



Arquitectura referencial de software del Ministerio de Economía y Finanzas

Versión 1



2023

**Oficina General de tecnología de la Información
Oficina de Sistemas de Información**

HISTORIAL DE REVISIONES

Versión	Fecha	Cambios realizados	Elaborado por:	Revisado por:	Aprobado por:
1	05.05.2023	Versión inicial	Proyecto PE-005 del Plan de Gobierno Digital 2022-2024. Consultor Adolfo Escudero José Visalot Delfor Chacón	Julio Molina G. Agustín Roble Raúl Tapia	Eduardo Ibarra

INDICE

I INTRODUCCIÓN.....	4
II OBJETIVO.....	5
II.1 Objetivos Específicos	5
a) Mostrar las vistas arquitectónicas de sistema desde las perspectivas de los intereses o preocupaciones de los interesados del ministerio.....	5
b) Mostrar los puntos de vistas particulares para las vistas arquitectónica que se desarrolle.....	5
c) Mostrar los estilos o patrones arquitectónicos que actualmente se tiene definidos y operativos en los sistemas de información del MEF.....	5
III ALCANCE.....	5
IV GLOSARIO Y DEFINICIONES	5
V MARCO CONCEPTUAL.....	6
V.1 Modelo conceptual de la descripción de arquitectura.....	6
V.1.1 Identificación de las partes interesadas y de sus preocupaciones	6
V.1.2 Arquitectura y descripción de la Arquitectura.....	12
V.1.3 Puntos de vistas	14
V.1.4 Tipo de modelo.....	14
V.1.5 Vistas de arquitectura	14
V.1.6 Modelo arquitectónica.....	15
V.1.7 Vistas arquitectónicas.....	¡Error! Marcador no definido.
V.1.8 Estilos arquitectónicos referenciales.....	21
V.1.9 Capas Arquitectónicas.....	25
V.2 Documentación sobre la arquitectura referencial	27
V.2.1 Componentes de la arquitectura referencial	27
V.2.2 Documentación de apoyo a la arquitectura referencial	31
V.2.3 Documentación sobre estándares y lineamiento arquitectural	34

I INTRODUCCIÓN

La Oficina General de Tecnologías de la Información (OGTI) del Ministerio de Economía y Finanzas ha identificado la necesidad de facilitar instrumentos normativos y estandarizados de lineamiento que permita establecer un Marco de Referencia de Arquitectura de Software en el Ministerio.

La OGTI, de acuerdo a sus funciones establecidos en el “Texto Integrado del Reglamento de Organización y Funciones del Ministerio de Economía y Finanzas” y cuyo numeral 81 a) que indica que la OGTI formula, propone y evalúa las políticas de tecnología de información en el Ministerio, ha venido desarrollando este documento de referencia sobre la arquitectura de software en la que se pretende describir sus componentes principales, relaciones y su interacción entre sí.

Así mismo, el MEF ha formulado su Plan de Gobierno Digital para el período 2022-2024 estableciendo un conjunto de proyectos tecnológicos para adecuarnos y adoptar la Política de Gobierno y Transformación Digital en el MEF. En tal sentido, en dicho plan se encuentra el Proyecto “**P-005 Fortalecimiento e implementación de la arquitectura distribuida de desarrollo y operación segura de aplicaciones y sistemas de información**”, en la que como uno de sus objetivos específicos busca la masificación de la Arquitectura de Referencia de la OGTI, que será una directriz para el desarrollo de las aplicaciones informáticas en el MEF.

Mediante Decreto Supremo N° 029-201-PCM, se aprueba el Reglamento del DL N° 1412, que aprueba la Ley de Gobierno Digital y establece disposiciones sobre las condiciones, requisitos y uso de las tecnologías y medios electrónicos en el procedimiento administrativo, por lo que este documento se adecua a los principios rectores del Marco de la Arquitectura Digital del Estado Peruano en cuanto a la; neutralidad tecnológica, valor ganado, reusabilidad, automatización y seguridad de la información. Asimismo, mediante este documento, se pretende tener un enfoque respecto del Modelo de Referencia de Aplicaciones descrito en el referente documento normativo indicado.

Por lo tanto, el presente documento busca establecer soluciones generales para el cumplimiento de los requisitos técnicos y operativos inmerso en una solución y desarrollo de software, buscando el rendimiento máximo de sus componentes y seguridad. La arquitectura de las aplicaciones es particularmente importante porque la forma en que se construye un sistema afecta directamente su capacidad para satisfacer los llamados atributos de calidad del sistema.

Además de las características de calidad, la arquitectura del software también juega un papel fundamental en la gestión del desarrollo de software; una de las muchas estructuras que lo componen se enfoca en descomponer el sistema en componentes que serán desarrollados por individuos o grupos de individuos. Definir esta estructura de tareas es esencial para respaldar las tareas de planificación de proyectos de desarrollo de software. Por esta razón, los diseños arquitectónicos que se creen pueden reutilizar para construir diferentes sistemas, buscando la reducción de costos y mejorar la calidad, especialmente si tales diseños producen sistemas exitosos.

En resumen, lo que se pretende alcanzar con el presente documento es proveer una arquitectura referencial de las aplicaciones, sobre la base de la existente y con una visión de mejora de forma tal que provea patrones de diseño para las aplicaciones, así como inducir las prácticas para la integración y despliegues continuos de las soluciones de las aplicaciones y sistemas de información del MEF.

II OBJETIVO

El propósito principal de este documento es brindar orientación para el desarrollo de una arquitectura referencial de software que muestre los aspectos fundamentales de la arquitectura de un sistema de información o aplicación, a través de sus elementos y relaciones y en los modelos de su diseño.

II.1 Objetivos Específicos

- a) Mostrar las vistas arquitectónicas de sistema desde las perspectivas de los intereses o preocupaciones de los interesados del ministerio.
- b) Mostrar los puntos de vistas particulares para las vistas arquitectónicas que se desarrolle.
- c) Mostrar los estilos o patrones arquitectónicos que actualmente se tiene definidos y operativos en los sistemas de información del MEF.

III ALCANCE

El presente documento muestra, en el marco de la Norma *ISO/IEC 42010 Arquitectura de Sistemas y Productos Software*, las descripciones de la arquitectura de los sistemas de información, para ser tomados en cuenta en el desarrollo descriptivo de la arquitectura de los sistemas de información del MEF. También muestra una guía referencial arquitectónica de los sistemas de información y productos software que actualmente posee el Ministerio.

IV GLOSARIO Y DEFINICIONES

- a) **Arquitectura:** conceptos o propiedades fundamentales de un sistema en su entorno representado en sus elementos, relaciones, y en los principios de su diseño y evolución (ISO/IEC 42010).
- b) **Arquitectura Empresarial:** Es un conjunto completo de modelos, diagramas, documentos y especificaciones que describen cómo una organización funciona en términos de procesos de negocio, sistemas de información, tecnología, infraestructura y recursos humanos. Su objetivo es alinear la estrategia de la organización con su estructura y operaciones, y proporcionar un marco que permita a la organización gestionar y adaptarse eficazmente a los cambios en su entorno
- c) **Diseño de Arquitectura:** proceso de concebir, definir, expresar, documentar, comunicar, certificar la correcta implementación, mantenimiento y mejora de una arquitectura a lo largo del ciclo de vida de un sistema (ISO/IEC 42010).
- d) **Patrón arquitectónico:** Un patrón arquitectónico de software es una solución estructurada y probada para un problema recurrente en el diseño de software. Estos patrones son abstracciones de soluciones comunes que se han identificado a lo largo del tiempo, y se utilizan como guías para la arquitectura y diseño de sistemas de software complejos
- e) **Sistemas:** Es un conjunto de elementos interrelacionados que trabajan juntos para lograr un objetivo común. Estos elementos pueden ser objetos físicos, procesos, personas, programas de software o cualquier combinación de ellos. Un sistema consta de varios componentes que interactúan entre sí para cumplir una función específica. Estos componentes pueden incluir entradas (inputs), procesos, salidas (outputs) y retroalimentación (feedback), que permiten a un sistema recibir información sobre su desempeño y ajustar su comportamiento para mejorar su eficacia
- f) **Vista de Arquitectura:** Es una representación abstracta y estructurada del sistema que proporciona una visión global y coherente de su estructura y

comportamiento. Esta vista está diseñada para comunicar los aspectos esenciales de la arquitectura del sistema a diferentes audiencias, como los arquitectos, desarrolladores, usuarios y gerentes de proyecto

- g) **Modelo de Arquitectura:** El modelo de arquitectura se utiliza para describir diferentes aspectos de la arquitectura, como la estructura, el comportamiento, la funcionalidad, la calidad y la evolución del sistema. Estos aspectos se organizan en un conjunto de vistas coherentes que proporcionan una visión global del sistema y permiten a los arquitectos, desarrolladores y otros interesados en el sistema comprender su diseño y funcionamiento

V MARCO CONCEPTUAL

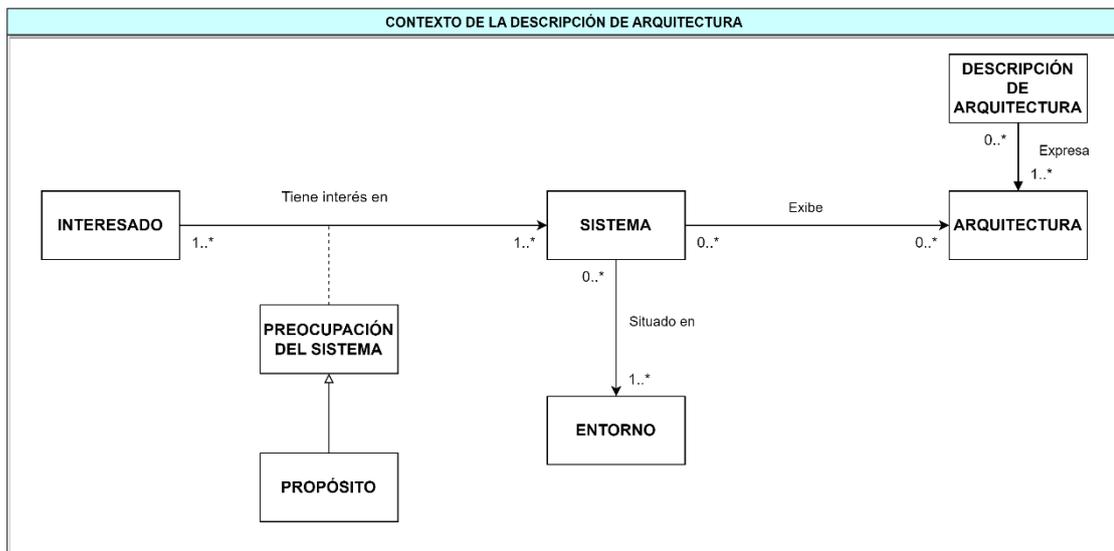
V.1 Modelo conceptual de la descripción de arquitectura

En el modelo conceptual de la descripción de la arquitectura nos permite conocer los conceptos importantes que están relacionados alrededor de los sistemas para entender de donde proviene dichos elementos de interés que van a describir su arquitectura. En la figura 1 se tiene el modelo en cuestión

Se entiende que para tener establecido un sistema o producto software, éste debe estar adoptado por uno o más interesados, los mismo que son identificados, catalogados y descritos, y el interés de éstos deben estar asentado por propósitos funcionales y no funcionales, lo mismo que se constituyen en preocupaciones en la que tiene que verse reflejada en el desarrollo final de un sistema.

El sistema está ubicado en el entorno. El entorno determina todos los alcances que debe tener o esta sobre el sistema a lo largo de su ciclo de vida, incluidas sus interacciones con ese entorno. El entorno del sistema puede contener otros sistemas.

Figura 1: Modelo conceptual de la descripción de la arquitectura



Elaboración: ISO/IEC 42010 Ingeniería de sistemas y software - Descripción de la arquitectura

V.1.1 Identificación de las partes interesadas y de sus preocupaciones

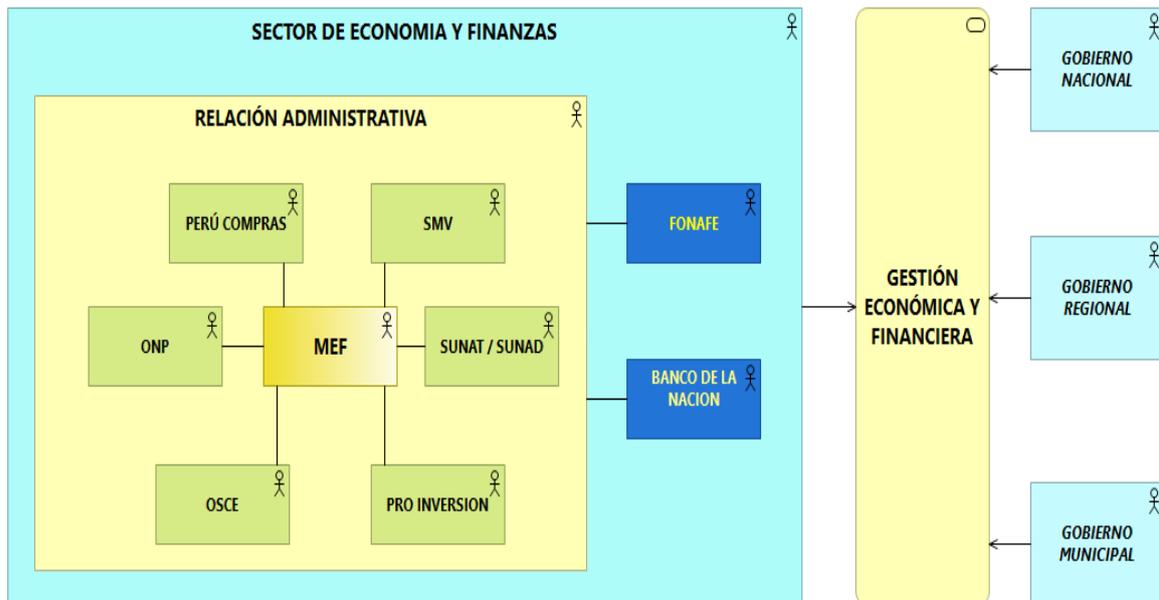
La identificación de los interesados y de sus preocupaciones y propósitos permite delinear los objetivos de la arquitectura de manera concreta, clara y precisa.

Además, aporta para establecer consensos en la construcción del diseño de la arquitectura y priorizar esfuerzos para la solución de problemas específicos que se puedan localizar o focalizar.

Se identifican las funcionalidades y no funcionalidades, supuestos, reglas o metas las que permitirán delinear el diseño, estilo o patrón arquitectónico del sistema el cual se va a desarrollar o implementar, dando una visión holística de la dimensión de los recursos a contar para llevar a cabo el desarrollo e implementación de la arquitectura del sistema.

Para este análisis nos podremos apoyar en la arquitectura empresarial (**AE**) que, en sus diversos dominios arquitecturales en su capa de negocio o misión, nos podría proporcionar de forma detallada y mapeada, el conjunto de interés y propósito que buscan los interesados en función o de acuerdo a su misión o modelo de negocio. Así tendríamos para el Ministerio de Economía y Finanzas las siguientes vistas de interés que tiene la organización viéndolo en una vista de arriba hacia abajo:

Figura 2: El Sector de Economía en la gestión económica y financiera nacional



Elaboración: Propia

En la figura 2 se tiene al MEF como eje central en el funcionamiento de la economía y finanzas del país, estableciendo relaciones funcionales y operativas con instituciones que conforman el sector economía y en la que, en base a su actuación, viabilizan la gestión económica y financiera para los distintos gobiernos nacional y subnacional del país, constituyéndose en el motor principal.

Para efectos arquitecturales de sistemas ya nos van dando muestras de los siguientes elementos funcionales y no funcionales que los sistemas deben poseer como parte de sus arquitecturas, como: interoperabilidad, seguridad, disponibilidad, etc.

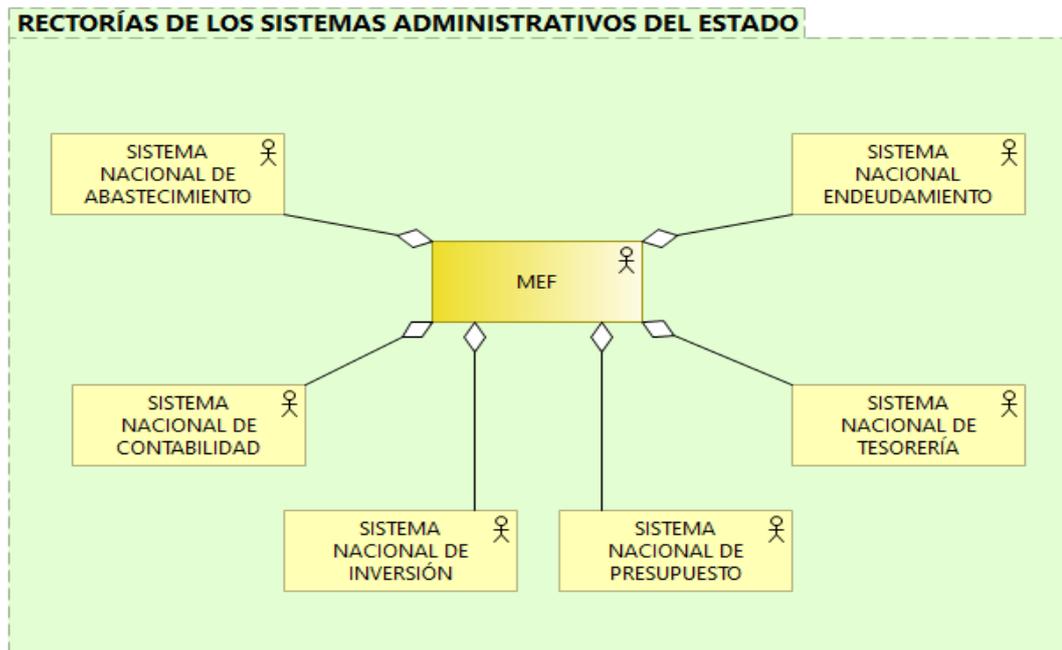
En la siguiente vista (figura 3) se identifica que el MEF por su Ley Orgánica, es Ente Rector de los sistemas nacionales administrativos que se muestran. Cada sistema administrativo tiene por finalidad regular la utilización de los recursos en las entidades de la administración pública, promoviendo la eficacia y eficiencia en su uso. Y las atribuciones como Ente Rector tienen las siguientes funciones:

- Programar, dirigir, coordinar, supervisar y evaluar la gestión del proceso administrativo
- Mantener actualizada y sistematizada las normativas del sistema
- Llevar registros y producir información relevante de manera actualizada y oportuna, entre otros¹.

Y siguiendo en el entendimiento e identificación de las partes interesadas, y tomando como ejemplo al Ministerio, tenemos en la siguiente figura los actores orgánicos que la constituye, para dimensionar las preocupaciones que puedan tener éstos en el desarrollo particular de sistemas e identificar sus necesidades.

En la figura 4, se muestra a los órganos que constituyen el Ministerio de Economía y Finanzas y de las cuales tienen funcionalidades específicas que integradas unos con otras desde lo administrativo o funcional, dan funcionamiento a la institución.

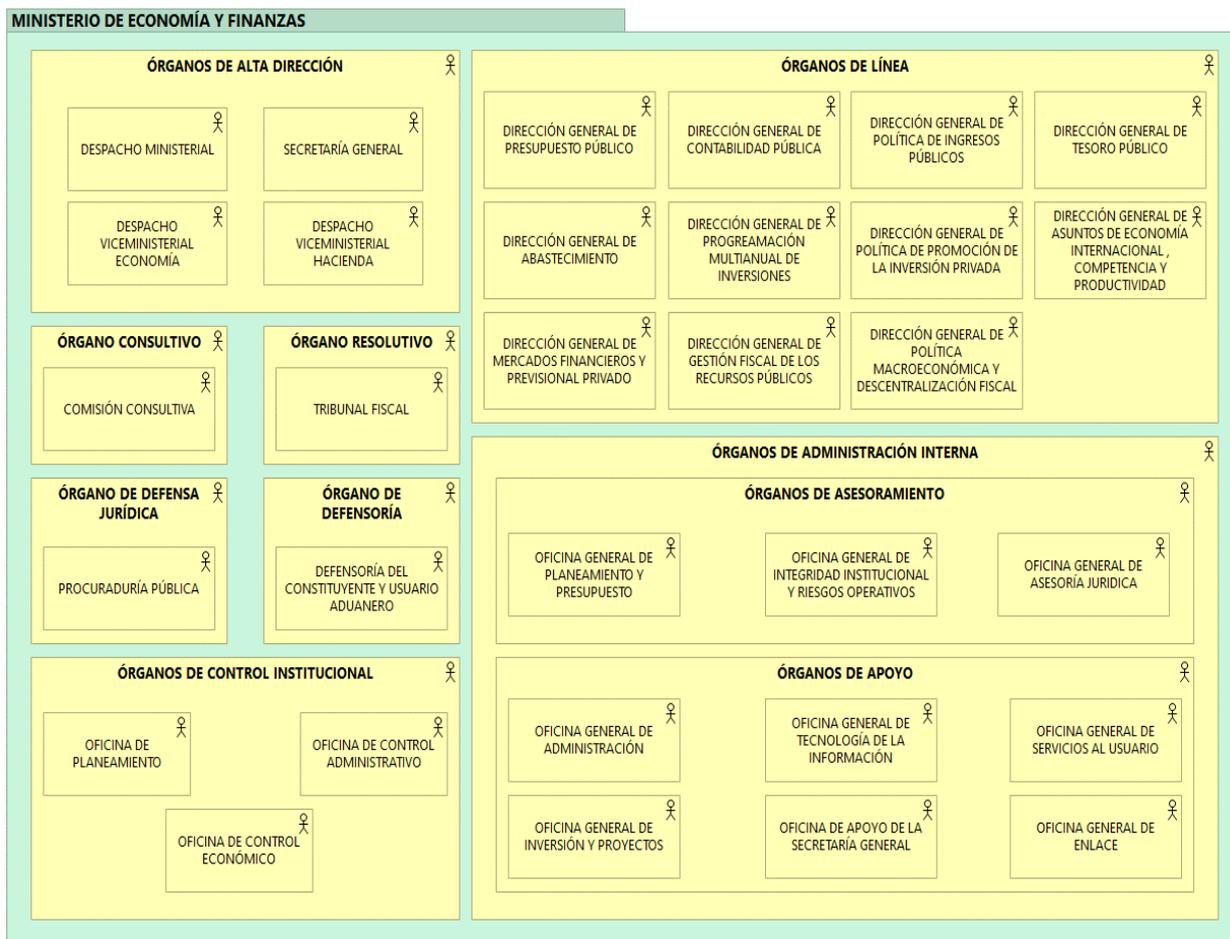
Figura 3: Rectorías que el MEF tiene asignado por Ley



Elaboración: Propia

¹ Ver descripción de responsabilidades de los Ente Rectores en la Ley N° 29158 "Ley Orgánica del Poder Ejecutivo"

Figura 4: Órganos que constituyen el MEF



Elaboración: Propia

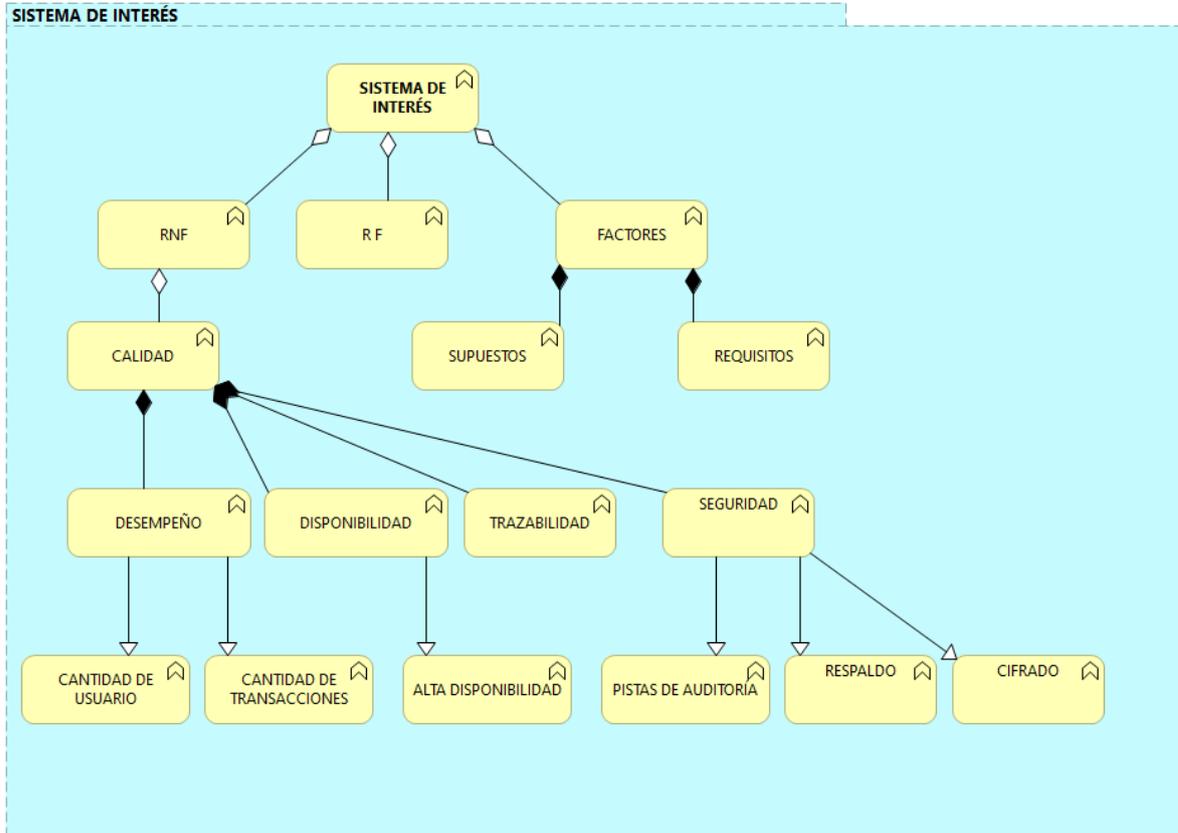
Como producto de la identificación de los interesados también se obtiene la identificación de las preocupaciones consideradas de importancia para la arquitectura del sistema de interés. Cuando se está en la identificación de la arquitectura se recomienda tener en cuenta los siguientes aspectos:

- El propósito del Sistema.
- Habilidad para crear e implementar un sistema.
- Riesgos potenciales del sistema e impacto en sus interesados a lo largo de su ciclo de vida.
- Mantenimiento y evolución del sistema.

En la figura 5 se muestra las preocupaciones e interés identificados y la que constituye, por su amplitud y especialización, un sistema de interés sobre el

sistema a ser visto para su desarrollo o mantenimiento, y en la que debe ser exhibida o expuesta en su arquitectura tal como se indica en la figura 1.

Figura 5: Interés / Preocupaciones del Sistema



Elaboración: Propia

A continuación, se describe ciertos requerimientos y factores que delinear la arquitectura del sistema, para ejemplarizar, se está tomando como base el sistema SIAF-SP, aplicativo informático que permite transacciones que se realiza en la Administración Financiera, en la que el MEF es Ente Rector:

REQUERIMIENTO		
RNF/Calidad	Característica	Descripción
Desempeño	Cantidad de usuarios	El sistema debe permitir su acceso a un número máximo de usuario en sus distintas estacionalidades del ejercicio presupuestal, contable y de tesorería, a nivel nacional. Toda vez que éste aplicativo es utilizado a nivel nacional por más de 3,000 unidades ejecutoras y a un estimado mínimo de usuarios de 60,000. Estas estacionalidades a tener en cuenta son; <ul style="list-style-type: none"> • Transacciones diarias

REQUERIMIENTO		
RNF/Calidad	Característica	Descripción
		<ul style="list-style-type: none"> Cierre contables mensual, trimestral y semestral Y, al término del ejercicio fiscal tanto a fin de año presupuestal y marzo del siguiente año.
	Cantidad de transacciones	El aplicativo al ser utilizado a nivel nacional por las unidades ejecutoras las transacciones financieras y contables se realizan por segundo, lo cual debe permitir su máximo de transmisiones de información. Estadísticamente en promedio, se estima que se realizan 86 transacciones por segundo haciendo un promedio anual de 45 millones de transacciones a nivel nacional.
Disponibilidad	Alta Disponibilidad	En virtud de que el aplicativo permite el ejercicio presupuestal y el pago de obligaciones del Estado, Aproximación al 99% de disponibilidad, priorizando el horario y días laborales, días de cierre, fines de mes y fin de año y en función a la infraestructura donde está soportada.
	Tolerancia a Fallas	Los errores en cualquiera de sus componentes deben alertar al operador sin que el sistema se bloquee. Se considera el manejo de errores transaccionales, la descripción del error y las acciones a realizar por parte del usuario final, y la generación de alertas al operador. Considere crear un registro de auditoría que permita que el sistema se relaje en caso de un error.
Seguridad	Auditoria	Generar pistas de auditoría que permitan identificar el estado de las transacciones y el usuario que la generó.
	Copias de Respaldo	El repositorio donde se almacena la información debe tener un procedimiento de backup para asegurar la continuidad del negocio.
	Seguridad perimetral	Se debe considerar un mecanismo para el filtrado y monitoreo del tráfico HTTP para la protección de ataques (WAF).
	Cifrado	La información sensible que se determine, debe de almacenarse de manera cifrada.

FACTORES	
Convivencia	Mantener convivencia con las aplicaciones web legadas, y progresivamente sus funcionalidades deben ser implementadas sobre la arquitectura de referencia descrita en este documento

Uso de DevSecOps	La arquitectura en el la que se desarrolle las aplicaciones modernizadas, debe permitir una mayor velocidad en el despliegue de actualizaciones que se les realice. El uso de DevSecOps como práctica común en el área de Tecnologías de la Información permitirá establecer un proceso ordenado, ágil y seguro para el despliegue de aplicaciones en los diferentes ambientes de trabajo.
Infraestructura Tecnológica	La infraestructura tecnológica debe tener la capacidad de soportar la nueva arquitectura
Portabilidad en ejecución	Las aplicaciones del Ministerio (como el SIAF) deben ser portables para su ejecución. Pudiendo estar éstas en la nube (pública o privada) o en el centro de datos del MEF, sin que ello involucre cambios significativos en la solución de arquitectura.

Cabe indicar que los atributos de calidad son de importancia para establecer un sistema de evaluación de la calidad del producto. Por lo que es indispensable establecer las características de calidad que deben considerarse, como parte del interés, que deba poseer el sistema y que la arquitectura debe asumir.

Estos elementos de calidad contribuyen a que estos drivers de arquitectura sean el soporte para planteamiento de los estilos, patrones y estrategias arquitectónicas que forman parte de la propuesta, sin embargo, esto no limita el uso de otros atributos de calidad, sino que al encontrarnos con conceptos que se interponen entre sí, estos drivers tendrían mayor peso en el orden indicado.

V.1.2 Arquitectura y descripción de la Arquitectura

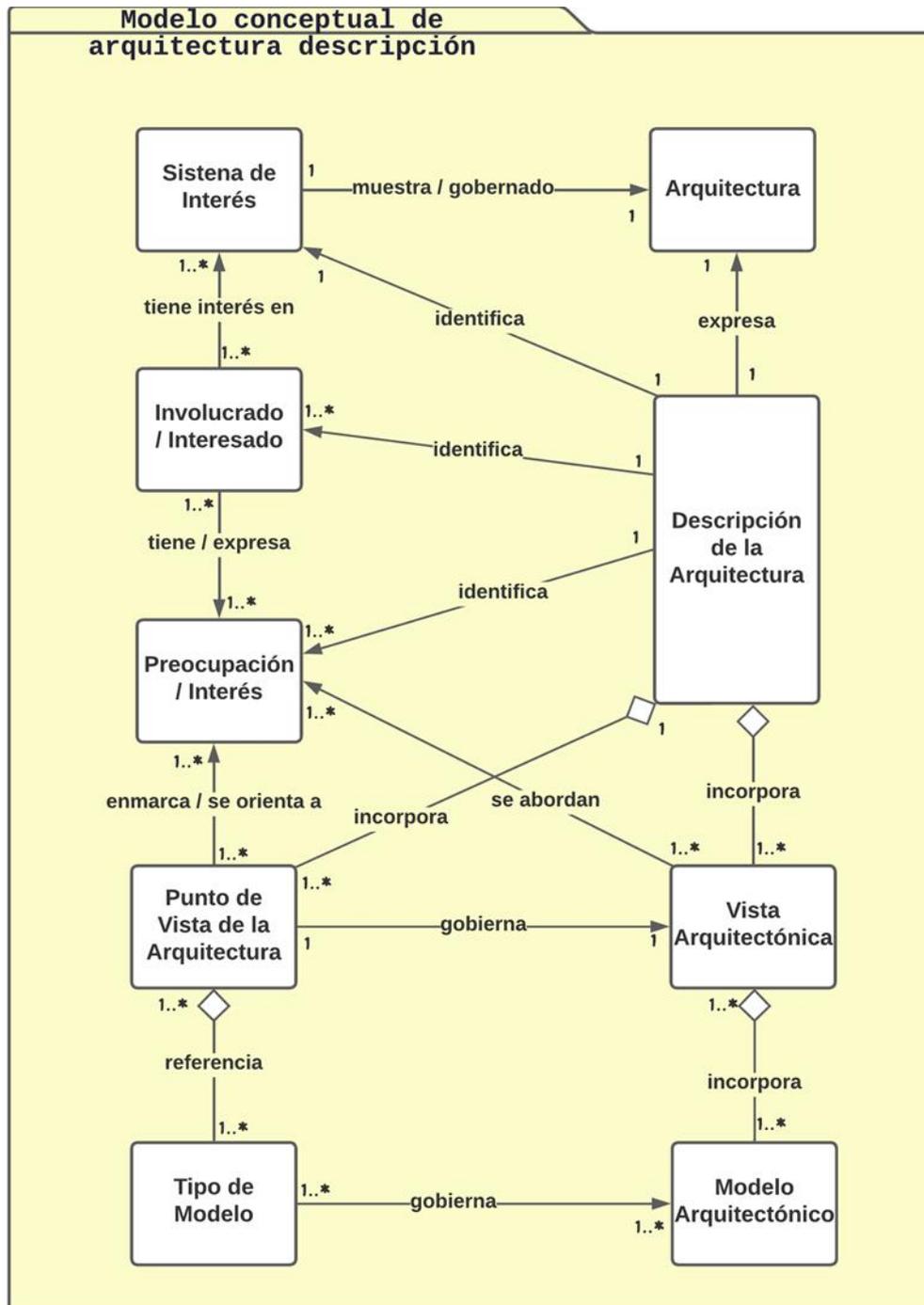
Una descripción de la arquitectura expresa la arquitectura de manera completa e integral utilizando uno o más "puntos de vista" que abordan los intereses/inquietudes de las partes interesadas sobre un sistema o componente técnico. En general, la arquitectura es difícil de expresar completamente con una sola vista.

De acuerdo a la ISO/IEC 42010 "Ingeniería y software – Arquitectura Descripción", en la figura 6 representa todos los conceptos relacionados con la práctica de la descripción de arquitectura cuando se aplica para producir una descripción. Las descripciones de arquitectura son productos de trabajo de sistemas y arquitectura de software.

En el punto V.1.1 se ha podido identificar tanto las partes interesadas, sus preocupaciones y los intereses que del sistema de poseer en la arquitectura. Ahora se desarrolla de forma ejemplar la a arquitectura y descripción de la arquitectura.

Como se puede ver en el diagrama de la Figura 6, cada sistema/solución tiene una arquitectura sobre la cual se diseña, implementa y controla su desarrollo, esté documentado, estructurado o no, dicho sistema fue o será construido de acuerdo con algunos principios de diseño representados por su método de construcción.

Figura 6: Conceptos relacionados de la descripción de arquitectura



Elaboración: ISO/IEC 42010 Ingeniería de sistemas y software - Descripción de la arquitectura

Para la descripción de la arquitectura, existen varios de marcos de referencia de arquitectura que apoyan a este propósito, entre los más conocidos tenemos, por ejemplo:

- Archimate**, proporciona una representación esquemática unificada de la arquitectura empresarial y se desarrolla en el lenguaje de modelado de

arquitectura empresarial ArchiMate. Proporciona un enfoque arquitectónico integrado que puede describir y visualizar diferentes dominios arquitectónicos y sus relaciones y dependencias subyacentes.

- **Modelo de vista 4+1 de Kruchten**, Este modelo es utilizado para describir la arquitectura de un sistema de software basado en el uso de distintos puntos de vista: vista lógica, vista de desarrollo, vista de procesos, vista física y la de escenarios, esta sirve para crear vínculos entre las demás.
- **TOGAF**: es un esquema (o marco de trabajo) de arquitectura empresarial que proporciona un enfoque para el diseño.
- **UAF**: marco de arquitectura unificada de OMG, es un enfoque para representar la arquitectura empresarial que permite a las partes interesadas centrarse en áreas específicas de interés empresarial manteniendo una visión global. En este caso arquitectura respecto a la tecnología y a las aplicaciones software.
- **Marco Zachman**: es un esquema, una clasificación unificada de un conjunto de representaciones descriptivas relevantes para la descripción de una empresa u organización que están normalizada de acuerdo a lo establecido por este marco.

V.1.3 Puntos de vistas

Los puntos de vistas establecen el gobierno, notación convenciones, zonas, tipos de conectores, nivel de abstracción de las **vistas arquitectónicas**; una vista no puede existir sin un elemento que lo gobierne.

Cada preocupación identificada debe estar enmarcada por al menos un punto de vista, esto para que todas las preocupaciones identificadas estén cubiertas. Estos puntos de vistas se especifican mediante uno o más tipos de modelos que se utiliza.

Además, para ayudar a los arquitectos, las vistas pueden definir cualquiera de las siguientes opciones:

- Criterios, normas y métodos de comparación para verificar la integridad (de declaraciones) o la consistencia (entre declaraciones).
- Técnicas, heurísticas, métricas, patrones, reglas o pautas de diseño, mejores prácticas y ejemplos para ayudar a crear y sintetizar vistas.

V.1.4 Tipo de modelo

Define la convención de generación de modelos de diagramación como pueden ser: UML, BPMN, Archimate, lenguajes, convenciones, técnicas de modelado, entre otros. El tipo de modelo captura convenciones para enmarcar un conjunto de preocupaciones.

V.1.5 Vistas de arquitectura

Una descripción de arquitectura incluye una o más vistas de arquitectura. Una vista de arquitectura (o simplemente, vista) aborda una o más de las preocupaciones de las partes interesadas del sistema. Una vista de arquitectura expresa la arquitectura del sistema de interés de acuerdo con un punto de vista de arquitectura (o simplemente, punto de vista). Hay dos aspectos en un punto de vista: las preocupaciones que enmarca para las partes interesadas y las convenciones que establece sobre las opiniones.

Las vistas arquitectónicas ayudan a visualizar en sus diferentes puntos de vistas, la complejidad del sistema y, además, permite que esta sea visualizada en los distintos niveles técnicos y funcionales para distintos interesados de acuerdo a su rol involucrado.

En el marco de la mejora y modernización de las aplicaciones del Ministerio, y en base al sistema de interés y preocupaciones detalladas en el punto V.1.1, se presentan las siguientes vistas de arquitectura que contribuye a los puntos de vistas de solución a ser atendidas en base a las preocupaciones e interés de los interesados del Ministerio

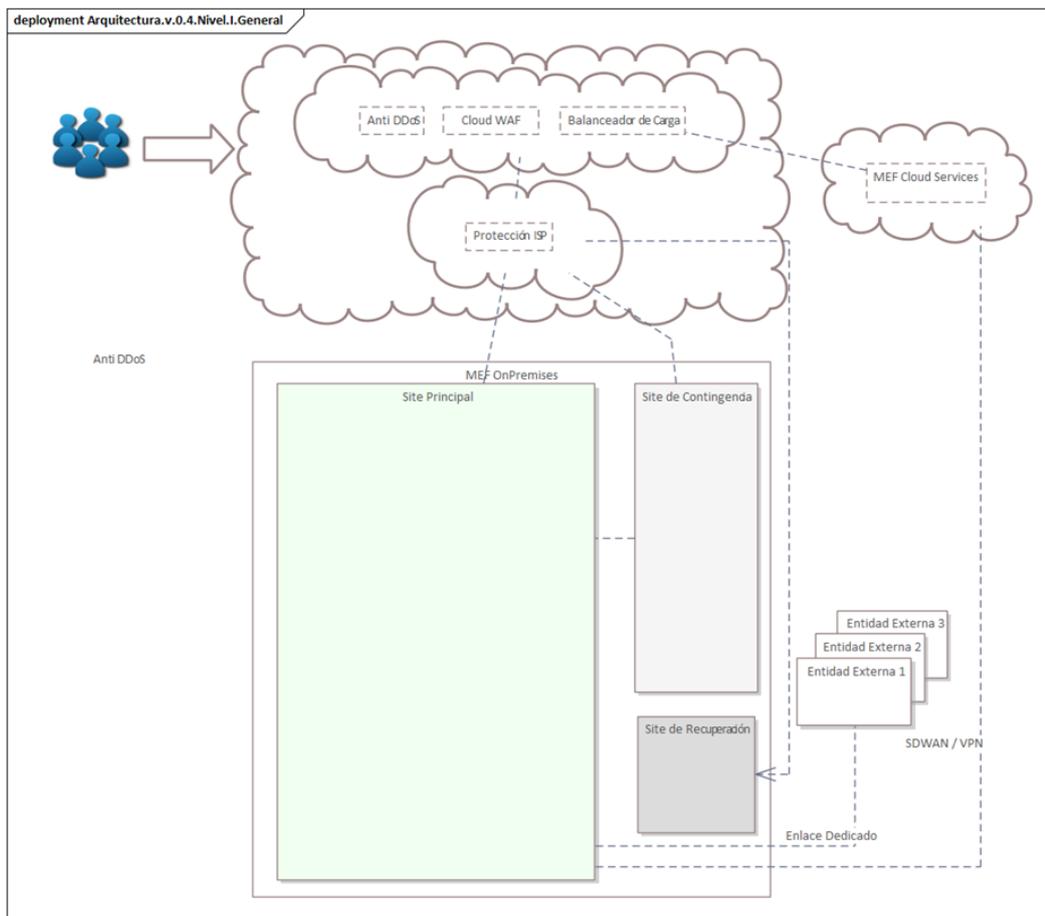
V.1.6 Modelo arquitectónico

Es una representación a escala de la realidad, sobre el tema en particular. El metamodelo que se indica en la figura 6, está compuesta por uno o muchos modelos, así como un modelo puede estar incluido en muchas vistas.

Una vista de arquitectura se compone de uno o más modelos de arquitectura. Un modelo de arquitectura utiliza convenciones de modelización adecuadas a las preocupaciones que deben abordarse. Estas convenciones se especifican por el tipo de modelo que rige ese modelo. Dentro de una descripción de arquitectura, un modelo de arquitectura puede ser parte de más de una vista de arquitectura

i. Vista de la Arquitectura de la Plataforma de los Servicios Digitales del MEF

Figura 7: Arquitectura de la plataforma de los servicios digitales del MEF. Fuente MEF-OGTI

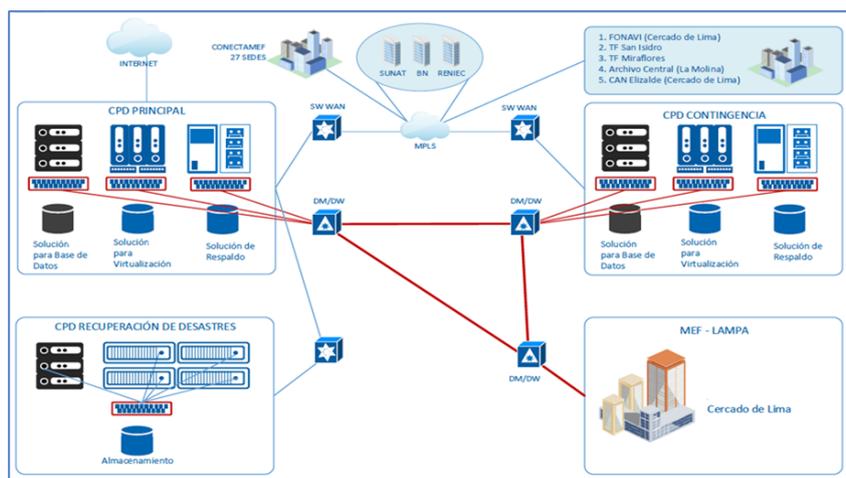


Elaboración: Propia

Numeración	01
Denominación de la Vista	Distribución de los centros de datos con los que cuenta la organización y servicios externos
Propósito	Representar los centros de datos al interior de la organización, su comunicación con el tráfico exterior
Contexto (preocupación / interés)	Identificar los centros de datos en la organización e identificar las capas de seguridad y formas de conexión del tráfico externo.
Problema a resolver	Mostrar que las soluciones a implementar deben considerar diversos centros de datos, para lograr una alta disponibilidad en los servicios digitales
Componente de la vista de arquitectura	Descripción del componente
Site principal	Representa el centro de cómputo principal
Site de contingencia	Representa el centro de cómputo de contingencia
Site de recuperación	Representa el centro de cómputo para recuperación de desastres
Anti DDOs , WAF, Balanceador de Carga	Servicios de protección de Denegación de Servicios, servicio de cortafuegos web y balanceo de carga provistos por un proveedor externo.
Protección ISP	Conjunto de servicios de protección provistos por el proveedor de internet
MEF cloud service	Representa los servicios contratados por MEF en proveedores de Nube, cuyo enlace puede darse a través de conexión tipo VPN o WAN extendidas definidas por software
Entidad Externa 1,2,3	Representa a las entidades externas que pueden acceder a servicios del MEF a través de enlaces dedicados

ii. Vista de la Arquitectura de Tecnológica

Figura 8: Vista de la arquitectura referencial tecnológica

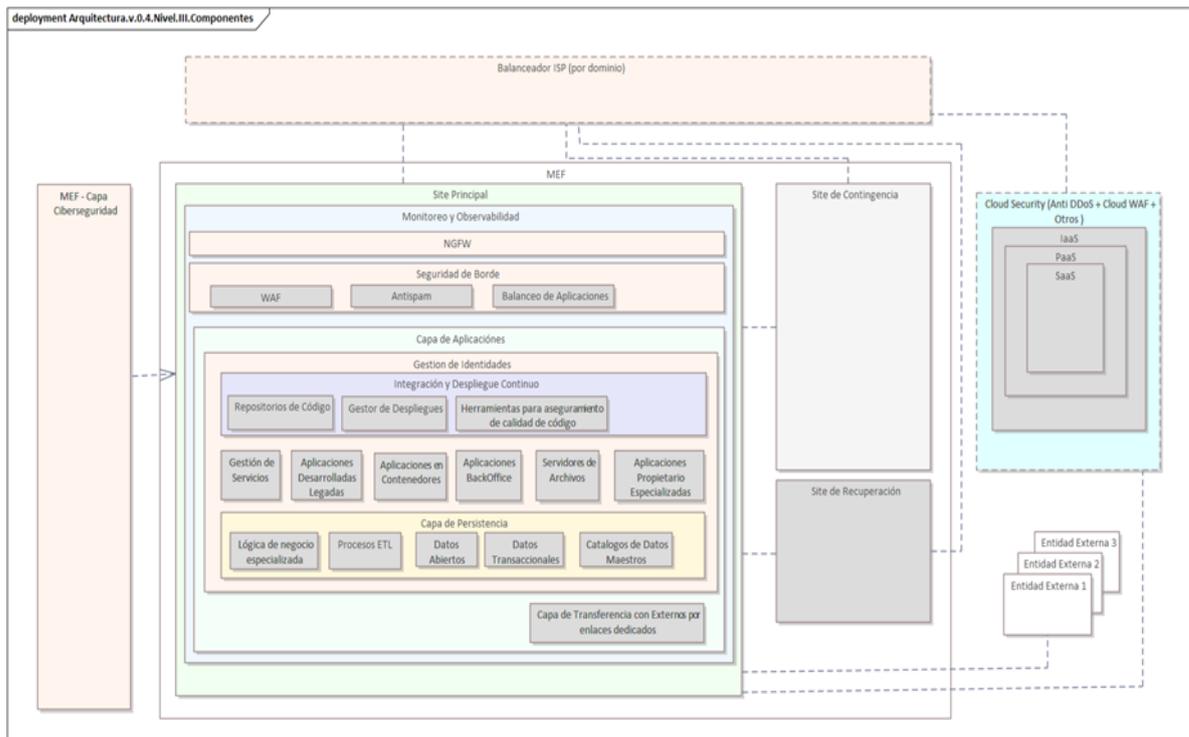


Elaboración: MEF - OGTI

Numeración	02
Denominación de la Vista	Distribución de la infraestructura tecnológica
Propósito	Representar en alto nivel la distribución de componentes de infraestructura en cada centro de datos
Contexto (preocupación / interés)	Identificar la distribución física de infraestructura
Problema a resolver	Describir a alto nivel los componentes de infraestructura en cada site y los componentes de borde utilizados para la comunicación.
Componente de la vista de arquitectura	Descripción del componente
CPD Principal	Representación de alto nivel de las capacidades de infraestructura en el centro de cómputo principal
CPD Contingencia	Representación de alto nivel de las capacidades de infraestructura en el centro de cómputo de contingencia
CPD Recuperación de Desastres	Representación de alto nivel de las capacidades de infraestructura en el centro de cómputo para recuperación de desastres.

iii. Vista de la Arquitectura de Aplicaciones

Figura 8: Arquitectura referencial de Aplicaciones. Fuente: MEF-OGTI



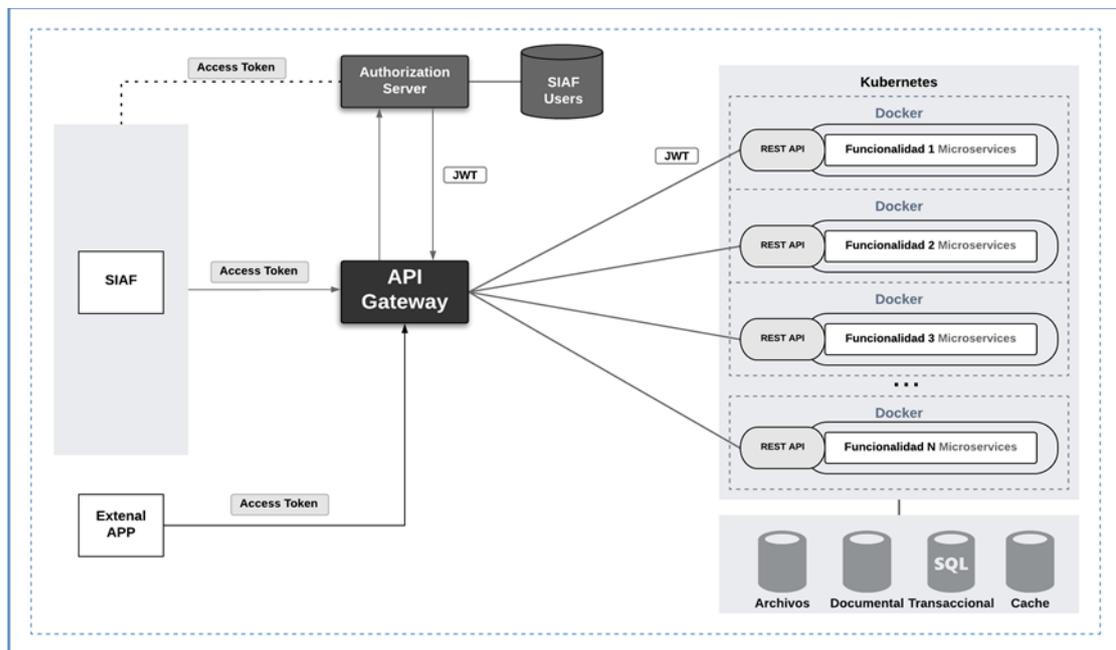
Elaboración: Propia

Numeración	03
Denominación de la Vista	Agrupación de aplicaciones en el site principal
Propósito	Representar los grupos de aplicaciones existentes en el site principal
Contexto (preocupación / interés)	Identificar los grupos de aplicaciones existentes
Problema a resolver	Representar las aplicaciones agrupadas y categorizadas en el site principal
Componente de la vista de arquitectura	Descripción del componente
Site Principal	Centro de Computo principal
Monitoreo y Observación	Conjunto de componentes de software y prácticas que permiten monitorear los componentes en operación
NGFW	Cortafuegos de última generación
Seguridad de Borde	Conjunto de aplicaciones que brindan protección para el acceso web

Capa de Aplicaciones	Capa que delimita la ejecución de aplicaciones heterogéneas para diversos servicios digitales
Gestión de Identidades	Conjunto de componentes utilizados para la autenticación y autorización hacia cualquiera de las aplicaciones.
Integración y Despliegue Continuo	Conjunto de aplicaciones que integran el código fuente, facilitan su despliegue y revisión de calidad.

iv. Vista de la Arquitectura de operaciones de la plataforma con el API Getway y Tokens

Figura 9: Arquitectura referencial de operaciones de la plataforma con el API Gateway y Tokens.
Fuente: MEF-OGTI



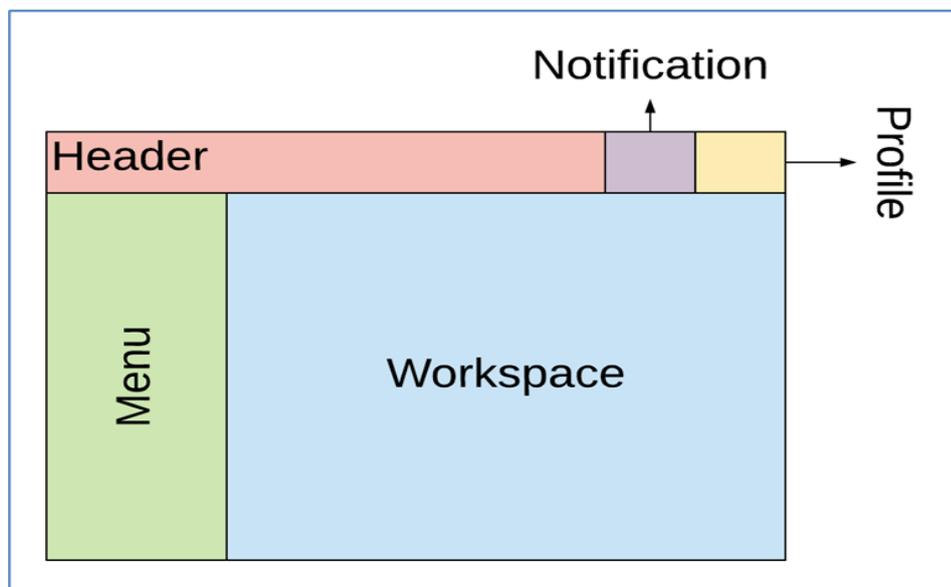
Elaboración: Propia

Numeración	04
Denominación de la Vista	Componentes principales de la arquitectura de aplicaciones
Propósito	Representar en alto nivel los componentes principales y las relaciones de los componentes que conforman la arquitectura de referencia
Contexto (preocupación / interés)	Identificar el acceso de los usuarios de aplicaciones tipo SIAF a través de los componentes hacia los servicios REST
Problema a resolver	Evitar que las invocaciones a servicios API REST se realicen con implementaciones disimiles
Componente de la vista de arquitectura	Descripción del componente

API Gateway	Componente que se encarga de gestionar las peticiones a los diferentes grupos de aplicaciones backend
SIAF	Representa la invocación de los usuarios a través de la aplicación SIAF
External APP	Representa la invocación de las aplicaciones No SIAF
Autorization Server	Componente para la autenticación
SIAF Users	Base de Datos de usuarios autorizados para el ingreso a las aplicaciones SIAF
Kubernetes	Plataforma para la orquestación de contenedores donde se ejecutan los servicios backend

- v. Vista de la Arquitectura de la Distribución de los componentes FrontEnd en un contenedor principal.

Figura 10: Arquitectura referencial de distribución de componentes FrontEnd. Fuente: MEF-OGTI



Elaboración Propia

Numeración	05
Denominación de la Vista	Vista de la distribución de componentes frontend en el contenedor principal
Propósito	Establecer un modelo guía para un estilo de entrada único para la navegación de las aplicaciones web de las soluciones que se determine

Contexto (preocupación / interés)	<ul style="list-style-type: none"> • Un punto de entrada único que se va transformando y adaptando mediante acciones. • Optimización del tiempo de carga. • Permite una navegación fluida. • Permite el uso de operaciones offline desde dispositivos móviles, permitiendo la sincronización posterior al detectar Internet. • Diseñado para PWA (Progressive Web App), de tal forma que permita la navegación de la aplicación WEB en los dispositivos móviles como si fueran aplicaciones nativas, incluyendo la funcionalidad offline. • El stack tecnológico en SPA se puede consultar en el Anexo 1 - "Descripción de los componentes". • La visualización para el usuario es un contenedor que permite destruir diferentes proyectos especializados, que pueden evolucionar independientemente. • Priorizar la creación de módulos reutilizables, denominados Web Componentes. • Hacer uso de PWA (Web Workers), para por lo menos, crear caché de contenido estático y dependiendo de la aplicación hasta lograr un funcionamiento offline
Problema a resolver	Evitar acoplamiento innecesarios en las aplicaciones Frontend, manteniendo un estilo único para las aplicaciones web
Componente de la vista de arquitectura	Descripción del componente
Header	Aplicación que contiene el logo de la aplicación y permite expandir o colapsar el proyecto "Menú"
Menu	Aplicación que contiene las opciones disponibles para el usuario. Las opciones se pueden agrupar por funcionalidad
Notification	Aplicación que permite mostrar al usuario los mensajes generados por el sistema, o notificaciones generadas durante un proceso de negocio
Profile	Aplicación que contiene la información del usuario, y las funcionalidades para actualizar los datos del usuario, solicitar un ticket de soporte, realizar un tour por la aplicación o cerrar sesión
Workspace	Aplicación donde se muestran los microfrontend con las funcionalidades de negocio

V.1.7 Estilos arquitectónicos referenciales

Los estilos de arquitectura de software son patrones de diseño de software utilizados para estructurar y organizar sistemas de software. Cada estilo arquitectónico tiene características, ventajas y desventajas únicas que se utilizan para abordar diversos problemas arquitectónicos.

Los estilos de arquitectura de software son patrones de diseño repetibles que se utilizan para organizar y estructurar las partes de un sistema de software. Estos patrones especifican las relaciones y la distribución de las partes del sistema. El

uso de estilos arquitectónicos es parte del marco para la descripción y documentación de la arquitectura de sistemas proporcionada por el estándar ISO/IEC 42010.

Los estilos de arquitectura de software, en pocas palabras, son patrones de diseño que ayudan a los arquitectos de software a elegir los componentes fundamentales de la estructura de un sistema de software. Cada estilo se utiliza para abordar varios tipos de problemas arquitectónicos y tiene sus propias características, ventajas y desventajas.

A continuación, se presentan algunos estilos arquitectónicos más comunes a tener en cuenta:

i. N-CAPAS (N-TIER)

<p>Descripción:</p>	<p>Una arquitectura de N capas, divide una aplicación en capas lógicas y físicas. Las capas son una forma de separar las responsabilidades y gestionar las dependencias. Cada capa tiene una responsabilidad específica. Una capa superior puede usar servicios en una capa inferior, pero no al revés.</p> <p>Esta separación en capas también permite distribuir los elementos de la arquitectura conforme a las restricciones de infraestructura y comunicaciones que existen como zonas militarizadas, no militarizadas, nubes privadas, red interna, etc. Ayudando a dotar de mayor seguridad a la solución.</p>
<p>Justificación</p>	<p>Facilita el mantenimiento de la aplicación, al tenerse separada la funcionalidad en capas independientes, que pueden actualizarse sin que se afecten otras, tanto desde una perspectiva global de la plataforma única como desde la perspectiva de un servicio.</p> <p>Facilita la escalabilidad de la plataforma única, dado que las capas se despliegan de manera independiente, es posible incrementar de manera horizontal cada capa.</p> <p>Brinda flexibilidad, dado que permite cambiar una capa por otra sin afectar a otras.</p>
<p>Drivers asociados:</p>	<p>Fiabilidad → Tolerancia a fallas</p> <p>Mantenibilidad → Modularidad</p> <p>Portabilidad → Capacidad ejecutarse en diferentes ambientes</p>
<p>Driver en conflicto:</p>	<p>Eficiencia → Utilización de recursos</p>

ii. Microservicios

Descripción	El estilo de microservicios está relacionado a aplicaciones Nativas de Nube, este estilo permitirá dar cumplimiento a los drivers de: Eficiencia, Fiabilidad y Portabilidad; en orden de importancia.
Justificación	Se ha elegido el estilo arquitectónico de Microservicios por las siguientes razones: <ul style="list-style-type: none"> ● Por necesidad de un desarrollo modular en sprints, el cual requiere de cambios constantes y a una alta velocidad de liberación a producción. ● Porque necesita ser altamente escalable. ● Porque está asignado a una organización con equipos de desarrollo pequeños. Las ventajas de utilizar Microservicios son las siguientes: <ul style="list-style-type: none"> ● Implementaciones independientes. Puede actualizar un servicio sin volver a implementar la aplicación completa y revertir o adelantar una actualización si algo sale mal. Las correcciones de errores y versiones de funciones son más manejables y menos riesgosas. ● Desarrollo independiente. Un solo equipo de desarrollo puede construir, probar y desplegar un servicio. El resultado es una innovación continua y un ritmo de liberación más rápida. ● Equipos pequeños y enfocados. Los equipos pueden enfocarse en un servicio puntual. El alcance más pequeño de cada servicio hace que la base del código sea más fácil de entender para el equipo actual, así como para los nuevos miembros del equipo. ● Aislamiento de fallos. Si un servicio falla, no penalizará a toda la aplicación. Sin embargo, eso no quiere decir que la aplicación no podrá fallar, se puede aplicar otras técnicas de resiliencia para completar un modelo óptimo. ● Uso de tecnología mixta. Los equipos pueden elegir la tecnología que mejor se adapte a su servicio. ● Escalamiento granular. Los servicios se pueden escalar de forma independiente, en función a la demanda, y aprovechando los recursos de las máquinas virtuales por completo Los Microservicios están altamente relacionado con el uso de contenedores, porque es la tecnología que permite una correcta implementación de este estilo arquitectónico, y por ende a un manejador de contenedores. Con los Microservicios aparece un nuevo elemento en la arquitectura, que es el API Gateway, necesario para enrutar a los diferentes URL desde una misma IP o DNS
Drivers asociados	Eficiencia → Utilización de recursos Fiabilidad → Disponibilidad Tolerancia a fallas Capacidad de recuperación Mantenibilidad → Modularidad Portabilidad → Capacidad ejecutarse en diferentes ambientes

Driver en conflicto	
---------------------	--

iii. Command and Query Responsibility Segregation (CQRS- Command and Query Responsibility Segregation)

Descripción:	El estilo arquitectónico CQRS que permite separar la responsabilidad de comandos y consultas hacia la base de datos.
Justificación	<p>Con este estilo arquitectónico se busca que distintas funcionalidades de persistencia sean segregadas, considerando inclusive el uso de diferentes bases de datos físicas, según sea la necesidad del negocio (Polyglot Persistence²).</p> <p>Para el caso de la aplicación PLATAFORMA SIAF se observa dos necesidades bien marcadas del uso de la data, la primera es que se requiere persistir la data operacional y la segunda es que se requiere consultar o generar reportes de esta. Al separar en base de datos por ejemplo de escritura y lectura evita que se penalicen entre ellas.</p>
Drivers asociados:	Fiabilidad → Disponibilidad Tolerancia a fallas Capacidad de recuperación Eficiencia → Desempeño
Driver en conflicto:	DR-008: Eficiencia → Utilización de recursos

iv. Orientado a Eventos (Event-Driven)

Descripción:	El estilo arquitectónico event-driven que permite generar un flujo de eventos.
Justificación	Con este estilo arquitectónico se busca poder transmitir la información desde la aplicación hacia el legacy en Oracle de forma casi instantánea y asegurando que llegue a destino.
Drivers asociados:	Fiabilidad → Recuperable Eficiencia → Desempeño

²<https://docs.aws.amazon.com/whitepapers/latest/modern-application-development-on-aws/polyglot-persistence.html>

Driver en conflicto:	Ninguno
----------------------	---------

V.1.8 Capas Arquitectónicas

Las capas arquitectónicas son un método común que se utiliza en la arquitectura y el diseño de software para colocar discretamente en capas los diversos componentes funcionales de un sistema. Cada capa tiene una responsabilidad específica y una interfaz bien definida para comunicarse con las capas adyacentes.

Dos ideas clave en la arquitectura y el diseño de software son los estilos y las capas. Los estilos arquitectónicos, como se ha indicado, son patrones de diseño comunes que se utilizan para organizar el software en términos de sus partes constituyentes y cómo interactúan. Por otro lado, las capas son un método popular para estructurar el software en capas lógicas y físicas, donde cada capa es responsable de un conjunto específico de tareas. Los estilos arquitectónicos pueden influir en la forma en que se organizan las capas de un sistema de software.

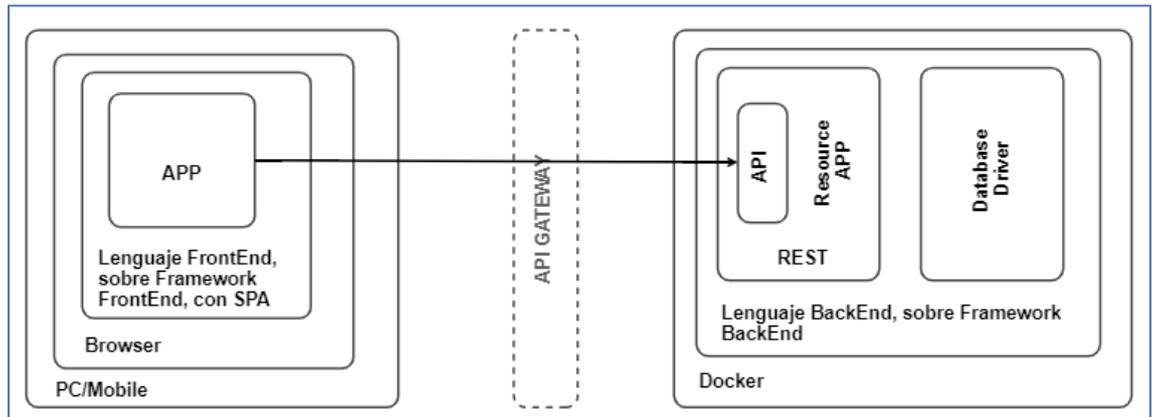
Así tenemos las siguientes capas arquitectónicas referenciales a tener en cuenta:

- i. Capa de servicios (procesos)
La Capa de servicios está desarrollada con API REST, con las siguientes características:

Stack BackEnd – APIs

- El stack del BackEnd estará basado en microservicios, este será identificado haciendo uso de DDD (Domain Driven Design) o por funcionalidad
- El stack tecnológico del BackEnd se puede consultar en el
- Un componente en el BackEnd principalmente expondrá APIs REST.
- La coreografía de las APIS será vía el API Gateway.
- Los componentes construidos en el BackEnd será Dockerizados.
- Uso del estándar OpenAPI version 3 para la especificación de las API REST.
- Los componentes backend deben incluir además del código fuente, lo casos de prueba. El porcentaje de cobertura de código es parte de la especificación del componente de revisión de código estático.

Figura 11: Representación de la invocación desde una capa Front hacia una capa back. Se incluyen las descripciones de la tecnología sobre la que están implementadas ambas capas



ii. Capa de seguridad

La Capa de Seguridad cuenta con las siguientes características:

- Es transversal a todas las capas de desarrollo que necesiten de su uso.
- Utiliza un módulo de gestión de perfiles y accesos.
- Permite el registro de logs de operación de los programas y acciones de cada usuario.
- Permite registrar información relevante para auditoría.
- Busca alinearse a los principios del nuevo S-SDLC (Ciclos de vida del Software Seguro) basado en metodologías ágiles.
- Cubre los 10 riesgos de seguridad identificados por la organización OWASP (Open Web Application Security Project):
 - A1-Inyección
 - A2-Pérdida de autenticación
 - A3-Exposición de datos sensibles
 - A4-Entidades Externas XML (XXE)
 - A5-Pérdida de Control de Acceso
 - A6-Configuración de Seguridad Incorrecta
 - A7-Secuencia de Comandos en Sitios Cruzados (XSS)
 - A8-Deserialización Insegura
 - A9-Componentes con vulnerabilidades conocidas
 - A10-Registro y Monitoreo Insuficientes
- En cuanto a la normativa peruana, la arquitectura se alinea a:
 - Ley de Gobierno Digital, Decreto Ley N° 1412 y su reglamento aprobado por D.S. N° 029-2021-PCM.
 - Ley de protección de datos personales, Ley N° 29733, y su reglamento aprobado por D.S. N° 003-2013-JUS.
 - NTP-ISO/IEC 27001:2014, de uso obligatorio según R.M. N° 004-2016-PCM, sobre sistemas de gestión de seguridad de la información. Para el

propósito mencionado por la DGPIP, bastaría la aplicación del Anexo A, controles de referencia.

- Dispositivos internos.
 - Política de seguridad de la información (RM 210-2020-EF/44)
 - Metodología de gestión de riesgos de seguridad de la información (RD 120-2016-EF/43.01)
 - Manual de políticas de gestión de TI (RD 478-2016-EF/43.01)

iii. Capa de persistencia

Esta capa contempla diversos escenarios:

- Base de Datos relacionales, como un repositorio intermedio entre los microservicios y la base de datos transaccional principal.
- Base de Datos no relacional, para la generación de información de trazabilidad, logs, y auditorías.
- Los microservicios se desarrollan e implementan como contenedores de forma independiente entre ellos. Este enfoque implica que un equipo de desarrollo puede desarrollar e implementar un microservicio determinado sin afectar a otros subsistemas, considerando inclusive el uso de diferentes bases de datos físicas, según sea la necesidad del negocio con el fin que evitar contención en las Bases de Datos y también facilitaría el mantenimiento evitando dependencias por acoplamiento a un sistema de base de datos que atiende varios otros sistemas.

iv. Capa de cache

Cuenta con las siguientes características:

- Ayudará a reducir las consultas que se harían a las bases de datos relacionales y no relacionales.
- Estará bajo un diseño NONSQL que soporta clave y valor, para contar con lectura rápida.

V.2 Documentación sobre la arquitectura referencial

V.2.1 Componentes de la arquitectura referencial

A continuación, se detalla la tecnología base que constituye la arquitectura referencial tecnológica del Ministerio a tener en cuenta para los desarrollos de sistemas o software. En la lista siguiente, se describe el componente y la documentación referencial de apoyo para mayor conocimiento e interés.

Esta documentación, que acompaña en la lista se puede obtener a través de la siguiente dirección electrónica: <https://collca.mef.gob.pe/index.php/f/28163> .

Cabe mencionar que para acceder a la documentación mediante la dirección electrónica indicada línea arriba, previamente el solicitante debe estar debidamente autorizado por el Director del Órgano correspondiente y a su vez debe registrar su solicitud, acompañando la autorización en la Mesa de Servicio de TI (<https://mesadeservicioti.mef.gob.pe/>) en el Servicio: Aplicativos Institucionales / Solicitud de acceso de otros aplicativos, para generarle la autorización de acceso correspondiente.

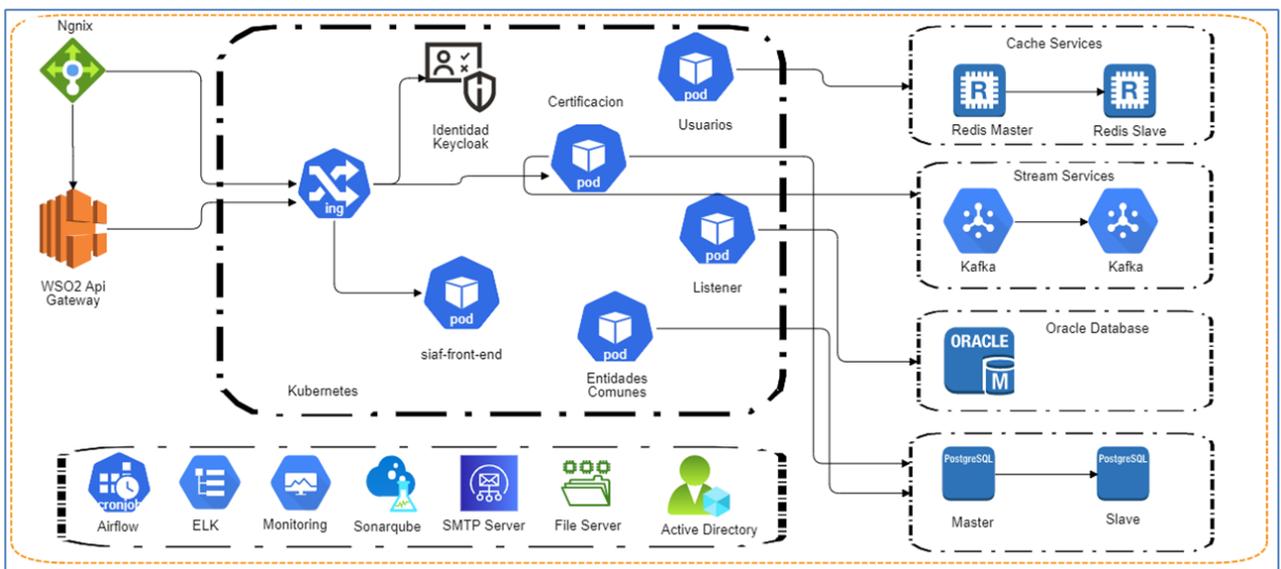
N°	Componente	Descripción	Documentos de Apoyo
Capa de seguridad			
1	Gestor de Identidades	Permite la gestión de acceso e identidad para aplicaciones SPA, aplicaciones móviles y API REST	ARQ.GI.INS (Instalación) ARQ.GI.OPE (Operación) ARQ.GI.DET (Descripción de operación con usuario y contraseña) ARQ.GI.DET.DNI (Propuesta de la operación con DNI Electrónico) ARQ.GI.DET.AD (Descripción de operación con ActiveDirectory)
Capa de aplicación			
2	Contenerización de aplicaciones	Permite la gestión de contenedores ligeros y portables para que las aplicaciones de software puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina, facilitando los despliegues	
3	Orquestador de Contenedores	Es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Facilita la automatización y la configuración declarativa.	
4	Aplicaciones Backend	El stack del BackEnd se basa en microservicios y principalmente expone APIs REST sobre componentes dockerizados.	ARQ.BE.EST (estándar de desarrollo)
			ARQ.API.EST (estándar de nomenclatura y uso)
5	Aplicaciones Frontend	El stack FrontEnd está desarrollada con tecnologías SPA (Single Page Application), permitiendo una navegación fluida, a través de un solo contenedor, pero con diferentes aplicaciones	ARQ.FE.EST (estándar de desarrollo)
6	Componente para la	Permite la creación, publicación y gestión de las	ARQ.GS.INS (Instalación) ARQ.GS.OPE (Operación)

N°	Componente	Descripción	Documentos de Apoyo
	gestión de servicios API	APIs (interfaces de acceso a nuestros recursos o servicios) y su ciclo de vida	
7	Base de Datos Relacional Transaccional	Componente que permite la persistencia de datos relacionales en la capa de aplicación. En las aplicaciones SIAF, estos datos viajarán a través de colas hacia la base de datos Oracle Central. También se utiliza para el almacenamiento de la metada de diversos componentes.	ARQ.BDR.INS (Instalación) ARQ.BDR.OPE (Operación)
8	Base de Datos No Relacional	Base de datos de documentos que permite la persistencia de trazas de auditoría y trazabilidad de las operaciones transaccionales.	ARQ.BDNR.INS (Instalación) ARQ.BDNR.OPE (Operación)
9	Base de Datos En Memoria	Componente que permite utilizar la memoria cache para el almacenamiento de información constantemente consultada y pocas veces modificada, evitando leer desde disco, agilizando la velocidad de respuesta y reduciendo tráfico entre componentes.	ARQ.BDM.INS (Instalación) ARQ.BDM.OPE(Operación)
10	Gestor de Colas	Plataforma de transferencia de datos a gran escala, que permite acceder, almacenar y administrar datos como transmisiones continuas, utilizada para los flujos de negocio que requieran distribuirse desde la Base de datos Relacional Transaccional hacia la Base de Datos Oracle Central.	ARQ.GC.INS (Instalación) ARQ.GC.OPE (Operación)
11	Gestor de Tareas	Componente que permite crear, programar y monitorear flujos de trabajo diversos, utilizando para: - La sincronización de datos desde la Base de Datos Oracle Central hacia la Base de datos en Memoria y la Base de datos Relacional Transaccional	ARQ.GT.INS (Instalación) ARQ.GT.OPE (Operación)

N°	Componente	Descripción	Documentos de Apoyo
		<ul style="list-style-type: none"> - La ejecución de tareas en la base de datos Oracle Central - La ejecución de Shell scripts en los servidores de enlace con el Banco de la Nación 	
13	Monitoreo	Componente de monitoreo en base indicadores en series de tiempo. Para ello envía las solicitudes por HTTP (métricas) y lo almacena en una base de datos de serie de tiempo, en base a estos indicadores se puede identificar la disponibilidad y estado de los diferentes componentes.	ARQ.MON.INS (Instalación) ARQ.MON.OPE (Operación)
14	Explotación Gráfica del Monitoreo	Componente que permite mostrar gráficamente la información de las métricas obtenidas con el componente de monitoreo.	ARQ.GRAF.INS (Instalación) ARQ.GRAF.OPE (Operación)
15	Gestión de Logs	Componente que centraliza los logs de ejecución y de operación de los diferentes componentes.	ARQ.LOGS.INS (Instalación) ARQ.LOGS.OPE (Operación)
16	Servidor de archivos	Componente que proporciona un lugar de almacenamiento centralizado para los archivos en sus propios soportes de datos. Por ejemplo, se utiliza para guardar archivos .pdf, .jpg que formen parte de los procesos de negocio SIAF y SIGA	ARQ.FSERV.INS (Instalación) ARQ.FSERV.OPE (Operación)
Capa de persistencia Definitiva			
17	Base de Datos ORACLE - Central	Este componente legado contiene la información centralizada de SIAF. Contiene también procedimientos almacenados con lógica de negocio.	ARQ.BDR.NOM.ORACLE (Estándar de nomenclaturas) ARQ.BDR.DES.ORACLE (Estándar de programación en PLSQL)
Capa de Seguridad			
18	Repositorio de Código Fuente	Contiene el código fuente de las aplicaciones tanto de front como de backend. Utiliza una base de datos relacional.	ARQ.GI.INS (Instalación) ARQ.GI.OPE (Operación)

N°	Componente	Descripción	Documentos de Apoyo
19	Analizador de código estático	Forma parte del pipeline de despliegue y se encarga de analizar el código fuente en busca de vulnerabilidades	DEVOPS.SEG.INS (Instalación) DEVOPS.SEG.OPE (Operación)

Figura 12: Componentes de arquitectura con tecnología de implementación



Elaboración: propia

V.2.2 Documentación de apoyo a la arquitectura referencial

A continuación, se listan los documentos que describen la implementación, pruebas, análisis comparativos o definiciones del conjunto de componentes de arquitectura.

N°	Documento	Descripción	Código del Documento
1	Documento de Despliegue	Contiene el detalle de las máquinas virtuales, sus IP, sus características de	ARQ.DESPLIEGUE

		<p>HD, RAM, Cantidad de vCPU, el ambiente que atiende, y su relación entre ellos. Este documento se mantendrá permanentemente actualizado, con la finalidad que refleje el estado de la plataforma SIAF</p>	
2	Adopción de tecnología SPA	<p>Contiene los motivos para el uso de la tecnología SPA en esta arquitectura de referencia</p>	ARQ.SPA.Analisis
3	Escenario de casos de prueba para la continuidad y recuperación del negocio	<p>Describe los Escenarios y Pruebas de Continuidad del Negocio de los componentes de Arquitectura. Esto busca responder las preguntas:</p> <ul style="list-style-type: none"> • ¿Cómo se comporta la solución en su totalidad? • ¿Cuáles son los puntos que ocasionan un fallo global de la solución, y cuales un fallo parcial? • Acciones necesarias para mitigar el riesgo de los fallos globales. • ¿Cuál es la estrategia general de recuperación de los componentes? 	<p>ARQ.PRB.CONT.PruebasContinuidad_De_Negocio.docx</p> <p>ARQ.PRB.RECUP.PruebasContinuidad_de_Recuperación.docx</p>
4	Escenarios y pruebas de convivencia con SIAF Cliente	<p>Describe los escenarios de pruebas de convivencia de los componentes de la Nueva Arquitectura con el sistema legado SIAF Cliente</p>	ARQ.PRB.CONV.Pruebas_de_Convivencia.docx
5	Gestión de opciones de menú y accesos de usuarios a la plataforma	<p>Describe la operación del menú, acceso por usuario, y creación del menú.</p>	ARQ.GST.MENU.Gestión.opciones.Menu.docx
6	Generación y Explotación de logs de auditoria	<p>Contiene los lineamientos que permiten identificar las transacciones realizadas por los usuarios una vez que superaron la autenticación, permitiendo tener los indicadores:</p> <ul style="list-style-type: none"> • Cantidad de ingreso de un usuario al sistema • Detalle de los ingresos del usuario al sistema 	<p>ARQ.AUD.GEN.Generación.Logs.Auditoria.docx</p> <p>ARQ.AUD.EXPL.Explotación.Auditoria.docx</p>

		<ul style="list-style-type: none"> • Tiempo de las operaciones, desde que realiza el registro hasta que se aprueba el registro. • Otros datos de auditoría 	
7	Propuesta de seguridad perimetral para los componentes de arquitectura	Contiene la propuesta a seguir como guía de la seguridad perimetral de los componentes de arquitectura.	ARQ.SEG.PERIM.Explotación.Auditoria.docx
8	Propuesta de seguridad red para los componentes de arquitectura	Contiene la propuesta a seguir como guía de la seguridad de red de los componentes de arquitectura	ARQ.SEG.RED.SeguridadRed.docx
9	Procedimiento para el envío de mensajes	Contiene el procedimiento para la generación y envío de mensajes por correo electrónico desde la plataforma SIAf	ARQ.MNSJ.OPE.EnvioMensajesEmail.docx
10	Procedimiento para la generación de notificaciones al usuario	Contiene el procedimiento para la generación, almacenamiento y visualización de las notificaciones que se muestran al usuario final a través de los formularios frontend. Esto implica habilitar una pestaña especializada a nivel del proyecto "Profile" en el frontend	ARQ.NOTIF.OPE.Generación.Notificaciones.docx
11	Procedimiento para la generación de mensajes al buzón del usuario	Contiene el procedimiento para la generación, almacenamiento y visualización de los mensajes que se muestran al usuario final a través de los formularios frontend. Esto implica habilitar una pestaña especializada a nivel del proyecto "Profile" en el frontend	ARQ.BUZON.OPE.Generación.Mensajes.Buzon.docx
	Procedimiento para la generación de trazabilidad en las operaciones de negocio	Contiene el procedimiento para generar y registrar a través de un microservicio las acciones de registro, transmisión, aprobación, rechazo, y cualquiera que corresponda asociar con las funcionalidades de negocio. En el momento de la elaboración se han realizado las pruebas con la Certificación Presupuestal, pero se puede aplicar a cualquier funcionalidad de negocio. Esto implica habilitar una	ARQ.TRZ.OPE.TrazabilidadNegocio.docx

		pestaña adicional en el frontend para que se muestre el detalle de trazabilidad.	
--	--	--	--

V.2.3 Documentación sobre estándares y lineamiento arquitectural

A continuación, se listan los documentos que describen los lineamientos para la construcción, implementación y nomenclatura de los objetos que forman parte de los componentes de arquitectura.

N°	Documento	Descripción	Código del Documento
Codificación			
1	Lineamientos BackEnd	Convenciones o estándares proporcionan recomendaciones o reglas a seguir para mejorar las legibilidad y mantenibilidad del código generado para la lógica de negocio que se expone a través de APIs	COD.JAVA.BACK
2	Lineamientos FrontEnd	Convenciones o estándares proporcionan recomendaciones o reglas a seguir para mejorar la legibilidad y mantenibilidad del código utilizado para interactuar con el usuario final.	COD.ANG.FRONT
3	Lineamientos para programación en PLSQL	Convenciones o estándares proporcionan recomendaciones o reglas a seguir para mejorar la legibilidad y mantenibilidad del código en base de datos ORACLE, implementado en funciones, procedimientos y paquetes.	COD.PLSQL.BACK
Nomenclatura			
4	Nomenclatura de Objetos ORACLE	Establece convenciones de nombramiento para los objetos físicos de la base de datos ORACLE	COD.ORACLE.DB