seguridad y Gobernanza	Keycloak como servidor de identidad y acceso (IAM)
Observabilidad y Resiliencia	Prometheus, Grafana
DevOps y Automatización	GitHub, Terraform, OpensShift Pipeline (IC)
Repositorio de artefactos	Artifactory
Respaldo y recuperación	Pendiente de definir por parte de la DIAN

Herramientas específicas de apoyo a la construcción

Capa	Productos
Framework para microservicios en java	Quarkus
Entorno de desarrollo	Visual Studio Code
Repositorio de código	GitHub
CI/CD (build/test/deploy)	GitHub Actions, OpenShift Pipeline (Tekton)
Seguridad	GitHub Advanced Security

El proveedor debe utilizar las herramientas, componentes, librerías, entre otras relacionadas en el stack tecnológico escrito en el presente numeral y en caso de requerir el uso de alguna diferente esta debe ser aprobada por la DIAN con el fin de que se valide su pertinencia.

Stack tecnológico factura electrónica transición

Para nuevos desarrollos que no se realicen sobre los componentes o proyectos actualmente existentes se indica que se debe apalancar en las siguientes tecnologías:

Proyecto Web (Signature.Cloud.Web):
Tecnologías: .Net 8+ LTS o versión de SDK con soporte extendido que se encuentre oficialmente soportada por Microsoft.

Lenguajes de programación: C# y JavaScript, SQL y NoSQL.

Otros Lenguajes: HTML5, CSS3, LESS

Otros archivos y/o estructuras: XML, JSON, XLST, Txt.

Componentes o Conexiones Externas: Azure Storage Account, SQL Server, Azure Cosmos DB, Azure EventGrid, Azure Functions, APIs REST, Redis Caché, Applications Insigths.

Almacenamiento utilizando componentes nativos de Azure: Storage Account, Cosmos DB, Azure SQL Server.

7. Gobernanza de datos en la DIAN (proyecto DataR)

Dentro de los lineamientos que se han definido se encuentran el documento Políticas De Gestión y Migración de Datos, el Esquema operativo y Roles para la migración de datos en la DIAN.

Se cuenta con un modelo de operaciones basada en datos, definido como DataOps, que consiste en procesos automatizados orientados a metodologías para mejorar la gobernabilidad, calidad y reducir el tiempo del ciclo de los análisis de datos. Se toman como base la arquitectura de referencia para la operación en la nube establecida por la Dirección de Gestión de Innovación y Tecnología - DGIT (el esquema se detalla a continuación), así como las consideraciones técnicas definidas en el modelo DataOps, con la visión de usuario final.

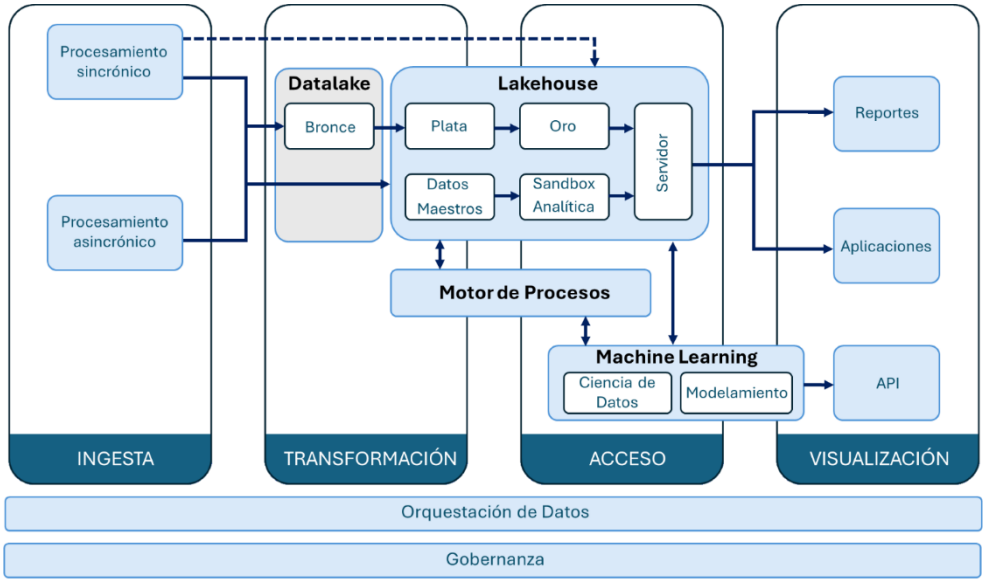


Diagrama de Arquitectura LakeHouse | Fuente: Subdirección de Procesamiento de datos - DGIT

Para garantizar la seguridad en el acceso a las capas de almacenamiento del datalake (bronce) y lakehouse (plata y oro), se cuenta con un control de acceso basado en roles. Estos roles incluyen dueños de datos, líderes de dominio, custodios, analistas de datos, analistas de calidad y otros usuarios finales.

Los usuarios podrán realizar operaciones con datos certificados por las áreas propietarias de la información, que actúan como la fuente única de datos. A partir de estos datos certificados, podrán

realizar las consultas necesarias para sus funciones y utilizar herramientas de inteligencia de negocios (BI) y consumo de datos para la generación de reportes.

8. Lineamientos

Accesibilidad: Cumplir con las directrices de Accesibilidad para sitios web suministradas por el Ministerio de Tecnologías de la información y las Comunicaciones – MinTIC (Anexo 1 de la Resolución MinTIC 1519 del 2020 “Directrices de accesibilidad web”), en cuanto al acceso y diseño donde deben cumplir con el nivel AA de accesibilidad a sitios web.

Codificación: Lo implementado deberá ser responsive, y deberá probarse en computadores (sistemas operativos: Linux, Windows y MacOS) y en dispositivos móviles (celulares que trabajen bajo sistema operativo IOS y Android).
Se debe utilizar la última versión de Angular más estable.
Lenguajes de programación: Java, C# y Phyton
El lenguaje adoptado es java, para aplicaciones en la nube Azure debe utilizarse Microsoft OpenJDK y para aplicaciones en la nube AWS debe utilizarse Amazon Correto. Se debe utilizar la última versión de soporte a largo plazo liberada LTS (Long Term Support).

DevSecOps: se adopta un enfoque DevSecOps que integra de manera continua la seguridad en todo el ciclo de vida del desarrollo, apoyándose en prácticas automatizadas de CI/CD para asegurar despliegues rápidos y confiables, y en Infraestructura como Código (IaC) para gestionar la infraestructura de forma reproducible, versionada y escalable, garantizando así calidad, seguridad y agilidad en las entregas.

Seguridad: Se debe garantizar la asignación de roles y responsabilidades en todas las fases del proyecto de desarrollo.

Pruebas: El proveedor deberá entregar la batería de pruebas automáticas en las herramientas definidas por la DIAN, se deberán hacer pruebas de rendimiento para garantizar una fluidez y experiencia de usuario adecuada.

Despliegues:
Se realizará siguiendo los procedimientos establecidos en DIAN, el despliegue lo realiza la Subdirección de Infraestructura, el proveedor realizará acompañamiento.

Sostenibilidad: Toda la implementación debe tener la facilidad de poderse modificar en cualquier momento y permitir su uso y/o modificación por parte de la entidad según esta lo requiera. El proveedor entregará código fuente, debidamente versionado.
Toda documentación que se genere deberá versionarse.

Mockups: El look and feel del sistema debe ser construido a partir de la información de la imagen corporativa de la DIAN, sujeto a aprobación por parte de DIAN y enfocado en que la experiencia del usuario (UX) sea superior.

Antes de implementar, para las funcionalidades que así lo ameriten el proveedor debe generar mockups que permitan al equipo DIAN avalar la experiencia de usuario (deseable uso de la herramienta Figma).

La herramienta utilizada para modelar y documentar es Enterprise Architect 17 en español. Se deben entregar fuentes de los diagramas generados, excepciones a este lineamiento deben ser autorizadas por el equipo de arquitectura de la DIAN.

Utilizar swagger para documentar api-rest

Se deben generar como mínimo los siguientes modelos utilizando UML:

- Modelo de contexto
- Modelo de estructura funcional (diagrama de componentes, que muestra los elementos del sistema, interfaces y conexiones entre elementos)
- Modelo de estructura de información estática (Diagrama Entidad/Relación, Diagrama de clases)
- Modelo de flujo de información (Diagrama de secuencia)
- Modelo del ciclo de vida de la información (Diagrama de estados)
- Modelo de plataforma en tiempo de ejecución (Diagrama de despliegue)

Para los nombres de las variables, documentos, ayudas, mensajes se debe utilizar el idioma español.

Adicionalmente la DIAN en el proceso de soluciones tecnológicas establece los siguientes formatos que deberían ser incluidos en la documentación:

Código	Titulo
FT-ADF-2556	Información técnica para clasificación de activos intangibles desarrollo interno
FT-IIT-1849	Diseño de casos de pruebas
FT-IIT-1851	Aceptación de pruebas de funcionalidad y salida a producción
FT-IIT-2003	Modelo de diseño
FT-IIT-2006	Historias de usuario
FT-IIT-2007	Especificación de requerimientos
FT-IIT-2008	Acta de entrega de etapa de requerimientos
FT-IIT-2180	Información de versión
FT-IIT-2428	Especificación reporte
FT-IIT-2708	Acta de entrega de la solución tecnológica
FT-IIT-2715	Especificación técnica de interoperabilidad
FT-IIT-2725	Plan de pruebas funcionales
	Plantilla de Pruebas Carga y Estrés
	Mapa de base de datos
	Mapa de despliegue

Los documentos/entregables generados en las diferentes etapas del ciclo de vida de desarrollo serán aprobadas por DIAN.

9. Proceso de Desarrollo - Metodología

La DIAN cuenta con suscripción a la plataforma Azure DevOps (DevOps-DIAN), sobre la cual debe realizarse la gestión del proyecto empleando metodologías ágiles ampliamente aplicadas en la industria del desarrollo de software.

Las fases que se describen a continuación corresponden a un entendimiento general de las instancias que deberá cumplir el desarrollo de la solución, con el fin de que la metodología se realice mediante el marco de desarrollo ágil y sea compatible con los procesos actuales de la entidad. Se deberán tener en cuenta estas fases y sus correspondientes entregables, sin embargo, el proveedor podrá proponer mejoras o ajustes metodológicos, los cuales serán revisados y aprobados o rechazados por la supervisión del contrato.

Los servicios contratados deberán ser prestados de acuerdo con el presupuesto asignado, las condiciones y especificaciones señaladas en el presente documento. El proveedor no podrá empezar a trabajar lo requerido y especificado en las historias de usuario sin que estas se encuentren previamente revisadas y aprobadas por el supervisor del contrato en la plataforma DevOps-DIAN, quien al revisar y aprobar velará que se cumpla con las funcionalidades, cantidades de horas trabajadas, plazos, recursos, hasta ejecutar el monto total del contrato.

9.1 Fases

El proveedor deberá ejecutar una fase de iniciación, en la cual contempla se realicen las siguientes actividades sin limitarse a las mismas:

FASES	ACTIVIDADES
FASE: INICIACIÓN	Visión del Proyecto. Definición del equipo del proyecto (equipo Scrum y otros roles). Lista priorizada de requerimientos a alto nivel. Presentación metodología de estimación de esfuerzo y aprobación por parte de DIAN Definición de épicas. Una épica corresponderá al entendimiento del sistema actual tanto técnico como funcional. Se deberán definir features dentro de cada épica. Definición de sprints (debe alinearse con la metodología de la DIAN en duración, estimación y medición de esfuerzos) Debe entregar un plan de fases o etapas con fechas y responsables.

Durante el desarrollo de la solución, se contempla que el proveedor ejecute las siguientes fases sin limitarse a las mismas:

FASES	ACTIVIDADES
FASE: PLANIFICACIÓN Y ESTIMACIÓN	Elaborar historias de usuarios (HU). Se deberá entregar, analizar, evaluar y profundizar acerca de las características técnicas y funcionales de los requerimientos a implementar. Deberían incluir requerimientos no funcionales. Estimar y aprobar historias de usuario. Elaboración de tareas. Elaboración de la lista de pendientes de SPRINT. Elaboración de la arquitectura de la solución y su aprobación

FASE: DESARROLLO E IMPLEMENTACIÓN	Análisis y Diseño (AyD): Elaboración de artefactos/diagramas de diseño requeridos por las HU del Sprint, entre otros: Diagramas de secuencia, diagramas de clases, diagramas de Entidad-Relación, diagrama de Despliegue. Justificación de cada componente o artefacto a crear. Elaboración de prototipos según necesidad. Preferir que sean reutilizables. Requerimientos y consideraciones especiales. Diseño de plan de pruebas técnicas y funcionales. Codificación de la solución. Versionamiento y código documentado. Ejecución de pruebas unitarias y pruebas integrales. Despliegues en ambiente de pruebas técnicas. Ejecución de pruebas técnicas funcionales. Ejecución de pruebas técnicas no funcionales y análisis estático del software. Automatización de pruebas Stand up Daily. Lista de pendientes de producto. Corrección de bugs. Despliegues en ambiente de pruebas funcionales.
FASE: REVISIÓN Y RETROSPECTIVA	Demostración y validación del SPRINT. Retrospectiva del SPRINT.

Para la estimación del esfuerzo de los desarrollos, la DIAN presentará al Proveedor la especificación funcional de alto nivel y el concepto de negocio especializado por cada requerimiento que se requiera ejecutar. El resultado de la estimación de esfuerzo de los anteriores componentes, serán propuestos a la supervisión, y de ser aceptado por la DIAN, constituirán el valor único a reconocer por la entidad una vez aceptados los entregables respectivos.

El proveedor debe entregar sus desarrollos desde ambiente de desarrollo de la DIAN, a partir de allí se promueve a pruebas (CI/CD) y posteriormente a producción, la asignación de ambientes de trabajo será la establecida en las cláusulas contractuales. Y en todos los casos el código fuente debe ser entregado en el DevOps-DIAN cumpliendo con lo requerido en el numeral “2.1 Normatividad y documentación interna” de este documento.

Las pruebas se realizarán en el ambiente de pruebas provisto por la DIAN. El proveedor entrega en ambiente de desarrollo DIAN, y la actualización del ambiente de pruebas DIAN se realizará a partir del ambiente desarrollo DIAN, esto en cada sprint.

El Proveedor probará el correcto funcionamiento de los componentes entregados, adicionalmente se validará que la documentación entregada esté completa y conforme a lo comprometido, y se validará que las fuentes de software instaladas hayan sido entregadas, correctamente versionadas, que el código entregado esté con comentarios y en su construcción se haya hecho uso de las buenas prácticas sugeridas por la DIAN y aquellas aportadas en la metodología del Proveedor. El proveedor deberá utilizar el repositorio GitHub de la DIAN para realizar análisis estático del software.

El código fuente debe ser entregado en el GitHub de la DIAN.

9.2. Requerimientos No Funcionales.

En proyectos de ingeniería de software, los requisitos no funcionales, también conocidos como atributos de calidad, son especificaciones que definen los atributos operativos de un sistema, en lugar de sus comportamientos específicos. A diferencia de los requisitos funcionales, que describen lo que un sistema debe hacer, los requisitos no funcionales describen cómo el sistema realiza ciertas funciones en condiciones específicas. Los requisitos no funcionales generalmente incrementan el costo, ya que requieren un esfuerzo especial durante la implementación. Sin embargo, al definirlos en detalle al inicio del proyecto, se pueden evaluar adecuadamente cuando el costo de su impacto en las decisiones de diseño posteriores es comparativamente bajo. [2]

Categoría	Atributo de calidad	Descripción
Requisitos operativos	Disponibilidad	Tiempo de actividad del sistema y accesibilidad para los usuarios.
	Integridad de los datos	Precisión y consistencia de los datos a lo largo de su ciclo de vida.
	Recuperación ante desastres y continuidad del negocio	Determinar los requisitos del sistema para la recuperación ante desastres y la continuidad del negocio, incluidos los procedimientos de respaldo y recuperación y las pruebas de recuperación ante desastres.
	Fiabilidad (Reliability)	Capacidad del sistema para mantener la funcionalidad en diferentes condiciones y escenarios de falla.
Requisitos de desempeño	Capacidad	Carga o volumen máximo que el sistema puede manejar dentro de criterios de rendimiento especificados.
	Actuación (Performance)	Defina los tiempos de respuesta esperados, el rendimiento y el uso de recursos de la solución.
	Escalabilidad	Determinar cómo el sistema manejará mayores cargas de usuarios o conjuntos de datos más grandes a lo largo del tiempo.
Requisitos de seguridad y cumplimiento	Cumplimiento	Cumplimiento de estándares y requisitos legales, reglamentarios y de la industria.
	Privacidad	Protección de información sensible y cumplimiento de la normativa de privacidad.
	Seguridad	Establecer los requisitos de seguridad del sistema, como autenticación, autorización, cifrado y cumplimiento de las regulaciones industriales o legales.
	Sostenibilidad	Capacidad de operar durante un período prolongado minimizando el impacto ambiental y el consumo de recursos.
Mantenibilidad del sistema	Interoperabilidad	Capacidad de interactuar e intercambiar datos con otros sistemas o componentes.
	Mantenibilidad	Facilidad de modificar, actualizar y ampliar el software a lo largo del tiempo.

[2] https://microsoft.github.io/code-with-engineering-playbook/design/design-patterns/non-functional-requirements-capture-guide/

	Observabilidad	La capacidad de medir el estado interno y el rendimiento de un sistema en función de los resultados que genera, como registros, métricas y seguimientos.
	Portabilidad	Capacidad de ejecutar el software en diferentes plataformas, entornos y dispositivos.
Requisitos de experiencia del usuario	Accesibilidad	La solución debe ser utilizable por personas con discapacidad. Cumplimiento de los estándares de accesibilidad. Compatibilidad con tecnologías de asistencia.
	Internacionalización y localización	Adaptación del software para su uso en diferentes idiomas y culturas. Adaptación del software a las necesidades específicas de cada región o región.
	Usabilidad	Intuición, facilidad de aprendizaje y satisfacción del usuario con la interfaz del software.

Fuente: Manual de Fundamentos de ingeniería. Captura de requisitos no funcionales https://microsoft.github.io/code-with-engineering-playbook/design/design-patterns/non-functional-requirements-capture-guide/

RTO y RPO

En recuperación ante desastres, RTO y RPO son dos métricas clave. El RTO (Objetivo de Tiempo de Recuperación) es el tiempo máximo permitido para restaurar un sistema o aplicación después de una interrupción. El RPO (Objetivo de Punto de Recuperación) es la cantidad máxima de datos que una empresa puede permitirse perder en caso de un desastre. En resumen, el RTO se centra en el tiempo, mientras que el RPO se centra en la pérdida de datos.

El usuario funcional (también conocido como dueño del negocio o product owner) es quien entiende el impacto operativo de una interrupción o pérdida de datos. Por eso, solo él puede responder preguntas como:

- RTO: ¿Cuánto tiempo puede estar inactivo el sistema sin causar un daño crítico?
- RPO: ¿Cuántos datos se podrían perder sin afectar gravemente el negocio?

Conjunto de requerimientos no funcionales a modo de ejemplo

A continuación, se presenta un conjunto de requerimientos no funcionales a modo de ejemplo, aplicables a un aplicativo dentro de la entidad. Esta lista no es exhaustiva y debe ser utilizada únicamente como guía de referencia. Cada proyecto debe identificar y documentar sus propios requerimientos, tanto funcionales como no funcionales, según su contexto y necesidades específicas.

Categoría	Subcategoría	Requisito
Usabilidad Describe cuán fácil es para los usuarios aprender,	Aprendibilidad	Un usuario nuevo debe poder completar un flujo o solicitud en menos de 10 minutos sin capacitación formal.
	Intuición	El sistema debe permitir completar el flujo o solicitud sin pasos ambiguos ni instrucciones externas.
	Ayuda contextual	El sistema debe incluir ayudas visuales en español (tooltips, guías, preguntas frecuentes).

entender y usar el sistema.	Consistencia	Interfaces coherentes en estilo, flujo y terminología en todas las pantallas, siguiendo el estándar grafico de la DIAN. Adicionalmente debe ser posible cambiar los logos y estilos de una forma fácil. El estándar grafico también incluye los reportes.
	Accesibilidad	Cumplimiento de WCAG 2.1 nivel AA para personas con discapacidades.
	Portabilidad UI	La interfaz debe funcionar correctamente en escritorio y móvil (responsivo).
	Compatibilidad	Soporte para navegadores modernos (Chrome, Firefox, Edge, Safari, últimas 2 versiones).
	Retroalimentación	Mensajes claros ante errores, validaciones y éxito (ej. campo faltante, validación de RUT).

Categoría	Subcategoría	Requisito
Rendimiento y escalabilidad Relaciona la capacidad del sistema para responder rápidamente y crecer ante mayor carga.	Tiempo de respuesta	Las operaciones deben responder en menos de 2 segundos bajo carga normal.
	Capacidad	Procesar al menos 2,500 solicitudes mensuales con capacidad de escalar hasta 10,000
	Escalabilidad	Escalable horizontalmente (más instancias) o verticalmente (más CPU/RAM).
	Carga concurrente	Soportar al menos 50 usuarios concurrentes sin degradación del servicio.
	Análisis de carga	Se deben realizar pruebas de carga para simular picos de 5 veces el promedio mensual.

Categoría	Subcategoría	Requisito
Seguridad Asegura que la información y los procesos estén protegidos contra accesos no autorizados.	Autenticación	Se debe usar el sistema de identidad de la DIAN.
	Autorización	RBAC - Roles bien definidos: ciudadanos, funcionarios, revisores. Utilizando el sistema de identidad de DIAN.
	Cifrado	HTTPS/TLS 1.2 o superior para todo el tráfico. Datos sensibles cifrados en base de datos (AES-256).
	Trazabilidad	Registro de todas las acciones del usuario (registro de auditoría).
	Protección de datos	Cumplimiento de normativa local (Habeas Data, Ley 1581 en Colombia, GDPR si aplica). Ver Anexos: • Manual de políticas y lineamientos de seguridad de la información. MN-IIT-0072 (DIAN) • Manual de políticas y lineamientos de protección de datos personales MN-IIT-0062 (DIAN)

Categoría	Subcategoría	Requisito
Fiabilidad Se refiere a la capacidad del sistema de mantener operaciones sin fallos.	Tolerancia a fallos	El sistema debe seguir operando parcialmente ante fallos de componentes (ej. base de datos secundaria en modo lectura).
	Disponibilidad	99.9% de disponibilidad mensual (máx. 43 minutos de inactividad permitida por mes).
	Durabilidad	No debe haber pérdida de información ante fallos; uso de transacciones para consistencia.
	Pruebas de resiliencia	Deben ejecutarse pruebas ante caída de servicios críticos como autenticación, almacenamiento o colas de mensajes.

Categoría	Subcategoría	Requisito
Mantenibilidad Capacidad del sistema para ser actualizado, depurado y mejorado fácilmente.	Modularidad	Separación clara de responsabilidades (ej. backend, frontend, autenticación, pagos).
	Documentación	Código y APIs documentados. Manuales técnicos y de usuario actualizados.
	Gestión de errores	Manejo centralizado de errores con logs legibles y almacenados de forma segura.
	Actualización	Las actualizaciones deben poder realizarse sin afectar la disponibilidad del sistema.
	Observabilidad	Monitoreo mediante Prometheus, Grafana, ELK u otras herramientas para trazabilidad.

Categoría	Subcategoría	Requisito
Portabilidad Capacidad del sistema de funcionar en distintos entornos y tecnologías.	Entorno flexible	El sistema debe poder ser desplegado en contenedores (ej. Docker/Kubernetes). Arquitectura de transición AKS o EKS y posteriormente OpenShift.
	Independencia	Evitar dependencias específicas del sistema operativo o proveedor de nube. Seguir arquitectura de referencia, y para transición solicitar aval del equipo de arquitectura.
	Migración fácil	Capacidad de migrar datos e instancias con mínimo esfuerzo hacia otros entornos (QA → Producción).

Categoría	Subcategoría	Requisito
Interoperabilidad Capacidad del sistema para interactuar con	Seguridad	Autenticación y autorización utilizando el sistema de identidad de la DIAN.
	Integración	API RESTful seguras y bien documentadas para integrarse con otros servicios (RUT,Arquitectura,SINOT notificaciones, etc.).
	Estándares	Uso de estándares como JSON, XML, JWT, OAuth2.

otros sistemas o servicios.	Conectividad	Tiempos de respuesta aceptables para integraciones externas (timeout máximo 5 segundos, reintentos si aplica). Estrategias de tolerancia fallos

9.2. Pruebas Unitarias

- Requisito: Se deberá entregar el código fuente con pruebas unitarias implementadas.
- Cobertura mínima: Se recomienda al menos un **80%** de cobertura de código para componentes críticos.
- Herramientas: JUnit (java), NUnit (.net), xUnit (.net), Jest (javascript), Mocha (javascript).
- Evidencias: Reporte de cobertura de código (por clase, método y línea).

9.3. Pruebas de Integración

- Requisito: Deben desarrollarse pruebas de integración entre los componentes del sistema o entre servicios.
- Objetivo: Validar el correcto funcionamiento del sistema como un conjunto.
- Entregables esperados: Scripts de prueba, datos de prueba y evidencia de ejecución exitosa.

9.4. Pruebas funcionales

El proveedor debe entregar a DIAN todos los artefactos que le permitan a la DIAN repetir de manera autónoma la automatización de pruebas en ambiente de pruebas DIAN, las pruebas funcionales se deben automatizar.

Para aquellos flujos y funcionalidades que no puedan ser automatizados se realizaran pruebas manuales. Estas pruebas incluirán la validación de la interfaz de usuario, la usabilidad, y la verificación de escenarios específicos de negocio que requieran intervención humana para su validación, las cuales deben estar documentadas y aprobadas en los formatos establecidos en el sistema de gestión de la Entidad (uso de DevOps de la DIAN).

Siempre que se posible se deberán automatizar pruebas de regresión.
Herramientas: Selenium, Cypress, Postman, RestAssured.

9.5. Automatización de pruebas para API REST

Para una API REST, se pueden automatizar varios tipos de pruebas para asegurar calidad, funcionalidad y rendimiento.

Pruebas funcionales
- Validar que cada endpoint responde correctamente con los datos esperados.
- Verificar métodos HTTP (GET, POST, PUT, DELETE).
- Comprobar validaciones de entrada, formatos y códigos de respuesta HTTP.
- Pruebas de flujos completos (por ejemplo, crear un recurso, actualizarlo, eliminarlo).

Pruebas de regresión
- Repetir pruebas funcionales automáticamente después de cambios para detectar errores.

Pruebas de integración
- Verificar interacción correcta entre varios servicios o componentes mediante APIs.
- Validar la integración con bases de datos o servicios externos.

Pruebas de rendimiento / carga
- Medir tiempos de respuesta y comportamiento bajo carga.
- Identificar cuellos de botella y límites de escalabilidad.

Pruebas de seguridad
- Validar autenticación y autorización (tokens, roles, permisos).
- Detectar vulnerabilidades comunes como inyección, XSS, etc.

Pruebas de contratos
- Validar que la API cumple con el contrato definido (por ejemplo, OpenAPI/Swagger).
- Detectar cambios que rompan compatibilidad con clientes

9.6. Pruebas técnicas no funcionales

Las pruebas técnicas no funcionales deben poder ser reproducibles por la DIAN y deben lanzarse de manera automática.

Para la realización de este tipo de pruebas, en el caso que aplique, el proveedor deberá tomar una medición previa de cada requerimiento no funcional, empleando la línea base del código que indique la DIAN, que permita realizar una comparación posterior contra las mediciones empleando las versiones del software que haya modificado con motivo de la ejecución del proyecto. El proveedor deberá realizar los ajustes requeridos para mantenerse en los valores aceptables por la entidad.

El proveedor deberá utilizar las herramientas señaladas en el lineamiento “Lineamientos y estandarización para la gestión de repositorios de código fuente y buenas prácticas en el desarrollo de automatización de pruebas de la DIAN”, ver numeral 2.1 “Normatividad y documentación interna” del presente documento.

El proveedor deberá realizar el levantamiento de requerimientos no funcionales (que incluye métricas, como por ejemplo tiempos máximos de respuesta permitidos, tasa de transferencia, tiempo máximo de procesamiento bajo carga, usuarios concurrentes, latencia máxima permitida) que se convierten en criterios de aceptación.

Las pruebas técnicas no funcionales deberán contemplar:

- Carga: Pruebas para verificar el correcto funcionamiento del sistema ante un alto número de peticiones, o cantidad usuarios concurrentes.
- Rendimiento: Pruebas para verificar el tiempo de respuesta del sistema.
- Volumen: Pruebas que permitan verificar que el sistema pueda cargar y generar archivos con grandes cantidades de datos en las funcionalidades que aplique.
- Esfuerzo: Realización de pruebas que sobrecarguen el sistema durante un tiempo y posteriormente permitan observar una correcta recuperación y funcionamiento posterior.
- Seguridad: Pruebas que permitan observar el comportamiento del sistema ante accesos no autorizados, Denegación del Servicio (DoS), entre otros.

- Se deben atender los lineamientos de la OSI relacionados con seguridad de la información
- Usabilidad: Pruebas que miden la facilidad de uso y la accesibilidad del sistema para los usuarios, incluyendo su capacidad para navegar por el sistema, realizar tareas y encontrar información.
- Escalabilidad: Pruebas que miden la capacidad del sistema para crecer/decrecer y adaptarse a medida que aumenta la demanda, como el número de usuarios o la cantidad de datos procesados.
- Disponibilidad: Pruebas que miden la capacidad del sistema para estar disponible y funcionar correctamente en todo momento, incluyendo la recuperación de fallos y la redundancia de hardware y software.
- Compatibilidad: Pruebas que evalúan la capacidad del sistema para funcionar correctamente en diferentes plataformas, navegadores y dispositivos.
- Los navegadores que deben soportar las soluciones desarrolladas para la DIAN son Chrome, Edge, Safari, Firefox en versiones que soporten características requeridas por la implementación.
- Mantenibilidad: Pruebas que miden la capacidad del sistema para ser mantenido y actualizado fácilmente, incluyendo el modularidad, la legibilidad del código y la facilidad de prueba y depuración.

Las pruebas de carga y estrés deben ejecutarse en ambiente de pruebas DIAN. El proveedor debe entregar a DIAN todos los artefactos que le permitan a la DIAN repetir de manera autónoma la automatización de pruebas en ambiente de pruebas DIAN.

9.7. Pruebas de Usabilidad

Requisito: El proveedor deberá realizar pruebas de usabilidad básicas sobre las interfaces de usuario.
Criterios: Facilidad de navegación, legibilidad, tiempos de respuesta percibidos, accesibilidad básica (WCAG nivel AA mínimo si aplica).
Entregables: Informe de hallazgos y ajustes realizados.

9.8. Pruebas de Rendimiento

Requisito: Las aplicaciones críticas o de alta concurrencia deberán superar pruebas de carga y estrés.
Herramientas: JMeter, Gatling, k6, locust.
Indicadores esperados:

- Tiempo de respuesta aceptable (< 2 segundos en el 95% de los casos).
- Capacidad de soportar el volumen estimado de usuarios concurrentes.
- Reporte con gráficas e interpretación de resultados.

Criterios de aceptación sugeridos para pruebas de carga y stress, y ANS del servicio de interoperabilidad con terceros (ejemplo bancos)

Métrica	Valor aceptable
Tiempo de respuesta (latencia) Tiempo promedio que tarda la API en responder	< 300 ms para el 95% de las peticiones (P95)
Throughput	througput > 6000 peticiones por segundo
Porcentaje de errores Proporción de respuestas con códigos 4xx y 5xx	< 0.1% de errores 5xx

	< 1% de errores 4xx (esperados por errores del cliente)
Tolerancia a inyecciones y ataques API protegida contra OWASP API Top 10	0 vulnerabilidades conocidas; cumplimiento verificado en pentesting
Compatibilidad e interoperabilidad: -Especificaciones de contrato (OpenAPI/Swagger) -Versionado de API	-Publicación de contrato con versión controlada 100% de los endpoints documentados y validados automáticamente -Uso de versionado semántico (v1,v2..) y backward compatibility para cambios no disruptivos
Documentación api	Se deberá diligenciar el formato FT-IIT-2715 Especificación técnica de interoperabilidad Para cada operación debe incluir ejemplos de ejecución exitosa y de fallo También se debe documentar cada uno de los campos de entrada y de salida
Trazabilidad de transacciones Cada llamada debe estar trazada con un ID único y registrarse de extremo a extremo	100% de las llamadas con correlation-id y logs estructurados
Auditoría Registro de cada acceso y operación sensible	100% de acciones relevantes auditadas
Política de rate limiting, throttling	La API debe implementar una política de rate limiting basada en identificador de cliente (API key o JWT), retornando código HTTP 429 con encabezados estándar cuando se supere el límite. Los valores de límite deben ser configurables y auditables

9.9. Pruebas de Seguridad

Requisito: Se deberá aplicar un análisis de seguridad básico (DAST/SAST) y pruebas de validación de controles.
Herramientas: OWASP ZAP, SonarQube (con reglas OWASP activadas), Checkmarx.
Entregables esperados:

- Reporte de vulnerabilidades detectadas.
- Evidencia de corrección o justificación de aceptación de riesgo.

9.10. Análisis Estático de Código y Análisis de dependencias

Se debe realizar análisis de código estático y dinámico haciendo uso de la herramienta GitHub Advanced Security, en caso de detectarse vulnerabilidades de nivel crítico, alto o medio, se deben corregir para nuevamente analizar y lograr el nivel de madurez de seguridad requerido por la entidad, lo anterior previo el paso a producción.
Debe realizarse análisis de dependencias usando las funcionalidades de GitHub.

Obligatorio: Todo el código entregado deberá superar el análisis estático sin errores críticos ni bloqueantes aplicado sobre los repositorios de la entidad.
Herramienta: GitHub Advanced Security.
Umbrales mínimos:
0 vulnerabilidades críticas o bloqueantes.
Máximo 10% de código duplicado.
Máximo 5% de deuda técnica en componentes principales.

GitHub Advanced Security es un conjunto de características de seguridad para repositorios que incluye:

- Code scanning (análisis estático de código con CodeQL)
- Secret scanning (detección de secretos o credenciales en el código)
- Dependabot alerts (alertas de vulnerabilidades en dependencias)

Se deberá realizar análisis de código estático, detección de secretos o credenciales en el código y alertas de vulnerabilidades en dependencias

9.11. Revisión de Complejidad y Patrones de Diseño

Con el fin de asegurar la calidad, mantenibilidad y alineación con los lineamientos arquitectónicos de la DIAN, el proveedor deberá realizar una revisión técnica del código fuente que permita identificar niveles de complejidad innecesarios, prácticas de codificación deficientes y desviaciones frente a los patrones de diseño definidos por la entidad.

Todos los módulos de código entregados deberán ser analizados mediante herramientas de análisis estático para medir la complejidad y calidad del código, se utilizarán y entregará un informe con el resultado de las métricas de complejidad ciclomática de código, duplicación de código, y adherencia a estándares definidos en la arquitectura.

- Se considerará aceptable una complejidad ciclomática promedio por método menor o igual a 10[3].
- No se aceptarán clases con más de 500 líneas ni métodos con más de 50 líneas.
- La duplicación de código no podrá superar el 3% por módulo.
- Evitar el uso de anti-patrones o prácticas que dificulten la extensibilidad y el desacoplamiento.
- El proveedor debe garantizar que todo el código entregado cumpla con los principios de Clean Code, SOLID, y los lineamientos de codificación establecidos en los anexos del presente documento.

El reporte debe ser el resultado del análisis ejecutado por herramientas de análisis de código estático como SonarQube, GitHub Advanced Security, PMD, Checkstyle o herramientas equivalentes aprobadas por la DIAN.

Se deberán configurar reglas específicas para:

[3] Esta metríca se alinea a lo definido por McCabe, creador de la métrica - IEEE Transactions on Software Engineering, 1976 y se alinea con las métricas definidas en estándares como SEI CERT Coding Standards https://wiki.sei.cmu.edu/confluence/display/java/MET00-J.+Maintain+low+cyclomatic+complexity, ISO/IEC 25010:2011 https://iso25000.com/index.php/en/iso-25000-standards/iso-25010, NASA Software Assurance Technology Center (SATC) y reglas por defectos en herramientas como SonarQube https://rules.sonarsource.com/java/RSPEC-3776.

- Complejidad ciclomática.
- Tamaño máximo de clases y métodos.
- Niveles de duplicación.
- Uso de convenciones de codificación y estructura.

El uso de patrones de diseño será validado a través de:

- Revisión de código por parte del equipo de arquitectura o líderes técnicos asignados por la DIAN.
- Checklist de validación de patrones definidos por proyecto.
- Análisis de dependencias para validar acoplamientos indebidos.

Criterios de aceptación mínimos:

Métrica	Umbral aceptable
Complejidad ciclomática por método	≤ 10
% de duplicación de código	≤ 5% por módulo
Tamaño máximo de clase	500 líneas
Tamaño máximo de método	50 líneas
Uso de patrones de diseño autorizados	100% en componentes relevantes
Anti-patrones	0 (deberán ser refactorizados)

Entregables esperados:

- Reporte de análisis de complejidad generado por la herramienta correspondiente.
- Evidencia de revisión de patrones y cumplimiento de arquitectura (puede ser checklist o documento de validación técnica).
- Actas o comentarios de revisión de código (pull request, revisión arquitectónica).
- Informe de refactorizaciones aplicadas, si fueron necesarias.

Observaciones adicionales:

- Los hallazgos que excedan los umbrales definidos deberán ser corregidos antes de avanzar al ambiente de pruebas funcionales o ser justificados y autorizados por la supervisión del contrato y/o el equipo de arquitectura.
- El proveedor debe garantizar que todo el código entregado cumpla con los principios de Clean Code, SOLID, y los lineamientos de codificación establecidos en los anexos del presente documento

Asimismo, se validará que el diseño del software implemente patrones de diseño aprobados institucionalmente, descritos en las guías de desarrollo definidas por la DIAN.

El incumplimiento de estos criterios podrá ser causal de rechazo del entregable, conforme a los términos contractuales establecidos.

9.12. Entregables de Pruebas

Cada ciclo o entrega del proyecto deberá venir acompañado de:

- Matriz de pruebas realizadas.
- Evidencia de ejecución (capturas, logs, reportes).
- Plan de pruebas y casos utilizados.
- Informe de errores encontrados y su resolución.
- Reportes de análisis estático (y dinámico si aplica).

Se deberá entregar lo siguiente para pruebas funcionales automatizadas de manera que sea posible su reproducción:

- El código o script de la prueba automatizada
- El entorno o configuración necesaria: Detalles sobre el entorno donde se ejecuta la prueba (versiones de software, dependencias, variables de entorno, datos de prueba, configuración de red, etc.).
- Datos de entrada y condiciones previas: Información sobre qué datos usar y cómo preparar el sistema antes de ejecutar la prueba.
- Instrucciones claras para ejecutar la prueba: Comandos, herramientas o scripts para correr la prueba paso a paso.
- Resultados esperados y criterios de éxito: Para validar que la prueba pasó correctamente o identificar fallos.
- Automatización de entorno (en los casos en que aplique): La prueba se deberá ejecutar en un entorno automatizado y reproducible (por ejemplo, contenedores Docker, máquinas virtuales o pipelines CI/CD) para asegurar consistencia.

9.13. Verificación y Aceptación

Los entregables serán revisados por el equipo técnico de la entidad. El incumplimiento de los requisitos anteriores podrá dar lugar a:

- Solicitudes de corrección
- Rechazo de la entrega
- Penalizaciones conforme al contrato

9.14. Gestión de Defectos y Corrección

Todos los defectos encontrados durante las pruebas serán gestionados a través del sistema de seguimiento de incidentes, integrado con la plataforma de DevOps utilizada. El proveedor debe realizar pruebas de regresión, para asegurar que los cambios recientes en el código no hayan introducido errores en funcionalidades existentes, y dejar evidencia de estas. Se deberán realizar pruebas unitarias, de integración y de componentes en ambiente de desarrollo y en ambiente de pruebas, pruebas de regresión.

El proveedor debe generar, ejecutar y entregar la batería de pruebas funcionales junto con su automatización. La DIAN debe poder ejecutar esta automatización de manera autónoma, también será posible hacer pruebas adicionales por sus funcionarios.

Los bugs reportados por los usuarios funcionales deberán ser atendidos por el proveedor cumpliendo con los ANS establecidos en el contrato y, así mismo, disponerlos para que sean probados nuevamente.

El proveedor dará prioridad a aquellos bugs bloqueantes para poder agilizar la terminación de los escenarios de prueba correspondientes, esto de acuerdo con la prioridad que se establezca de manera consensuada entre el proveedor y la DIAN.

FASES	ACTIVIDADES
FASE: PRUEBA DE ACEPTACIÓN DE USUARIO (UAT)	Selección de casos de prueba UAT y adición de escenarios de prueba requeridos. Ejecución de pruebas UAT, carga de evidencias y registro de Bugs en el DevOps de la DIAN. Emisión de formato de aceptación para cada Historia de Usuario.

El Proveedor deberá tener presente que, durante las pruebas de aceptación, deberá contar con los profesionales encargados de las actividades de pruebas. Estas actividades incluyen:

- Documentación de incidentes, problemas y situaciones presentadas.
- Acompañamiento a los usuarios funcionales durante la ejecución de las pruebas de aceptación.
- Corrección de bugs reportados por los usuarios funcionales.
- Elaboración de los informes de pruebas.

Una vez probada la funcionalidad, la DIAN, podrá aceptar los desarrollos para ser presentados en el Comité de Control de Cambios o podrá devolver los desarrollos o documentación para ajustes. Se diligenciará acta de los resultados de las pruebas y en caso de devolución se hará constar las razones de la devolución, las acciones que se ejecutarán y las fechas afectadas en el plan de trabajo. La documentación técnica requerida para presentar a Comité de Cambios (autorización para paso a producción) debe ser generada por el proveedor. La documentación técnica será revisada por la Subdirección de Soluciones y Desarrollo, quienes verificaran costeo, que lo desarrollado cumple con la arquitectura y lineamientos definidos.

9.15. Puesta en producción

El despliegue de aplicaciones en el ambiente de producción se debe realizar siguiendo los pasos que se describan en el manual de instalación del software, que el Proveedor deberá entregar, dicho manual debe cumplir con los lineamiento de la Dirección de Gestión de Innovación y Tecnología o quien hagas sus veces, solo actuará como ejecutor de los pasos descritos y en caso de presentarse errores al momento de la ejecución, el manual deberá contener indicaciones de cómo se deben solucionar los errores presentados, se debe tener en cuenta que el despliegue debe ser automático (IaC, ApiOps).

Para el caso de Factura Electrónica, es necesario seguir el lineamiento y el procedimiento establecidos en los documentos anexos al RFC correspondiente. Esto con el fin de garantizar la calidad del despliegue, asegurar la disponibilidad de todos los insumos y documentación necesarios, y facilitar una transición adecuada hacia el ambiente de producción.

Ver anexos:

- Anexo RFC Factura Electrónica.
- Anexo Documento de Caracterización Despliegues en Azure Factura Electrónica - RFC

Una vez instalado y validado el correcto funcionamiento del sistema en producción se procederá a diligenciar acta de aceptación que deberá ser firmada por el Proveedor, la supervisión del contrato y el usuario líder. En caso de que sea necesario devolver la versión, el Proveedor trabajará de manera colaborativa para ejecutar las acciones necesarias dejando el sistema en el mismo estado que se encontraba antes de la instalación, dejando acta y evidencia de las labores ejecutadas.

Para garantizar la estabilidad de las nuevas funcionalidades desarrolladas y/o implementadas, la DIAN solo aprobará el paso a producción del producto cuando el Proveedor cumpla condiciones como:

- Que los nuevos desarrollos estén dispuestos en el ambiente de desarrollo y pruebas de DevOps-DIAN para su revisión y aprobación por parte del equipo verificador de la entidad y de acuerdo con el procedimiento establecido para despliegue de servicios en producción.
- Que se haya realizado la entrega de los programas fuentes de los aplicativos en la herramienta de gestión de código fuente dispuesta por la DIAN en la plataforma DevOps-DIAN (GitHub) en las ramas dispuestas por la DIAN.
- Que se haya realizado la entrega de los manuales técnicos y de usuario en medio electrónico y archivos editables cada vez que se implementen modificaciones o actualizaciones. Junto con los manuales técnicos el proveedor debe entregar las fuentes de los diagramas generados en Enterprise Architect versión 17 en español.
- Que la DIAN pueda solicitar al Proveedor, la realización de las repeticiones, correcciones y ajustes pertinentes, cuando los productos y/o servicios no se desarrollen o no se obtengan de acuerdo con los procedimientos aprobados, o no satisfagan los criterios de completitud, calidad y eficiencia en la ejecución establecidos.
- Que todos los entregables, documentos y/o informes definidos en el proyecto estén aprobados por el supervisor del contrato.
- Que se haya generado formato de aceptación de pruebas de todas las historias de usuario involucradas en la entrega.
- Que se hayan generado las certificaciones de pruebas técnicas realizadas.

9.16. Estabilización

Se debe proporcionar soporte técnico y mantenimiento a las nuevas funcionalidades después de su implementación para garantizar su funcionamiento adecuado y resolver cualquier problema que pueda surgir.

El proveedor deberá colocar el personal que requiera para cumplir con los ANS durante las fases de estabilización, soporte, mantenimiento y garantía.

El proveedor deberá alinearse con el modelo de soporte de la DIAN (modelo operativo y herramientas), es decir, su soporte deberá organizarse para que procedimentalmente opere con el funcionamiento de los equipos de soporte y desarrollo de la DIAN.

El mantenimiento y soporte será para el sistema desarrollado o los desarrollos que se realicen, no es para todo el sistema, sin embargo, si el desarrollo realizado afecta o modifica parte del sistema anterior o ya en funcionamiento, también deberá darse soporte a esa sección.

El proveedor debe actualizar la documentación si producto de la estabilización es necesario modificar el código fuente, deberá actualizar toda documentación técnica y funcional a la que haya lugar, también deberá actualizar la documentación relacionada con la gestión de problemas.

10. Garantía

La garantía del desarrollo empieza a contar una vez finalizado el contrato, el número de meses será lo establecido en el contrato. El proveedor solucionará cualquier defecto relacionado con calidad de los artefactos (documentos, código) generados en cualquiera de las etapas del ciclo de vida de desarrollo, siempre que le sean atribuidas.
Estas correcciones las realizará el proveedor sin costo para la DIAN.

El proveedor debe actualizar la documentación si producto de la garantía es necesario modificar el código fuente, deberá actualizar toda documentación técnica y funcional a la que haya lugar, también deberá actualizar la documentación relacionada con la gestión de problemas.

11. Capacitación y transferencia del conocimiento

El proveedor debe generar un plan de capacitación y/o transferencia de conocimiento en donde se detalle los contenidos a tratar, intensidad horaria, materiales de apoyo, ejercicios prácticos (talleres), encuesta de satisfacción, y la evaluación a realizar antes, durante (por módulos), y después de la misma.

El proveedor deberá entregar todos los materiales relacionados con la capacitación y/o transferencia de conocimiento, incluyendo videos, manuales, presentaciones y cualquier otro recurso relevante, con el objetivo de conformar una biblioteca digital dentro de la DIAN para la gestión continua del conocimiento. Estos materiales deberán ser entregados en formatos accesibles y reutilizables, garantizando su disponibilidad para futuras consultas y actualizaciones dentro de la entidad.

El plan, el material y los manuales para utilizar en las capacitaciones deben ser revisados y aprobados por parte de la Supervisión antes de realizar las capacitaciones, de requerirse algún ajuste el proveedor deberá desarrollar dichos cambios hasta la aceptación por parte de la entidad.

El proveedor deberá crear laboratorios y talleres prácticos que simulen escenarios del día a día, permitiendo a los usuarios resolver problemas reales y adquirir experiencia directa con la solución, sin que represente costo adicional para la DIAN.

El proveedor deberá proveer instructores certificados o expertos en el software, que tengan pedagogía básica.

Si los resultados de la capacitación/entrenamiento no cumplen con los objetivos establecidos basados en las evaluaciones de desempeño y las encuestas de satisfacción aplicadas, no alcanzan un nivel mínimo del 80%, el proveedor deberá implementará estrategias más efectivas y sesiones de refuerzo hasta superar el 80%, para lograr una mayor efectividad en el aprendizaje, sin costo adicional para la DIAN.

La intensidad horaria para cada capacitación y/o transferencia será concertada entre el proveedor y el supervisor del contrato, por parte de la DIAN participará un experto del área de uso y Apropiación

que se encargará de acordar los horarios y temáticas con aprobación de los subdirectores o expertos en el tema). Las sesiones que realizar no deberán superar los 90 minutos cada una.

El proveedor deberá entregar los materiales de la capacitación y/o transferencia, incluyendo los videos de las grabaciones de las capacitaciones en alta definición (1920x1090 HD formato MP4) para conformar una biblioteca digital al interior de la DIAN para gestión del conocimiento.

Esta se debe ser ejecutada por el Proveedor, con apoyo de la DIAN. El Proveedor, deberá entregar los manuales y la información necesaria para la elaboración de los materiales de capacitación, la cual incluye:

Capacitación funcional: A las áreas usuarias correspondientes y a los profesionales que la Dirección de Gestión de Innovación y Tecnología determine.

Capacitación técnica. A los funcionarios que la Dirección de Gestión de Innovación y Tecnología determine (por ejemplo, desarrolladores, ingenieros de pruebas, seguridad, arquitectura, administración técnica).

Dentro de la capacitación técnica se debe incluir tratamiento errores conocidos, resolución de problemas, lecciones aprendidas.

El Proveedor deberá aportar las evidencias de dichas capacitaciones a la Supervisión del Contrato: Listados de asistencia, presentaciones de la capacitación, manuales y demás información requerida.

Adicionales:

- Rutas de Aprendizaje Personalizadas
- Flexibilidad en las Sesiones de Capacitación
- Moderación Activa y Apoyo Continuo en l sesiones de capacitación
- Análisis de posibles obstáculos que los usuarios puedan enfrentar durante la transición, identificando barreras técnicas y culturales.
- Desarrollo de actividades de sensibilización (charlas, talleres y lo que se considere necesario)
- Adopción de estrategias de capacitación según las diferentes modalidades de aprendizaje de los usuarios (presencial, virtual, híbrido)
- Se deberá garantizar un soporte permanente durante todo el proceso de transición, que podrá incluir asistencia técnica, tutorías personalizadas y, de ser posible, espacios como foros de dudas

Refuerzo y Mejora Continua

- EL consultor garantizará que el proceso de transferencia de conocimiento sea sostenible y eficiente a largo plazo:
- Los procesos de capacitación se ajustarán de manera continua, implementando nuevas metodologías y recursos pedagógicos según sea necesario para garantizar la máxima efectividad y asegurar que los usuarios continúen utilizando y dominando la solución tecnológica a largo plazo.

Metodología Participativa

Para evitar que los usuarios sientan que las capacitaciones son una clase tradicional, se adoptará una metodología participativa:

- Se iniciarán las sesiones con activadores como preguntas interactivas, dinámicas de reflexión o pequeños retos para enganchar a los participantes desde el inicio.
- Las sesiones incluirán dinámicas interactivas, como grupos de trabajo, encuestas en vivo y sesiones de retroalimentación, para mantener el interés y fomentar la participación.

12. Infraestructura tecnológica

El proveedor deberá contar con ambiente de desarrollo y pruebas en la nube definida al inicio del proyecto (Azure, AWS), en donde pueda realizar pruebas de concepto.

El proveedor deberá entregar un dimensionamiento de la solución que entrega a la DIAN y las consideraciones a tener en cuenta para el crecimiento futuro por la utilización de la plataforma, debe incluir calculadora para dimensionar los requerimientos de infraestructura, se deberá realizar para aprobación del diseño por parte de DIAN y se generará nueva versión para aprobación como requisito para llevar a comité de cambios y colocar en producción.

El proveedor una vez superado el desarrollo, deberá entregar el código fuente en el repositorio dispuesto por la DIAN y las configuraciones necesarias para realizar el despliegue continuo y el aprovisionamiento de los recursos, de acuerdo con los procedimientos establecidos por infraestructura DGIT.
La DIAN entregará documentación que se tenga se entrega en el estado en que se encuentre.

13. Ejecución contractual

El Proveedor, deberá llevar a cabo la ejecución contractual en sus instalaciones, sin embargo, si la entidad lo considera pertinente y sólo en caso de ser necesario, podrá requerir el traslado de personal del Proveedor, a las instalaciones de la DIAN para ejecutar el servicio contratado. Dicho traslado se debe coordinar con la Supervisión del Contrato.

El proveedor administra su equipo de trabajo, por lo tanto, la DIAN no es responsable de la modalidad de trabajo del equipo.

14. Otros Aspectos a Considerar.

- Cláusula de Cesión de Derechos Patrimoniales de Propiedad Intelectual sobre el Software Desarrollado. Todos los desarrollos realizados en ejecución del contrato, incluyendo el código fuente, código objeto, scripts, algoritmos, diseños, configuraciones, manuales y demás elementos que resulten del presente contrato, incluidos los que se generen durante el análisis, diseño, desarrollo, implementación, prueba y puesta en funcionamiento de la solución tecnológica contratada serán de propiedad exclusiva de la DIAN.
- El proveedor cede a título gratuito y definitivo los derechos patrimoniales de autor, esta cesión incluye, pero no se limita a, los derechos de reproducción, transformación, distribución, comunicación pública y cualquier otra modalidad de explotación, conforme a lo dispuesto en la Ley 23 de 1982, la Decisión Andina 351 de 1993 y demás normas que las modifiquen o sustituyan, sin limitación territorial ni temporal.

- Cualquier componente de terceros, deberá contar con licencia para uso a perpetuidad y aprobación previa de la DIAN. El proveedor deberá entregar el código fuente y suscribir el acta de cesión antes de finalizar el contrato.
- El proveedor deberá hacer entrega formal al equipo de tecnología de la DIAN, explicando todo el desarrollo y dejando por escrito lo necesario para adoptar su evolución y mantenimiento posterior.
- La DIAN deberá revisar y aprobar todo desarrollo y documentación/entregables.
- El proveedor deberá entregar todos los manuales de desarrollo y de uso para el desarrollo completo.
- El proveedor deberá actualizar los manuales de uso vigentes en la entidad para todo lo ajustado por él.
- La DIAN no cuenta con documentación técnica y el proveedor deberá revisar el código y realizar la ingeniería inversa con el fin de adquirir el conocimiento necesario para adelantar la implementación del proyecto.
- Si el proveedor define algún cambio de tecnología de datos o su diseño necesita algún tratamiento de datos, deberá asumirlo como parte del contrato.
- El proveedor deberá seguir los estándares de diseño de API y el marco de interoperabilidad de la DIAN.
- Cuando se trate de fábricas el costo de las suscripciones de las herramientas de desarrollo deben ser asumidos por ellos durante el proyecto y la garantía. Las herramientas de desarrollo deben ser aprobadas por la DIAN.
- Toda solución que se implemente para la DIAN deberá cumplir con los procesos clave de ITSM, incluyendo la gestión de incidentes, cambios y configuración, garantizando el uso de sistemas de seguimiento (tickets) y la adhesión a los acuerdos de nivel de servicio (SLAs) con tiempos definidos de respuesta y resolución.
  Para la etapa de pruebas se deberá utilizar el IT Service Management (ITSM) cuando se requiera formalidad, trazabilidad y control en la gestión de incidencias, cambios y calidad.
  Se deberá utilizar la herramienta ITSM adoptada por la DIAN