PLIEGO DE PRESCRIPCIONES TÉCNICAS

Contratación del Mantenimiento de la plataforma Gestor Documental Corporativo de Osakidetza

NÚMERO DE EXPEDIENTE: 2025/01213

ÍNDICE Página

1 OBJETO DEL CONTRATO 4
2 ÁMBITO Y ALCANCE DEL CONTRATO 6
2.1 Organizaciones implicadas o lugar de prestación y/o ejecución 6
2.2 Alcance del contrato 6
2.2.1 Línea de Mantenimiento y Evolución 6
2.2.2 Línea de Desarrollo (evolutivo a gran escala) 8
2.2.3 Gestión del Inventario, Repositorios y Documentación 8
2.2.4 Gestión y Administración del servicio 8
3 LOTES 10
4 REQUISITOS NORMATIVOS GENERALES 11
5 REQUISITOS TÉCNICOS GENERALES 12
5.1 Plataforma tecnológica 12
5.1.1 Infraestructura tecnológica 12
5.1.2 Estándares 12
5.2 Herramienta de gestión del servicio 12
5.3 Metodología 13
6 VARIANTES TÉCNICAS 14
7 MUESTRAS 15
8 PLAN DE TRABAJO / PROGRAMA DE TRABAJO /PLAN DE PUESTA EN MARCHA / LUGAR DE ENTREGA SUMINISTRO 16
8.1 Plan de trabajo 16
8.1.1 Fases del servicio 16
8.1.2 Descripción detallada de los servicios de mantenimiento 18
8.2 Ubicación del equipo de trabajo 21
9 MODELO DE GESTIÓN DE SERVICIO 22
9.1 Modelo organizativo 22
9.2 Medios personales adscritos a la ejecución del contrato 22
9.2.1 Constitución inicial del Equipo de Trabajo 23
9.2.2 Modificaciones en la composición del Equipo de Trabajo 24
9.2.3 Veracidad de los datos 24
9.2.4 Condicionantes del equipo de trabajo ofertado 24
9.2.5 Currículo de los componentes del Equipo de Trabajo 25
9.2.6 Asignación de recursos a fases del proyecto 25
9.3 Medios materiales adscritos a la ejecución del contrato 25
10 EJECUCIÓN 26
10.1 Condiciones de ejecución 26
10.1.1 Horario de la prestación del servicio 26
10.2 Condiciones especiales de ejecución 26
10.3 Modelo de control y seguimiento 26
10.4 Programa de control 27

11 ANEXO I. RELACIÓN DE APLICACIONES 28
11.1 Repositorio Área Asistencial (DOKUGUNE) 28
11.1.1 Vista lógica 30
11.1.2 Vista Física 30
11.2 Repositorio Área Gestión y Administrativa 32
11.2.1 Vista Lógica 34
11.2.2 Vista Física 35
12 ANEXO II. DIRECTRICES Y ESPECIFICACIONES TÉCNICAS (DET) 38
12.1 Especificaciones para Desarrollo 38
12.1.1 Arquitectura de Referencia 38
12.1.2 Directrices de Desarrollo 40
12.2 Modelo de aplicaciones modernas 45
12.2.1 Especificaciones de desarrollo 55
12.2.2 Entorno tecnológico 57
12.3 Directrices tecnológicas de IOC 64
12.3.1 Entorno tecnológico 65
12.3.2 Especificaciones del cliente (maqueta) 67
12.3.3 Dispositivos de entrada/salida 70
12.3.4 Seguridad, acceso, autenticación y autorización 71
12.3.5 Monitorización 72
12.3.6 Backup/Restore 73
12.4 Interoperabilidad 74
12.4.1 Servicios Web 74
12.4.2 Gestor de Eventos 76
12.4.3 Arquitectura API REST 78
12.5 Explotación Información- Business Intelligence 80
12.6 Soluciones Móviles 81
12.6.1 Tecnologías Cliente 82
12.6.2 Tecnologías Servidor 82
12.7 Calidad de uso 83
13 ANEXO III. CUESTIONARIO DE PERSONAL 86

1 OBJETO DEL CONTRATO

Osakidetza tiene implantada una solución de Gestor Documental Corporativo, con el objetivo de disponer de una plataforma de almacenamiento, gestión y custodia de informes clínicos y documentación de naturaleza electrónica.

El Gestor Documental, cuyo núcleo es el producto Documentum (Enterprise Content Management-ECM), proporciona mecanismos de alta disponibilidad, integridad, conservación a largo plazo y gestión eficiente del almacenamiento de documentación digital en los siguientes aspectos:

- Solución corporativa orientada a la gestión del ciclo de vida del documento.
- Proporciona mecanismos de integración con los actuales sistemas de información de la Organización, fundamentalmente con las soluciones de historia Clínica de **Osakidetza**, ERP (SAP), etc.
- Motor de indexación y búsqueda de alto rendimiento.
- Garantiza la no alteración de la información, mediante mecanismos de custodia electrónica.
- Asegura la conservación a largo plazo, habilitando mecanismos de transformación y *renderización* que garantizan la accesibilidad de la información.
- Arquitectura centralizada y continuidad de servicio.

Actualmente, la organización dispone de un repositorio de documentación referente a recursos clínicos de pacientes y un repositorio de documentación referente a recursos de gestión y administrativos.

Con respecto a los recursos clínicos, indicar que los informes clínicos finalizados y validados, se almacenan en el Gestor Documental en distintos formatos (pdf, word, multimedia, etc.). La orientación es siempre hacia la asistencia; cada informe está asociado a un paciente. El personal clínico desde la estación clínica de **Osakidetza**, sistema Osabide, accede al Gestor Documental tanto para registrar nuevos informes, como para realizar consultas sobre los mismos.

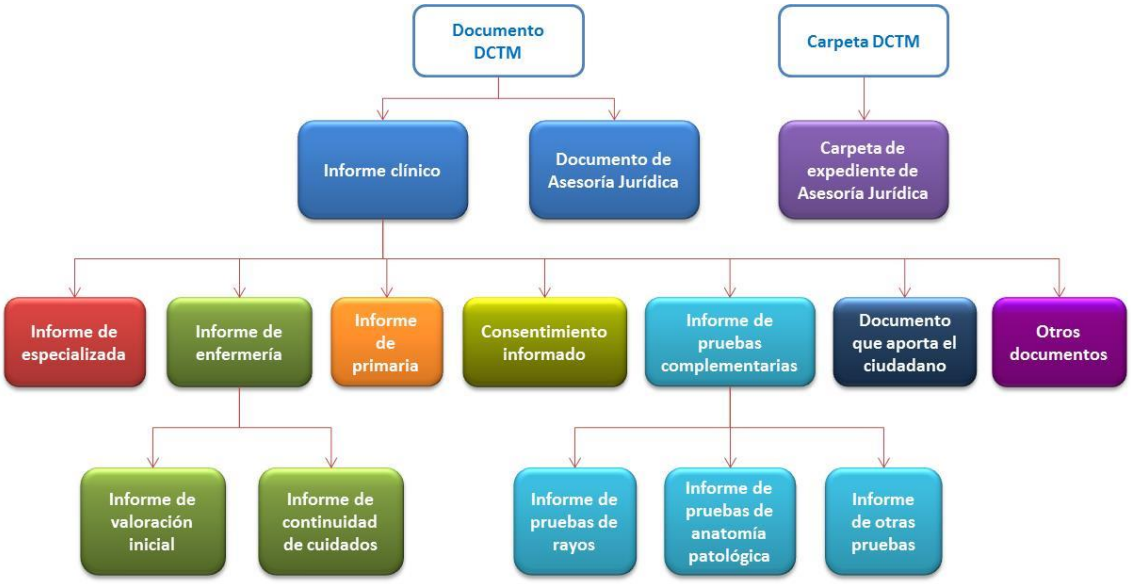
Lo mismo ocurre con los sistemas de información de gestión. El acceso al Gestor Documental se realiza desde los sistemas de información de **Osakidetza**.

Los elementos principales que sustentan el Gestor Documental son los siguientes:

- Documentum: Plataforma de gestión documental que centraliza los informes clínicos consolidados de los diferentes sistemas de **Osakidetza**.
- Servicios documentales: Conjunto de servicios que ofrecen funcionalidad a los sistemas de **Osakidetza** que trabajan con informes clínicos.
- Gestión de eventos: Conjunto de servicios que se integran con el gestor de eventos de **Osakidetza**.
- Aplicación D2 documentalistas: Aplicación configurada sobre Documentum D2 para acceso a los informes clínicos de documentalistas e investigadores.
- Extracción y Migración de documentación de sistemas candidatos a albergar sus

informes en el repositorio.

En el siguiente diagrama se representan los tipos documentales identificados en **Osakidetza** y su estructura jerárquica:



Las iniciativas mencionadas, complementan una plataforma de gestión documental completa y en continua evolución para **Osakidetza**, razón por la cual se emite el presente expediente.

2 ÁMBITO Y ALCANCE DEL CONTRATO

2.1 Organizaciones implicadas o lugar de prestación y/o ejecución

Los servicios solicitados se prestarán, principalmente, en las instalaciones del adjudicatario, aunque también se podrán prestar en las organizaciones de **Osakidetza**.

2.2 Alcance del contrato

Para la ejecución del contrato se tendrán en consideración los siguientes aspectos:

- El conjunto de aplicaciones que se relacionan en el ANEXO I formarán **la línea base** sobre el que los adjudicatarios realizarán sus trabajos.
- Esta línea base podrá ser modificada durante la vigencia del contrato, con la **incorporación y retirada de aplicaciones**, dentro de lo señalado en el presente pliego.
- Los adjudicatarios deberán garantizar los servicios que se detallan en el pliego, con la **calidad y condiciones de nivel de servicio** que se definen.
- El 50% del precio ofertado se dedicará a los servicios de **mantenimiento evolutivo de gran tamaño y nuevos desarrollos (Línea de Desarrollo),** con las condiciones que más adelante se explican.

Los servicios a prestar se encuadran en cuatro **líneas de trabajo**:

2.2.1 Línea de Mantenimiento y Evolución

Comprende servicios que requieren actuaciones de dimensión predecible en su mayor parte. En consecuencia, **se proveen con una plantilla predefinida** con una **dimensión base**, si bien podrá exigirse al proveedor modificaciones según los criterios y procedimientos contemplados en este pliego.

Los servicios que contempla esta línea de trabajo son los siguientes:

2.2.1.1 Mantenimiento Correctivo/Preventivo/Perfectivo

Este servicio contempla actividades propias de la operativa sobre incidencias, así como las tareas propias de la gestión de un servicio de esta naturaleza.

En el aspecto operativo, el adjudicatario se responsabilizará de la gestión y realización de las actividades necesarias para la corrección de las incidencias surgidas en las aplicaciones software incluidas en este expediente, así como la resolución de problemas e incidencias debidas a fallos en las mismas.

Las actividades que se incluyen para la realización del **mantenimiento correctivo** abarcan desde la recepción y registro de los errores e incidencias, su análisis, diagnóstico y propuesta de solución (que deberá ser aprobada por **Osakidetza**), hasta el seguimiento y resolución de las mismas. Así mismo, se incluye entre estas actividades, la interacción directa con el usuario de negocio, desde la detección de la incidencia hasta su completa resolución.

La puesta en producción de todo mantenimiento correctivo deberá ir acompañada de la correspondiente documentación en la que se indiquen claramente los errores que corrige, así como la relación de incidencias registradas que soluciona.

Cualquier actuación sobre el software motivada por un fallo o error de la aplicación será considerada siempre como actividad correctiva y en ningún caso actividad de tipo evolutivo.

La resolución de un error en producción puede motivar el **despliegue de equipos de desarrollo para edición de parche y actuaciones presenciales**.

Toda petición de mantenimiento correctivo, así como las actividades asociadas a la petición, quedarán registradas en la herramienta de gestión de incidencias y peticiones de **Osakidetza** y deberá venir acompañada de la documentación asociada correspondiente, indicando al menos la definición del problema y su solución, junto con el resultado de las pruebas realizadas.

Dentro del **mantenimiento preventivo** se incorporan todas las intervenciones periódicas con el fin de detectar posibles fallos ocultos antes que éstos aparezcan. incluye comprobación de consistencia de los datos, pruebas forzadas del software o hardware, errores en la configuración del hardware o software, incluyendo el gestor de base de datos, etc.

El **mantenimiento perfectivo** comprende mejoras en la operativa actual del software que no impiden el correcto funcionamiento de la actividad diaria y sí supone una mejora en el rendimiento y uso de los recursos.

2.2.1.2 Mantenimiento Adaptativo/Evolutivo a pequeña escala

Se describen los 2 tipos de mantenimiento:

- ***Mantenimiento Adaptativo***
  Son las modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, software de base, gestores de base de datos, comunicaciones, etc. Incluye, entre otros:
  - Cambios en el entorno de los datos o su procesamiento.
  - Cambios en la plataforma o arquitectura tecnológica.
  - Modificación de procedimientos existentes que no implican nuevas funcionalidades.
  - Exportaciones e importaciones de datos dedicados a la integración con otras aplicaciones del entorno, para mantenimiento de integridad de la información.
  - Integración con otros aplicativos a nivel de plataforma tecnológica.
  - La parametrización de aplicaciones.

  Este tipo de mantenimiento se regirá por los mismos criterios descritos en el mantenimiento evolutivo.

- ***Mantenimiento Evolutivo***
  Son las incorporaciones, modificaciones y eliminaciones necesarias para cubrir la evolución o cambio de las necesidades del usuario, es decir, la incorporación de nuevas funcionalidades a la cobertura actual del software. Incluye, entre otros:

- Cambios en los requisitos de la aplicación.
- Modificaciones derivadas de cambios en la normativa.
- Modificaciones de alcance limitado que supongan mejoras del aplicativo y por tanto incorporables a la versión base.

Se trata actividades relacionadas con el desarrollo evolutivo, cuyo tamaño y complejidad no son excesivos.

2.2.2 Línea de Desarrollo (evolutivo a gran escala)

Comprende las actuaciones necesarias para abordar el mantenimiento evolutivo de gran tamaño y los nuevos desarrollos derivados de la reingeniería o el desacoplamiento de funciones de las aplicaciones objeto del contrato. Depende de las necesidades de **Osakidetza** durante la vigencia del contrato y requerirán **recursos variables a lo largo del contrato** que se adecuarán a las necesidades de cada momento.

En el desarrollo de esta línea de trabajo el proveedor deberá nombrar un jefe de proyecto, como responsable único de cada demanda ante **Osakidetza**, estableciendo los plazos, hitos y entregables del proyecto.

En consecuencia, **la facturación será variable y la cuantía aplicable cada mes se determinará en función del trabajo realizado y los objetivos alcanzados**.

2.2.3 Gestión del Inventario, Repositorios y Documentación

Dada la naturaleza y características del servicio objeto del contrato se hace imprescindible contar con un inventario detallado de las aplicaciones que conforman su alcance. Por tanto, deberá recoger toda aquella información que se considere necesaria y suficiente para la gestión adecuada del servicio.

El inventario deberá estar accesible desde las instalaciones de **Osakidetza**.

Será responsabilidad del adjudicatario mantener actualizado en todo momento el inventario de aplicaciones, y, por ende, sus correspondientes fichas descriptivas.

Además del inventario, y específicamente en las aplicaciones consistentes en desarrollo propio, es imprescindible disponer de todos los artefactos de cada aplicación, necesarios para la realización de las actividades de mantenimiento encomendadas al adjudicatario, entre otros, el código fuente, los contenidos estáticos, los scripts de creación de base de datos, los ficheros de configuración, etc.

Se subraya también la necesidad de generar la documentación y elementos adicionales fijados por el Servicio de Implantación de **Osakidetza** para cada aplicación.

El incumplimiento sobre la obligación del mantenimiento y actualización del inventario, del repositorio y de la documentación generará la no aceptación del trabajo.

2.2.4 Gestión y Administración del servicio

El suministrador se responsabilizará de la gestión del contrato, de la gestión del servicio, del seguimiento y reporte a los distintos comités, de las actividades relacionadas con el aseguramiento de la calidad y la mejora continua del servicio, gestión de la organización

estructural y de trabajo de sus equipos y de generar la documentación necesaria.

Se contemplan principalmente los siguientes tipos de funciones:

- Priorización, coordinación, supervisión y seguimiento general de los servicios incluidos en el contrato.
- Coordinación con unidades funcionales de **Osakidetza.**
- Coordinación y supervisión de integraciones y estándares.
- Documentación, información y gestión del conocimiento.

Se incluyen en esta línea de trabajo las actividades relativas a la gestión y seguimiento del propio servicio, cuyos indicadores e informes deben ser reportados al comité de seguimiento para su aprobación y control.

Se deberán cumplimentar mínimamente los informes detallados en el apartado de "Control y seguimiento" del presente pliego de bases técnicas, valorándose tanto la incorporación de nuevos indicadores como informes adicionales que proponga el licitador y que contribuyan a la mejora en la gestión del servicio.

3 LOTES

No.

4 REQUISITOS NORMATIVOS GENERALES

El marco regulatorio de los trabajos objetos de esta contratación está conformado, en particular, por las siguientes normativas:

- Normativa en materia de Salud.
- Normativa de Protección de Datos.
- Normativa de Seguridad de la Información.
- Normativa de Administración Digital.

5 REQUISITOS TÉCNICOS GENERALES

5.1 Plataforma tecnológica

5.1.1 Infraestructura tecnológica

Osakidetza proveerá la infraestructura (hardware y software) necesaria para facilitar la prestación del servicio.

Con carácter general, los entornos de los sistemas a utilizar durante el proyecto, serán los detallados a continuación:

- **Entornos de Desarrollo:**
  - Se solicitará por parte de la empresa que configure un entorno de desarrollo para la realización del trabajo de manera remota, los costes de conexión, infraestructura (hardware y software) que fuesen necesarios, correrán por cuenta del suministrador.

- **Entornos de Pre-producción/certificación (integración):**
  - **Osakidetza** proveerá estos entornos y se hará cargo de su gestión.
  - Su finalidad es la de las pruebas de carga y certificación de arranques o nuevas versiones de productos.

- **Entornos de Producción:**
  - **Osakidetza** proveerá estos entornos y se hará cargo de su gestión.
  - En caso de que se conceda acceso a los entornos de producción a usuarios técnicos, el acceso se realizará bajo las condiciones previstas en este contrato con respecto al registro de accesos y las circunstancias que puedan justificar ese acceso.

Todos los entornos de trabajo se ubican en el Centro de Proceso de Datos (en adelante CPD) de EJIE.

Dado que, a priori, el equipo de trabajo realizará la mayor parte de su trabajo en oficinas propias, el adjudicatario deberá asumir la instalación de las infraestructuras requeridas para el desarrollo del servicio, esto incluye:

- Aislamiento total (desde nivel de red) de los puestos informáticos del equipo de trabajo del resto de la red del adjudicatario.
- Dotación de línea dedicada, de caudal suficiente, para la conexión a los entornos de preproducción y producción necesarios.

5.1.2 Estándares

En el ANEXO II, Directrices y Especificaciones Técnicas, se describen, a título informativo, los estándares actuales, si bien estos podrán ser modificados por **Osakidetza** durante el contrato.

5.2 Herramienta de gestión del servicio

Las herramientas de gestión a utilizar serán las siguientes:

- La herramienta utilizada por el CAU de **Osakidetza**, para registro y gestión de las consultas e incidencias técnicas y funcionales de los aplicativos, es el producto **HP-Service Manager.**

- La herramienta utilizada en **Osakidetza** para la Gestión de la Demanda y el flujo de Mantenimiento, es el producto **HP-PPM** (Project and Portfolio Management).

En caso de requerirse la integración de las herramientas actuales y futuras de **Osakidetza**, con la herramienta propia del proveedor, los costes derivados de esta integración correrán a cargo del adjudicatario.

5.3 Metodología

Deberá detallarse el marco metodológico a utilizar para la provisión del servicio.

6 VARIANTES TÉCNICAS

No.

7 MUESTRAS

No.

8 PLAN DE TRABAJO / PROGRAMA DE TRABAJO /PLAN DE PUESTA EN MARCHA / LUGAR DE ENTREGA SUMINISTRO

8.1 Plan de trabajo

Se deberá describir el plan de trabajo detallado para desarrollar las tareas solicitadas, en el plazo de tiempo establecido y con la garantía de calidad requerida. Este plan de trabajo deberá detallar los servicios a prestar.

8.1.1 Fases del servicio

Se identifican las siguientes fases:

8.1.1.1 Fase de transición

El objetivo de esta fase es la recepción de los elementos básicos e imprescindibles para la prestación del servicio, entre **Osakidetza** y el suministrador entrante.

Como primeras tareas de la fase de transición, el proveedor entrante deberá realizar las siguientes:

- Implantación completa del plan de infraestructuras.
- Presentación de un plan detallado de calidad, que deberá ser aprobado por **Osakidetza**.
- Presentación de un plan para, si se considera importante, completar los recursos humanos ofrecidos en su oferta con otros perfiles de mayor conocimiento específico.
- Cualquier otro condicionante necesario para la ejecución del proceso de transición del conocimiento y del plan de activación del servicio.

El suministrador entrante pondrá en conocimiento de **Osakidetza** la organización con la que proporcionará el servicio, así como la asignación de recursos y confirmación de roles, tareas y responsabilidades, necesarios para la gestión del servicio.

Esta fase se ejecutará de acuerdo al plan detallado de actividades del proceso de transferencia de conocimiento y al plan de activación del servicio, ambos realizados por el suministrador entrante en la fase de transición.

El suministrador entrante tiene obligación de documentar todas las actividades del proceso de transición, así como conseguir la aprobación de inicio del servicio.

Dentro de la fase de transición y como parte de la transferencia de conocimiento, el suministrador del servicio deberá adquirir y mantener posteriormente la información necesaria para la prestación del servicio.

En el ámbito de las **aplicaciones que consistan en desarrollo propio**, será responsabilidad del adjudicatario reinstalar en sus propias dependencias el conjunto de aplicaciones, incluyendo ya no solo el código fuente, sino también el conjunto de artefactos adicionales que la conforman (contenidos estáticos, scripts de base de datos, la base de datos, etc.) proveyéndose así de la autonomía suficiente para realizar las actividades de mantenimiento que le serán encomendadas.

Deberá proveerse además de una conexión simétrica con suficiente ancho de banda

como para soportar el tráfico que se genere durante el tiempo de resolución de las actividades de mantenimiento que le sean requeridas.

El coste de las licencias necesarias para poder realizar las tareas de mantenimiento en sus propias instalaciones correrá a cargo del adjudicatario.

Se iniciará también en esta fase la gestión del inventario de aplicaciones, así como la de los repositorios de artefactos software, y de documentación.

Una vez finalizada la fase de transición del servicio se procederá a la transferencia de la responsabilidad, que marcará el inicio de la fase de prestación regular. **Osakidetza** realizará una comprobación formal de la capacitación del suministrador entrante para asumir el servicio, así como la revisión de la documentación actualizada y/o generada y el acta de conformidad firmada por el suministrador entrante tras la fase de transición.

Las ofertas indicarán los plazos en que el proveedor se compromete a las siguientes acciones:

- Entrega de la documentación actualizada.
- Disponibilidad de las herramientas necesarias para la prestación y gestión del servicio y su integración conforme a lo exigido en el pliego.
- Implementación de los procedimientos necesarios para la prestación y gestión del servicio.
- Equipo base establecido completamente operativo.

El suministrador entrante recibirá, si no lo ha recibido durante las etapas anteriores, la documentación asociada a las aplicaciones transferidas, el código fuente de las aplicaciones transferidas y todas las peticiones de servicio especificados en el presente pliego.

La lista y priorización de peticiones serán realizadas directamente por **Osakidetza** en función de las necesidades.

8.1.1.2 Fase de prestación regular

La fase de servicio regular comenzará a la finalización de la fase de transición y finalizará con el inicio de la fase de devolución del servicio.

Durante esta fase, tanto **Osakidetza** como el suministrador entrante, podrán proponer las adaptaciones a los elementos del modelo que estimen oportuno.

En caso de que esto suceda, la parte solicitante deberá generar un informe que justifique la necesidad y los beneficios previstos de dicha adaptación.

El suministrador prestará el servicio bajo su plena responsabilidad, resolviendo las incidencias y peticiones existentes.

El suministrador entregará los informes acordados, que permitan realizar un seguimiento del servicio prestado.

Durante la fase de prestación del servicio se aplicarán las condiciones generales definidas en el presente Pliego.

8.1.1.3 Fase de devolución

Antes del cese o finalización de contrato, el suministrador estará obligado a devolver el control del servicio a **Osakidetza** y/o al suministrador o suministradores que ésta determine. Con anticipación suficiente al inicio de la fase de devolución del servicio, se hará una evaluación y planificación de todas las actividades.

El suministrador deberá realizar el proceso de transición de salida, conforme a la metodología que **Osakidetza** determine, responsabilizándose del cumplimiento de los siguientes puntos:

- Garantizar la viabilidad del proyecto.
- Asegurar que se mantienen los servicios a **Osakidetza** durante el traspaso del control de servicios.
- Colaborar activamente con **Osakidetza** y con el futuro suministrador entrante durante este proceso, para facilitar la transición de los servicios sin causar perjuicios.
- Entregar una planificación detallada de la transición para hacerse cargo el suministrador entrante por completo del servicio, incluyendo los tiempos de solape con los recursos existentes que serán remplazados, así como, los perfiles de las personas que lo llevarán a cabo.
- Asegurar la devolución de cualquier dato de carácter personal empleado en las pruebas.
- Incluir cualquier otra documentación que estime oportuna.
- Entrega al suministrador entrante de la documentación técnica de cada uno de los sistemas. Dicha documentación deberá incluir los mecanismos de integración de sistemas que garanticen aislar al resto de sistemas de **Osakidetza** de cualquier cambio que se considere necesario.

8.1.2 Descripción detallada de los servicios de mantenimiento

El conjunto de aplicaciones que se relacionan en el ANEXO I formarán **la línea base** sobre el que los adjudicatarios realizarán el servicio de mantenimiento.

Esta línea base podrá ser modificada durante la vigencia del contrato, con la **incorporación y retirada de aplicaciones**, tal y como se detalla a continuación.

Para la incorporación de nuevas aplicaciones se establecen las siguientes responsabilidades:

- Se mantendrá el modelo utilizado para la asunción del servicio, es decir, realizando las fases de transición y prestación regular, si bien su duración será acorde con la importancia y complejidad de la aplicación.
- El suministrador deberá tener acceso a la información sobre el estado de las aplicaciones, las incidencias existentes, incidencias resueltas, documentación funcional y técnica y aquella información que ayude a agilizar el traspaso de conocimiento de la nueva aplicación.
- El suministrador aportará un plan de trabajo y la estimación de proceso de inclusión de nuevas aplicaciones y su impacto dentro de la Dimensión Base. **Osakidetza** tiene la responsabilidad de ajustar y validar dicho plan.
- A esta nueva aplicación se le aplicarán los parámetros indicados dentro de los acuerdos a nivel de servicio, aunque por un periodo transitorio acordado no serán causa de deducción. Transcurrido este periodo, la aplicación pasará a formar parte de la línea de servicio regular.

Para la retirada de aplicaciones, el suministrador facilitará el traspaso de conocimiento tal y cómo se detalla en la fase definida dentro de este pliego como Fase de Devolución.

La Dimensión Base se revisará periódicamente con el suministrador. La frecuencia definida para dicha revisión es la siguiente:

- Quincenalmente durante las Fases de Transición.
- Trimestralmente durante la Fase de Prestación regular del servicio.

A continuación, se definen los servicios solicitados, a los que deberán ajustarse los licitadores en su oferta.

8.1.2.1 Línea de mantenimiento y evolución

8.1.2.1.1 Línea de Mantenimiento correctivo

Inicialmente, las aplicaciones sobre las que se realizarán estos servicios son las que figuran en el Anexo I, si bien esta relación puede ser modificada durante la vigencia del contrato por diferentes motivos, entre otros:

- Incorporación de aplicaciones que finalizan su desarrollo y puesta en producción.
- Incorporación de nuevas aplicaciones que asume **Osakidetza**.
- Retirada de aplicaciones por obsolescencia, sustitución por nuevos desarrollos o falta de adecuación a nuevas situaciones.

Una vez incorporadas las aplicaciones, regirán las mismas condiciones que para el resto.

El suministrador se responsabilizará de la gestión y realización de las actividades necesarias para la corrección de las incidencias surgidas en las aplicaciones software objeto del contrato, la resolución de problemas e incidencias debidas a fallos en las mismas, y la resolución de peticiones.

Las actividades que se incluyen para la realización del mantenimiento correctivo abarcan desde la recepción y registro de los errores e incidencias, su análisis, diagnóstico y propuesta de solución (que deberá ser aprobada por **Osakidetza**), hasta el seguimiento y resolución de los mismos. También se incluyen como responsabilidad del suministrador los desarrollos necesarios para corregir los datos erróneos por el mal funcionamiento de la aplicación.

La actividad de la línea base correctiva estará directamente ligada con la resolución de los problemas detectados durante la explotación de las aplicaciones, lo que implicará actualizaciones al código y actividades para la recuperación de estados estables, y que deberán ser sincronizadas con las actividades de desarrollo de cambios y nuevas versiones que se lleven a cabo sobre las mismas.

El suministrador participará en la gestión de la incidencia con **Osakidetza** con el fin de garantizar un menor impacto en el servicio. En el caso de considerarse necesario por parte de **Osakidetza**, ésta solicitará el establecer una fase de aceptación de la solución técnica propuesta por el Suministrador.

Cualquier actuación sobre el software motivada por un fallo o error de la aplicación será considerada siempre como actividad correctiva y en ningún caso actividad de tipo evolutivo.

Cada modificación realizada dentro de esta línea de mantenimiento correctivo deberá venir acompañada de la documentación asociada correspondiente, que constará de al menos: documentación del problema y de la solución y documento que recoja la propuesta y posterior resultado de ejecución de pruebas unitarias, de conjunto y de carga, cuando **Osakidetza** así lo determine.

La puesta en producción de una modificación de tipo correctivo en cualquier desarrollo quedará supeditada a la aceptación final de **Osakidetza**.

Toda petición de mantenimiento correctivo, así como las actividades asociadas a la petición, quedarán registradas en la herramienta de gestión de **Osakidetza**. El proveedor deberá hacer uso de la misma para la gestión del flujo de vida de la petición, siguiendo el procedimiento establecido por **Osakidetza**.

La actividad de mantenimiento correctivo está asociada a la puesta en Producción, por lo que deberá seguir los procedimientos definidos por **Osakidetza**. El Suministrador proporcionará mecanismos de escalado para la resolución de incidencias adicionales en las 48 horas posteriores a la implantación.

El equipo asignado a estas tareas estará formado por perfiles con capacidad de análisis y diseño y con la experiencia y conocimiento técnico y funcional necesario para el correcto desempeño de su trabajo.

8.1.2.1.2 Línea de Mantenimiento adaptativo/evolutivo

Este servicio contempla las actividades relacionadas con la gestión y la mejora de un producto software en producción que no impliquen grandes transformaciones.

La decisión de si una petición es tratada dentro de esta línea de servicio es tomada únicamente por Osakidetza.

Las tareas del Suministrador se centrarán, además de las habituales de programación, y según sea necesario, el análisis funcional, diseño de la solución, parametrización de producto, realización de pruebas y lo que corresponda de las actividades de pruebas de integración.

Toda petición de mantenimiento evolutivo quedará registrada en la herramienta de gestión de **Osakidetza**. El proveedor deberá hacer uso de la misma para la gestión del flujo de vida de la petición, siguiendo el procedimiento establecido por **Osakidetza**.

La actividad de esta línea de servicio estará asociada al desarrollo y puesta en explotación, por lo que deberá seguir los procedimientos definidos por **Osakidetza**.

Cada modificación realizada dentro de esta línea de mantenimiento evolutivo deberá venir acompañada de la documentación asociada correspondiente, que constará de al menos: documentación de requisitos funcionales y documento que recoja la propuesta y posterior resultado de ejecución de pruebas unitarias, de conjunto y de carga, cuando **Osakidetza** así lo determine.

La puesta en producción de una modificación de tipo evolutivo en cualquier desarrollo quedará supeditada a la aceptación final de **Osakidetza** o quien ésta determine.

El equipo asignado a estas tareas estará formado por perfiles con capacidad de gestión de proyectos de desarrollo, de análisis y diseño y con la experiencia y conocimiento técnico y funcional necesarios para el correcto desempeño de su trabajo.

8.1.2.2 Línea de Desarrollo (evolutivo a gran escala)

Dentro de los proyectos de actualización y evolución de infraestructuras del Gobierno Vasco (EJIE), se abordará dentro de esta línea de trabajo, la actualización de versiones de los sistemas de Gestión Documental de **Osakidetza**, así como el upgrade requerido para cada componente.

8.2 Ubicación del equipo de trabajo

Sin perjuicio de lo que expresen los siguientes párrafos, **Osakidetza** se reserva el derecho de modificar la política en cuanto a ubicaciones del equipo durante el transcurso del contrato, sin que ello suponga un coste adicional para **Osakidetza**.

Con carácter general, los trabajos se realizarán en las instalaciones del adjudicatario.

Para la realización del trabajo de manera remota, los costes de conexión, infraestructura (hardware y software) que fuesen necesarios, correrán por cuenta del suministrador. Se deberá garantizar la **conectividad bidireccional** desde el proveedor a **Osakidetza**, y viceversa, entre los entornos de ambas organizaciones para facilitar la construcción, pruebas e identificación de incidencias de la cartera de aplicaciones a mantener.

Todas aquellas empresas que vayan a hacer uso de factorías de software, deberán presentar una declaración responsable de estar en disposición de acreditar la **certificación de calidad CMMI como mínimo con nivel 3**. Los niveles de calidad en este formato de prestación deberán ser equivalentes a los de trabajo presencial.

Adicionalmente, y en el caso de factorías de software, se exige al suministrador una serie de condiciones para que puedan ser aceptables:

- Titularidad del Servicio: el suministrador no podrá subcontratar a un tercero la prestación de este tipo de servicios.
- La factoría deberá de estar ubicada en territorio de la Unión Europea.
- Infraestructura: **Osakidetza** deberá conocer y aceptar expresamente las medidas de seguridad, equipamiento y comunicaciones de las instalaciones del suministrador desde las que se preste este servicio.
- La instalación y adecuación de los locales será totalmente por cuenta del suministrador.

9 MODELO DE GESTIÓN DE SERVICIO

El adjudicatario del expediente deberá definir el Modelo de Gestión del Servicio, basado en las mejores prácticas y normas existentes en el mercado (ISO 20000, ISO 27001, ITIL, etc.) y futuras evoluciones de estos estándares.

9.1 Modelo organizativo

Para la gestión y el desarrollo del servicio, los licitadores deberán detallar en su propuesta el organigrama del equipo de trabajo, identificando responsables y funciones a desarrollar según considere para la ejecución del servicio de este pliego. Se deberá detallar también el modelo de relación con **Osakidetza** durante la vigencia del contrato.

El modelo de relación deberá garantizar la flexibilidad y la adaptación del servicio a la evolución de la Organización, pudiendo cambiar durante la vigencia del contrato, en particular ante eventuales reorganizaciones.

El licitador deberá presentar en su oferta el Modelo de Relación y el Modelo Organizativo que considere más adecuado para la prestación del servicio.

9.2 Medios personales adscritos a la ejecución del contrato

Las empresas interesadas en la adjudicación del contrato, presentarán en su propuesta la descripción de su equipo de trabajo, especificando los recursos humanos que destinarán al proyecto, sus perfiles profesionales y las dedicaciones por cada una de sus categorías.

Osakidetza, controlará mediante la figura de su Director de Proyecto, el cumplimiento de los términos acordados, así como la calidad y adecuación del servicio objeto de este contrato, quedando ambos sujetos a lo que marca la Ley de Contratos de las Administraciones Públicas (texto refundido), en cuanto hace referencia a completar los términos y calidad de los productos desarrollados.

El equipo de trabajo estará formado, al menos, por los siguientes perfiles. Se establece, además, un umbral mínimo de horas por perfil para los **tres años** de duración del contrato, de la siguiente forma:

Jefe Proyecto	Consultor Documentum Funcional/ Administrador	Consultor Documentum Técnico	Consultor OpenText (Servicios Profesionales)	TOTAL horas (equipo mínimo)
330	4.224	6.336	300	**11.190**

Aquellas empresas que no alcancen el mínimo de horas exigidas por perfil, quedarán excluidas automáticamente.

Para la gestión y el desarrollo del servicio, los licitadores deberán detallar los siguientes aspectos en su propuesta:

- Organigrama del equipo de trabajo, identificando responsables y funciones a desarrollar según considere para la ejecución del servicio de este pliego.
- Detalle del equipo total de trabajo, que los licitadores consideren necesario,

identificando las personas, categoría y funciones a desarrollar para la ejecución de las tareas previstas con el alcance definido en este pliego.
- Relación nominal de los participantes, junto su correspondiente documento de currículo

El equipo de trabajo propuesto, estará formado por personal técnico con categoría profesional y nivel de especialización adecuados a las necesidades planteadas en cada momento, de acuerdo con las actividades que se vayan desarrollando, **valorándose especialmente la inclusión de profesionales con certificaciones oficiales de OpenText** como, *System Administrator Certification* y *Documentum Consultant Certification*.

El licitador debe comprometerse, en caso de ser adjudicatario, a mantener el equipo, según lo establecido en su oferta, y durante el periodo fijado en cada actividad específica.

La gestión de la carga de trabajo durante las épocas vacacionales será la misma que para el resto del periodo del contrato y estará sujeta a la planificación acordada con **Osakidetza**. No podrá reducir unilateralmente la carga de trabajo durante las épocas vacacionales.

Asimismo, el suministrador deberá asegurar la disponibilidad de recursos con los conocimientos requeridos para mantener los niveles de servicio y la planificación acordados que permita hacer frente a sus posibles contingencias imprevistas en su personal.

La modificación de alguno de los componentes del equipo adscrito a la ejecución de los trabajos, sin observar el procedimiento y requisitos exigidos que se señalan a continuación, facultará a **Osakidetza** para calificar dicha modificación como una rotación no planificada. La reiteración en el número de rotaciones no planificadas (mayor o igual al 30 % del Equipo en un año) faculta a **Osakidetza** para instar a la resolución del contrato.

El equipo será configurado y dimensionado por el licitador en base a la información suministrada en el pliego, así como a su experiencia por el tipo de actividades a realizar.

9.2.1 Constitución inicial del Equipo de Trabajo

El equipo humano a incorporar tras la formalización del contrato para la ejecución de los trabajos, deberá estar formado por los componentes relacionados en la oferta adjudicataria.

Si tras la adjudicación, se observara que el equipo de proyecto no se corresponde con el Documento de Propuesta Técnica objeto de la misma y**:**

- Caso de que el adjudicatario presente justificación escrita, detallada y suficiente, explicando el motivo que suscita el cambio, se procederá a:
  - ✓ La presentación por el adjudicatario de posibles candidatos con un perfil de cualificación técnica igual o superior al de la persona que se pretende sustituir.
  - ✓ Aceptación de alguno de los candidatos por parte de la Dirección del Proyecto de **Osakidetza**
- Caso de que se demostrase que el cambio no se corresponde con causa

justificada, de fuerza mayor y no imputable al adjudicatario, **Osakidetza** se reserva el derecho no solo a la aprobación de la persona o personas sustitutivas, sino incluso a la revisión de la adjudicación y en su caso la rescisión del pedido/contrato, si este hecho fuera elemento determinante en la mencionada adjudicación.

9.2.2 Modificaciones en la composición del Equipo de Trabajo

Osakidetza podrá solicitar el cambio de cualquiera de los componentes del equipo, con un preaviso de un mes, por otro de igual categoría, si existen razones justificadas que lo aconsejen.

Si es el adjudicatario el que propone el cambio de una de las personas del Equipo Base, deberá solicitarlo con al menos quince días de antelación y cumplir los siguientes requisitos:

- Justificación escrita, detallada y suficiente, explicando el motivo que suscita el cambio.
- Presentación de posibles candidatos para un perfil cuya cualificación técnica sea igual o superior al de la persona que se pretende sustituir.
- Aceptación por **Osakidetza** de los candidatos propuestos.

Los posibles inconvenientes de adaptación al entorno de trabajo y al proyecto debidos a las sustituciones en los componentes del Equipo, deberán subsanarse mediante periodos de solapamiento sin coste adicional, durante el tiempo necesario. El plazo de solapamiento mínimo entre el perfil entrante y el saliente será de 15 días.

Los adjudicatarios deberán garantizar que disponen de los mecanismos adecuados para minimizar la **rotación no planificada del personal que compondrá el Equipo de Trabajo,** para evitar la pérdida de conocimiento y el impacto en los niveles de servicio e imagen.

Por **rotación planificada** se entiende aquella que se comunica a **Osakidetza** como mínimo un mes antes de que se produzca, y se acompaña de un solapamiento del recurso saliente con el entrante para la adecuada transferencia de conocimiento durante un periodo no inferior a un mes.

9.2.3 Veracidad de los datos

Osakidetza se reserva la facultad de solicitar, en cualquier momento, antes o después de la adjudicación y durante el curso de los trabajos, de cualquier otro tipo de documento complementario, en orden a la comprobación de cuantos datos haya ofrecido la empresa adjudicataria, tanto respecto a la misma, como a los recursos de que disponga.

La falsedad en los mismos podrá implicar asumir penalizaciones, y en último término, podrá provocar la resolución del contrato.

9.2.4 Condicionantes del equipo de trabajo ofertado

La falsedad en el nivel de conocimientos técnicos del personal ofertado, deducida del contraste entre lo reflejado en el currículo y los conocimientos reales demostrados en la ejecución de los trabajos, podría en último término, provocar la revisión de la adjudicación y en su caso la rescisión del contrato.

9.2.5 Currículo de los componentes del Equipo de Trabajo

Se deberá adjuntar el currículo individual detallado de todos y cada uno de los componentes del grupo de trabajo propuesto para la realización de las tareas y actividades de los trabajos objeto de contratación, junto con el papel/perfil (conforme a los indicados en el cuadro anterior) que asumen en la realización descrita.

En el caso de que el profesional disponga de alguna **certificación de OpenText** en los módulos objeto del expediente, deberá adjuntarla al currículo.

La no inclusión del currículo de alguno de los participantes, puede suponer la imposibilidad de una adecuada evaluación del criterio de "*Medios adscritos para la ejecución del proyecto*", pudiendo el licitador no ser puntuado por este concepto.

Para la descripción de los recursos humanos se utilizarán el formulario que se detalla en el ANEXO III.

9.2.6 Asignación de recursos a fases del proyecto

En la fase de prestación regular, el equipo estará configurado para proveer los **servicios de mantenimiento, soporte y evolución, así como desarrollos de tamaño limitado y no excesiva complejidad, con una Dimensión Base equivalente al 50% del total de recursos ofertados** y una distribución coherente con los servicios a prestar.

Durante la fase de transición el equipo tendrá una configuración diferente y una dimensión inferior.

Los recursos necesarios **para nuevos desarrollos y evoluciones más complejas (Línea de Desarrollo), serán requeridos en función de las necesidades y computarán económicamente en el 50% restante de lo ofertado**.

9.3 Medios materiales adscritos a la ejecución del contrato

El adjudicatario deberá aportar las infraestructuras informáticas, espacios y medios técnicos necesarios para la prestación del servicio y para poder integrarse en caso necesario en las herramientas y flujos de trabajo de **Osakidetza** desde la localización desde la que preste sus servicios (hardware, líneas de comunicaciones, dispositivos, herramientas con sus correspondientes licencias, entornos de desarrollo, etc.).

10 EJECUCIÓN

10.1 Condiciones de ejecución

10.1.1 Horario de la prestación del servicio

La dedicación de los recursos ofertados será de jornada completa. El horario de trabajo podrá verse afectado por las circunstancias y necesidades en cada momento de los proyectos y sistemas de información a mantener. Por lo tanto, los licitadores deberán comprometerse a una disponibilidad horaria según lo exija la criticidad o urgencia de determinados sistemas de información.

El horario normal de los trabajos será el siguiente:

- **De 8 a 18 horas, de lunes a jueves**.
- **De 8 a 15 horas, los viernes**

En el caso de que los servicios contratados pudieran implicar para el suministrador (por razones de cumplimiento de plazos u otras razones) la decisión de realización de los servicios en régimen de turnos o en sábados o festivos, o en régimen de nocturnidad, **Osakidetza** no aceptará sobrecostes adicionales por estas circunstancias, que deberán ser absorbidos siempre por el adjudicatario.

10.2 Condiciones especiales de ejecución

La empresa adjudicataria declarará que durante la ejecución del contrato se aplicarán medidas que respeten el medioambiente a través del uso de tecnologías y equipamientos que favorezcan el ahorro energético, el consumo responsable y la promoción del reciclado de materiales.

10.3 Modelo de control y seguimiento

El propósito de las tareas de control y seguimiento es el de proveer una visión objetiva del estado actual de la prestación del servicio y determinar las posibles desviaciones a fin de aplicar las acciones correctoras que sean necesarias. En este sentido se denomina seguimiento a la evaluación rutinaria del estado, mientras que llamamos control, a la toma de los valores que definen el estado.

Los productos resultantes de esta tarea son los informes de seguimiento, un documento donde se anotan los resultados de la evaluación de una iteración de control. Existirán una serie de informes de seguimiento requeridos para el correcto análisis de la evolución del Servicio.

Se deberán incluir mínimamente tres tipos de informes de periodicidad mensual:

- **Informes de Seguimiento:** Relación detallada de consultas, incidencias, problemas y peticiones, resumen acumulado del mes actual, acumulado del mes anterior y variación neta del mes, etc.

- **Informes del Nivel de Servicio:** Grado de cumplimiento de los indicadores, evolución anual, análisis de incidencias o problemas, propuestas de actuación, etc.

- **Evolución de la Documentación Técnica:** Aplicaciones documentadas con entidades asociadas, documentación en curso, grado de avance, etc.

De lo detallado anteriormente se deriva que el licitador deberá describir en su Oferta:

- El modelo de control y seguimiento a aplicar durante la prestación del servicio, incluyendo las reuniones de seguimiento a celebrar, su objetivo, su periodicidad, los asistentes necesarios, los controles que deben ser ejecutados, etc.
- Los informes de seguimiento que deberán obtenerse en cada fase del servicio, y para cada solicitud de trabajo realizada. Se detallará su objetivo y al menos parcialmente, su contenido.
- Los indicadores de calidad del servicio a medir, y su criticidad asociada.
- Los métodos de aplicación de las posibles acciones correctoras.

10.4 Programa de control

Se realizará el seguimiento con una periodicidad mínima mensual, en base al modelo de control y seguimiento definido.

11 ANEXO I. RELACIÓN DE APLICACIONES

11.1 Repositorio Área Asistencial (DOKUGUNE)

Se describe a continuación la plataforma Dokugune del área asistencial existente actualmente en **Osakidetza** y objeto del presente mantenimiento.

Dokugune es una plataforma de gestión documental basada en la suite de productos EMC Documentum, que permite la gestión de la documentación clínica electrónica. En la actualidad, se gestionan más de 160.000.000 de informes clínicos en su repositorio.

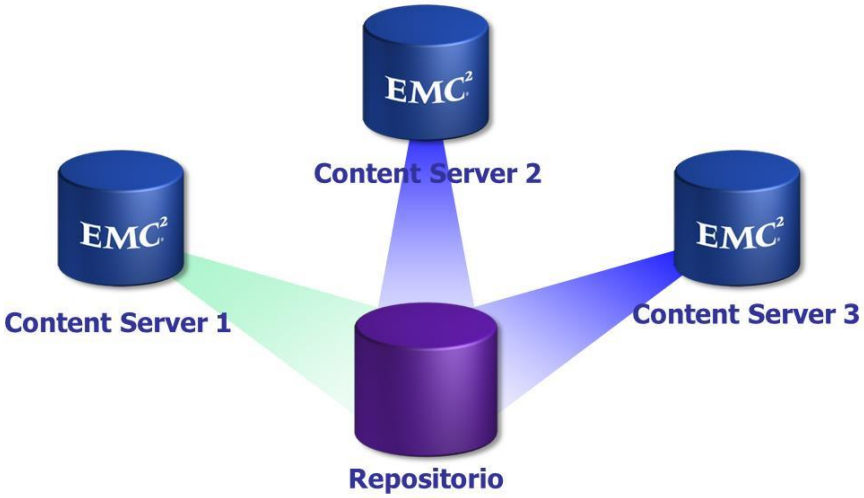
Los componentes más significativos, por su criticidad en el funcionamiento global de la plataforma son:

- **GDO**: Conjunto de Web Services desarrollados, para ofrecer a los distintos sistemas informáticos de la organización, la integración con los servicios de Gestión Documental corporativa. En la actualidad contiene integración e informes de 50 sistemas origen distintos.
- **Content Server**: como se indica en la sección anterior, es el componente central de la plataforma de gestión de contenidos.
- **Servidor de base de datos (Oracle)**: cada repositorio Documentum lleva asociado un esquema en la base de datos. La totalidad de los metadatos asociados a los objetos del repositorio se almacenan las tablas específicas dentro del esquema.
- **Almacenamiento**: todos los archivos de contenido se almacenan en uno o más sistemas de archivo, representados mediante las áreas de almacenamiento del repositorio.
- **Procesador de Eventos**: sistema que recibe y procesa los eventos publicados por los distintos sistemas asistenciales. Procesa cerca de 220.000 eventos semanales para versionar y actualizar los datos asociados a los informes de pacientes.
- **Reports**: sistema que genera de forma automática los distintos informes sobre la actividad y contenido registrado por cada sistema.
- **GDO Manager**: Aplicación desarrollada para la gestión del catálogo corporativo y Digitalización de informes. (Microservicio en Kubernetes, y desarrollo con framework Next.js sobre Node.js).
- **Alertas**: sistema que monitoriza la actividad y datos registrados en el Gestor Documental generando alertas personalizadas.

El **repositorio** es la “suma” de los metadatos y los contenidos, como se muestra en la siguiente figura:

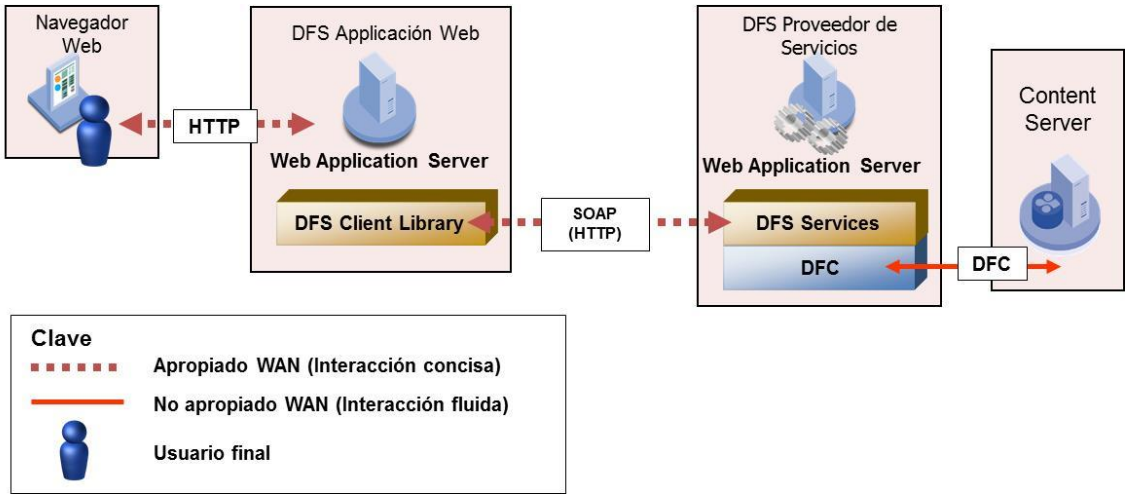


Content Server es un servicio que gestiona el repositorio. En producción se dispone de varios Content Servers a fin de proporcionar balanceo de carga a las aplicaciones, como se muestra en el esquema siguiente:



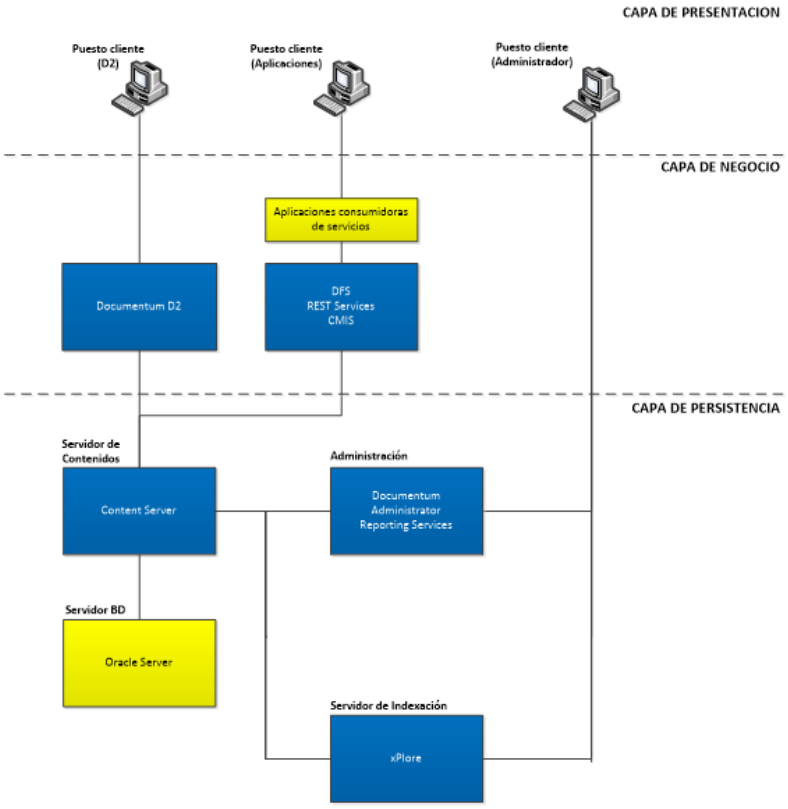
- **Documentum Foundation Services (DFS)/Documentum REST services**: son los componentes a través de los cuales las aplicaciones que implementan consumidores de servicios web acceden a Content Server (por ejemplo, en el entorno asistencial).

La interacción de los usuarios con Content Server se muestra en la figura siguiente:



11.1.1 Vista lógica

En el diagrama siguiente se muestra la disposición de los componentes de la plataforma EMC Documentum arriba descritos (color azul), incluyendo componentes y/o plataformas de terceros (color amarillo), y las interconexiones entre todos ellos.

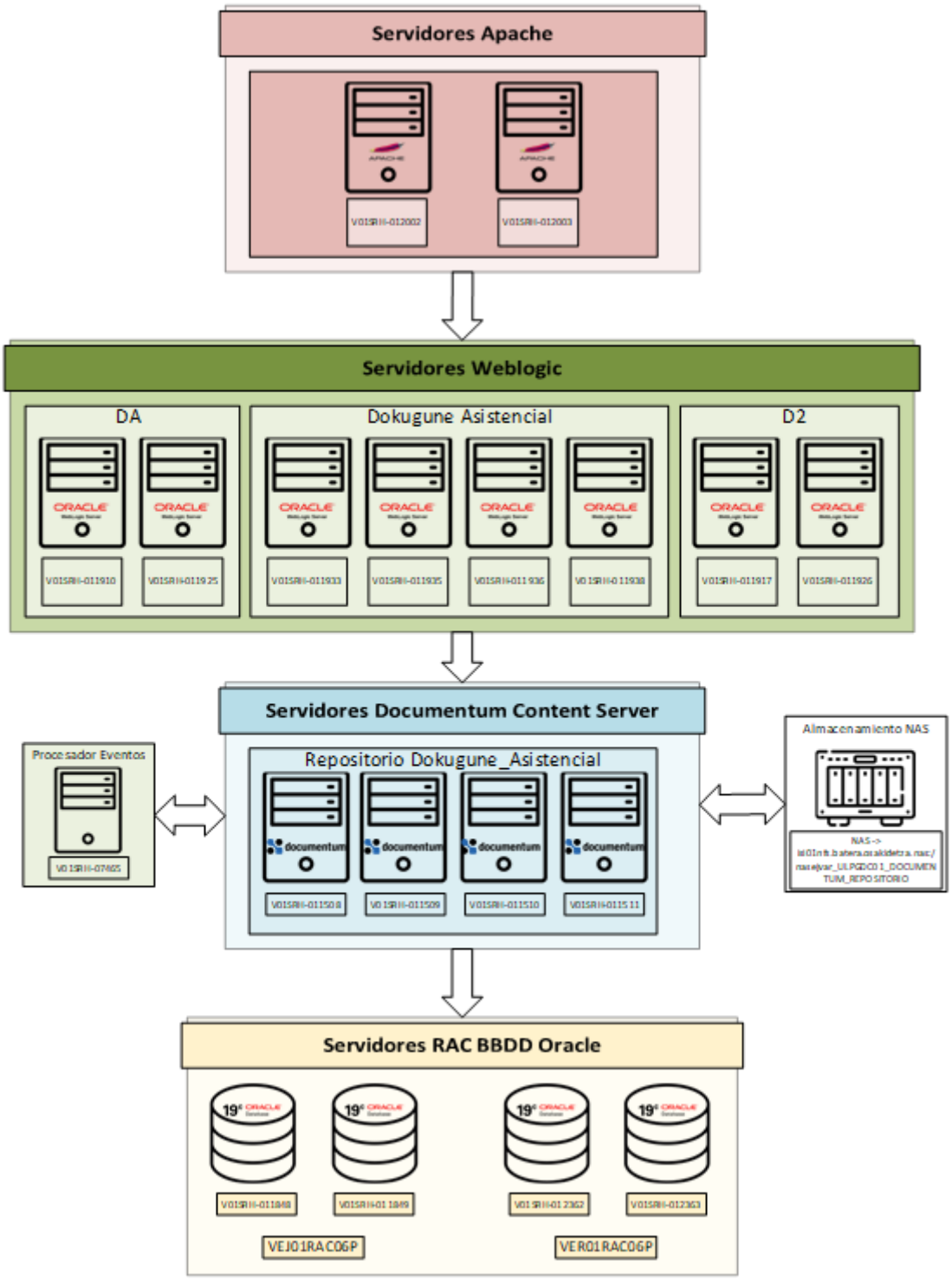


11.1.2 Vista Física

El sistema de gestión documental EMC Documentum se albergará sobre una plataforma virtual. El entorno operativo de esta plataforma es:

VMware vSphere Hypervisor (ESXi) Server

La única excepción la constituyen los servidores de base de datos Oracle, que serán albergados sobre servidores físicos y configurados con la opción Real Application Cluster (RAC) proporcionada por Oracle.



Tal y como se ha indicado anteriormente, en el marco del presente expediente, se realizará el upgrade de Documentum, así como de todos los sistemas y componentes asociados al Gestor Documental.

11.2 Repositorio Área Gestión y Administrativa

Se describe a continuación la plataforma "GDO-Varios" que existente actualmente en **Osakidetza** y objeto del presente mantenimiento.

GDO-VARIOS es una plataforma de gestión documental basada en la suite de productos EMC Documentum que permite la gestión de la documentación electrónica.

- DOKUPROF: Gestor documental para albergue de todos los informes de aplicaciones (web/SAP) relacionadas con RRHH y la información derivada de los profesionales y candidato. Contiene cerca de 3.000.000 informes.

- DOKUREG: Gestor documental para albergue de informes del nuevo libro de registro. Contiene cerca de 200.000 informes.

- DOKUSAPU: Gestor documental para el área se Servicio de Atención a Pacientes y Usuarios. Contiene cerca de 3.000.000 informes.

- GESTOREXP: Gestor documental para la gestión de expedientes. Contiene cerca de 270.000 informes.

- DOKUGMAO: Gestor Documental para el módulo de Gestión del mantenimiento de activos de Osakidetza. Contiene cerca de 10.000 informes.

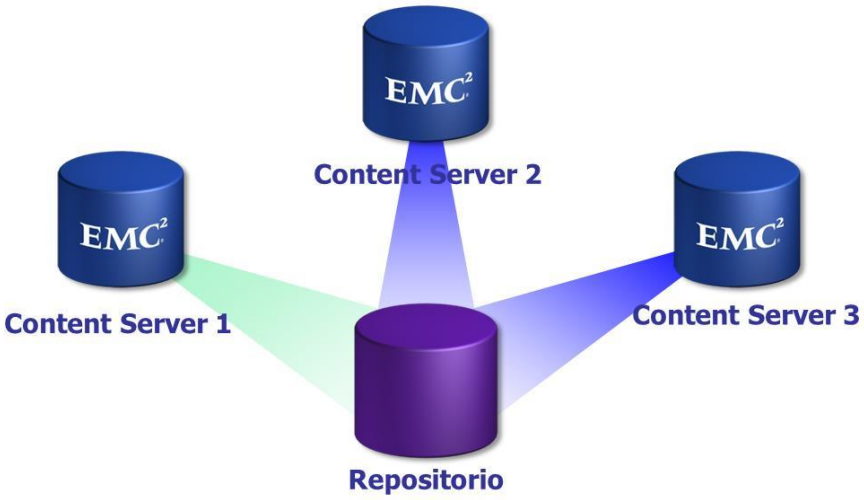
Los componentes más significativos, por su criticidad en el funcionamiento global de la plataforma son:

- **Content Server**: como se indica en la sección anterior, es el componente central de la plataforma de gestión de contenidos.
- **Servidor de base de datos (Oracle)**: cada repositorio Documentum lleva asociado un esquema en la base de datos. La totalidad de los metadatos asociados a los objetos del repositorio se almacenan las tablas específicas dentro del esquema.
- **Almacenamiento**: todos los archivos de contenido se almacenan en uno o más sistemas de archivo, representados mediante las áreas de almacenamiento del repositorio.
- **Reports**: sistema que genera de forma automática los distintos informes sobre la actividad y contenido registrado por cada sistema.

El **repositorio** es la "suma" de los metadatos y los contenidos, como se muestra en la siguiente figura:



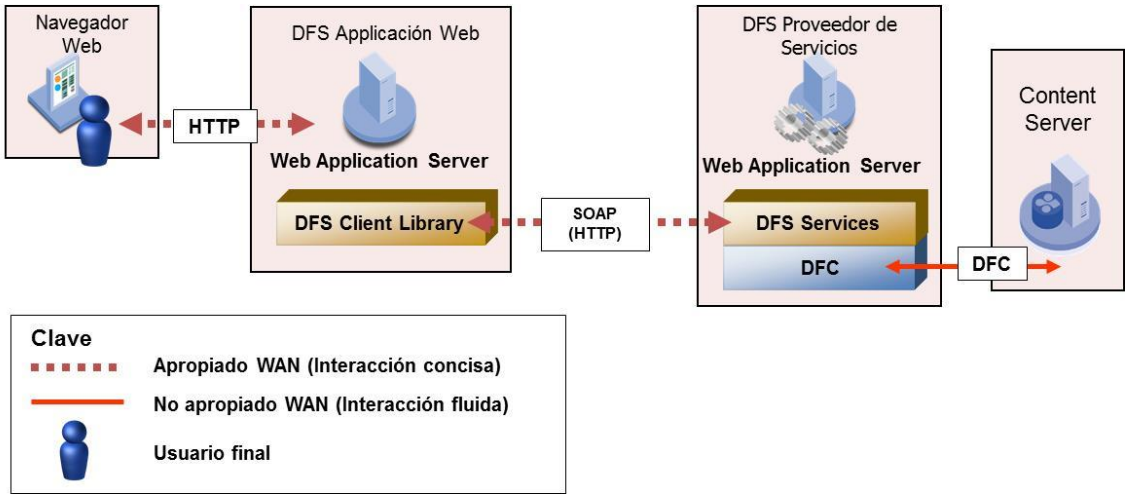
Content Server es un servicio que gestiona el repositorio. El repositorio sólo se puede acceder a través de Content Server. En general, un repositorio de producción es gestionado por 2 o más Content Servers a fin de proporcionar balanceo de carga a las aplicaciones, como se muestra en el esquema siguiente:



- **Documentum Foundation Services (DFS)/Documentum REST services**: son los componentes a través de los cuales las aplicaciones que implementan consumidores de servicios web acceden a Content Server (por ejemplo, en el entorno asistencial).

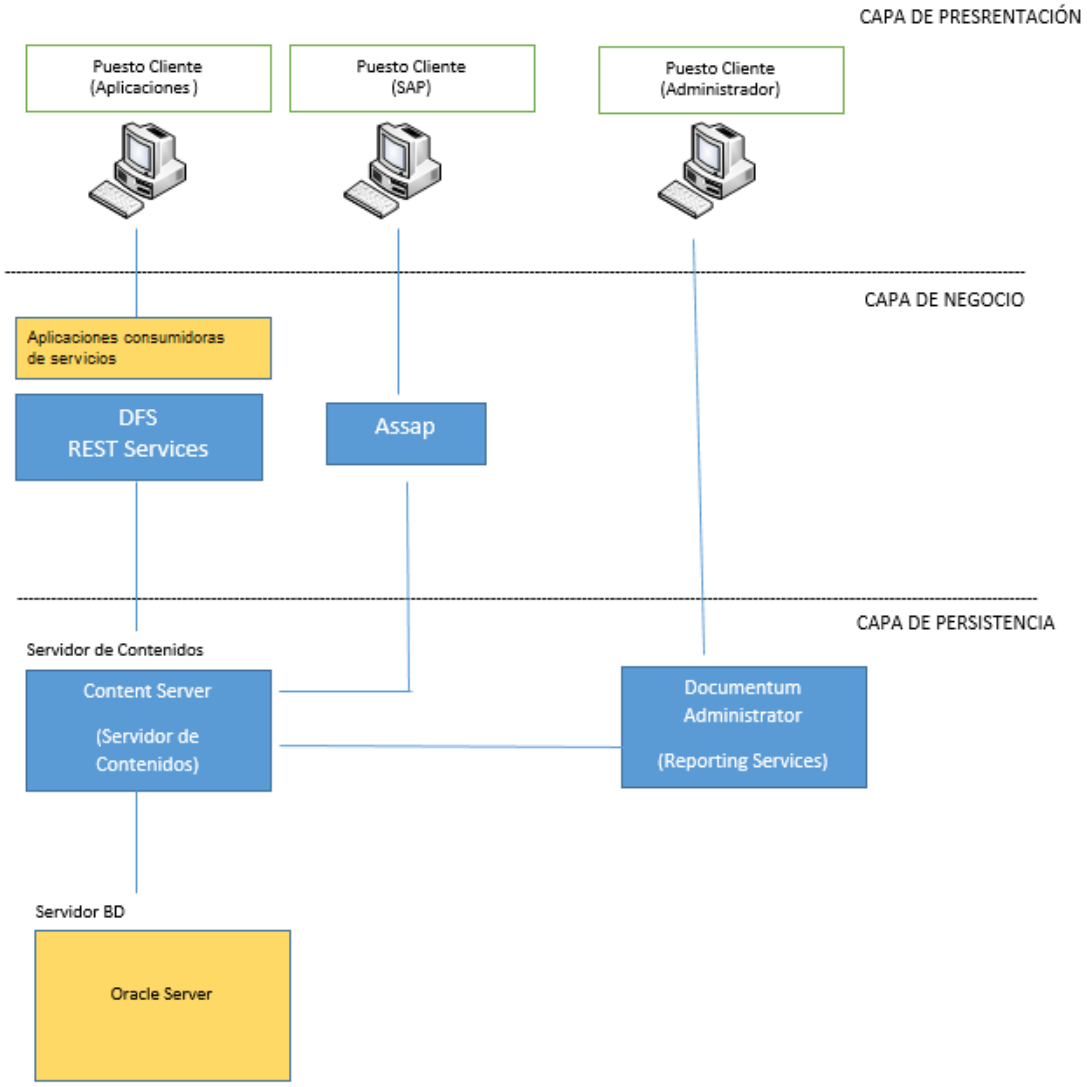
- **Assap**: componente a través de cual las aplicaciones SAP que implementan consumidores de servicios acceden a Content Server.

La interacción de los usuarios con Content Server se muestra en la figura siguiente:



11.2.1 Vista Lógica

En el diagrama siguiente se muestra la disposición de los componentes de la plataforma EMC Documentum arriba descritos (color azul), incluyendo componentes y/o plataformas de terceros (color amarillo), y las interconexiones entre todos ellos.

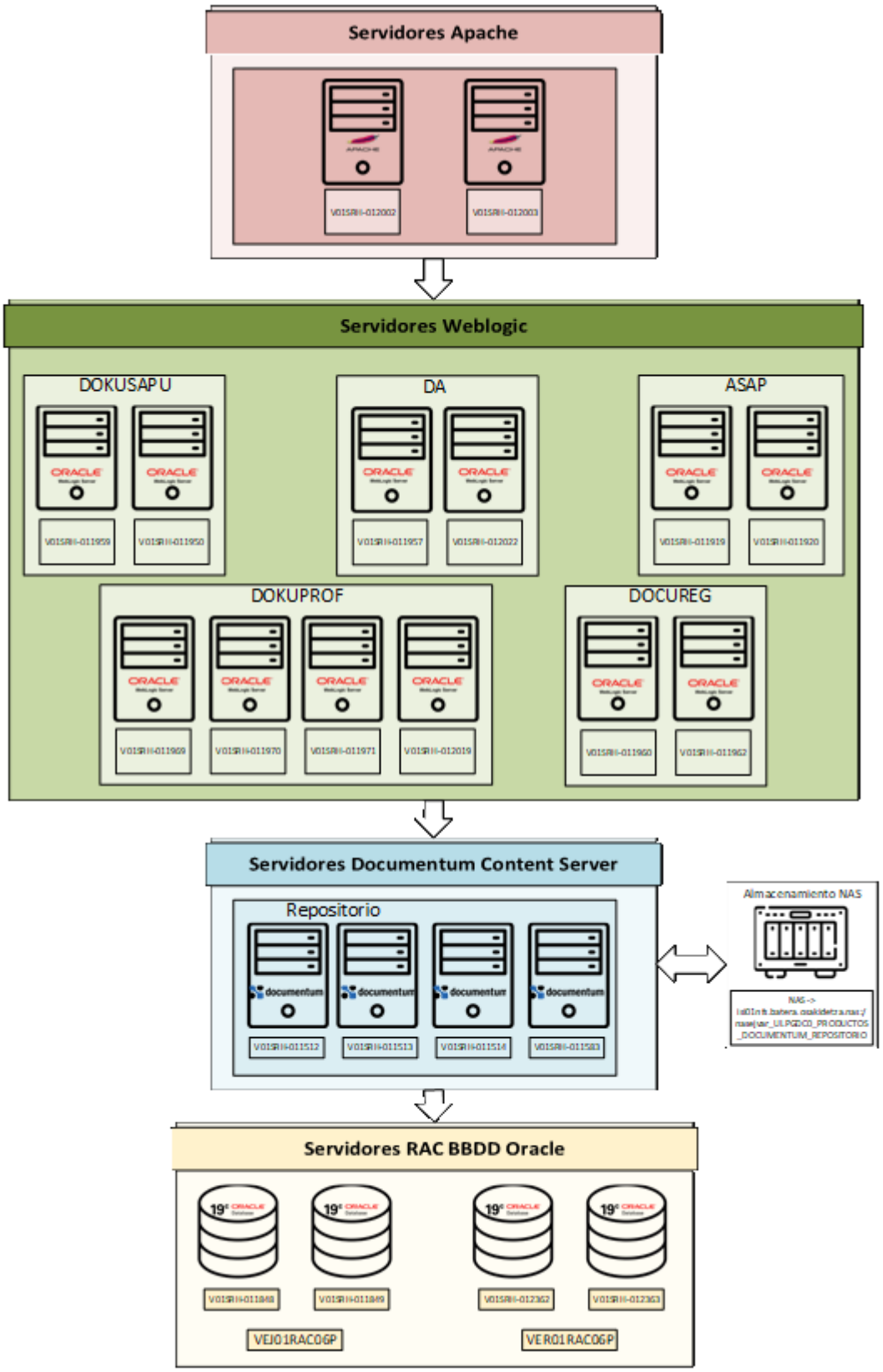


11.2.2 Vista Física

El sistema de gestión documental EMC Documentum se albergará sobre una plataforma virtual. El entorno operativo de esta plataforma es:

VMware vSphere Hypervisor (ESXi) Server

La única excepción la constituyen los servidores de base de datos Oracle, que serán albergados sobre servidores físicos y configurados con la opción Real Application Cluster ccess (RAC) proporcionada por Oracle.


Servidores Apache
V015RII-012002
V015RII-012003
Servidores Weblogic
DOKUSAPU
V015RII-011959
V015RII-011950
DA
V015RII-011957
V015RII-012022
ASAP
V015RII-011919
V015RII-011920
DOKUPROF
V015RII-011969
V015RII-011970
V015RII-011971
V015RII-012019
DOCUREG
V015RII-011960
V015RII-011962
Servidores Documentum Content Server
Repositorio
documentum
V015RII-011512
V015RII-011513
V015RII-011514
V015RII-011583
Almacenamiento NAS
Servidores RAC BBDD Oracle
V015RII-011848
V015RII-011849
V015RII-012362
V015RII-012363
VEJ01RAC06P
VER01RAC06P

Los componentes de software y versiones que se instalan en los servidores virtuales son los siguientes:

SO	EMC software	Otro software
RedHat Enterprise Linux release 8.x	Content Server v21.1	Oracle Client 19c Java 7.x.x Java 11.x.x
RedHat Enterprise Linux release 8.x	Documentum Administrator v21.1	Oracle Weblogic Server Java 11.x.x
RedHat Enterprise Linux release 8.x	Assap v21.2	Oracle Weblogic Server Java 11.x.x

12 ANEXO II. DIRECTRICES Y ESPECIFICACIONES TÉCNICAS (DET)

Este documento, DET (**Directrices y Especificaciones Técnicas V06.6),** contiene las Directrices y Especificaciones Técnicas de los Sistemas de Información de **Osakidetza** en el ámbito TIC.

12.1 Especificaciones para Desarrollo

Este apartado refleja los principales aspectos del diseño técnico de la arquitectura de referencia para aplicaciones web, definida por Osakidetza.

Una arquitectura de referencia es un conjunto de patrones diseñados y probados para ser usados en un contexto específico, en nuestro caso, en los sistemas de misión crítica de Osakidetza. Es una guía de ayuda para ser usada por los arquitectos y desarrolladores, que permite conceptualizar nuevas aplicaciones rápidamente reutilizando un diseño suficientemente probado.

La arquitectura de referencia refleja las decisiones y aspectos técnicos de diseño basados en la experiencia de Osakidetza en el desarrollo, mantenimiento y evolución de aplicaciones web. El principal propósito de la arquitectura de referencia para aplicaciones web es, por lo tanto, homogeneizar el desarrollo, despliegue y explotación en producción de las aplicaciones.

En este sentido, la implementación de la arquitectura de referencia aportará los siguientes beneficios perseguidos por Osakidetza:

- Mejorar el rendimiento de las aplicaciones.
- Garantizar la escalabilidad y la tolerancia a fallos.
- Facilitar las integraciones entre sistemas.
- Aprovechar las tecnologías estándares y capacidades proporcionadas por los recursos hardware y software existentes en la organización.
- Reducir el coste del mantenimiento (evolutivo y correctivo) mediante la simplificación del diseño de la solución y la implementación de este.

12.1.1 Arquitectura de Referencia

Una de las características generales de los proyectos con éxito es que existe un diseño previo bien fundamentado que define la arquitectura general del sistema, las soluciones de infraestructura y plataforma (por ejemplo: alta disponibilidad, recuperación frente a desastres, copia de seguridad y restauración, monitorización), y realiza un mapeo de requerimientos a elementos de la arquitectura.

Desde el punto de vista de sistemas de información, este requerimiento estratégico debe tener una base de arquitectura y diseño previo que permita cumplirlo durante la evolución del ciclo de vida.

Los **principios de la arquitectura de referencia** describen los principios fundamentales que se han considerado para el diseño de dicha arquitectura. Los principios representan los preceptos o reglas básicas que guían el proceso de diseño.

La arquitectura de referencia se describe usando tres vistas diferentes:

- **Vista lógica** o conceptual, en la que se presenta la arquitectura mediante una visión de alto nivel que permite entender los diferentes componentes que intervienen y las interacciones existentes entre ellos.
- **Vista de implementación**, en la que se proporcionan detalles adicionales sobre la tecnología y productos para la construcción de los diferentes componentes.
- **Vista de despliegue**, en la que se describe de forma pormenorizada la información del sistema relativa a la instalación y explotación que se realiza por entornos.

12.1.1.1 Vista lógica

El propósito de la vista lógica es permitir un entendimiento de alto nivel de la arquitectura de referencia y de las diferentes capas y componentes en las que está estructurada. También se define la integración con otros sistemas.

Sobre esta vista lógica se deberían plasmar todos los elementos que deben ser considerados cuando se diseña una nueva aplicación.

Las aplicaciones deben de tomar este diseño como referencia e indicar:

- Las capas de la arquitectura de referencia que se implementarán
- Servicios externos con los que interactúa la aplicación
- Clientes que usarán la aplicación
- Servicios de persistencia de datos

En la siguiente figura se muestra la **vista lógica** de la arquitectura de referencia para aplicaciones web.

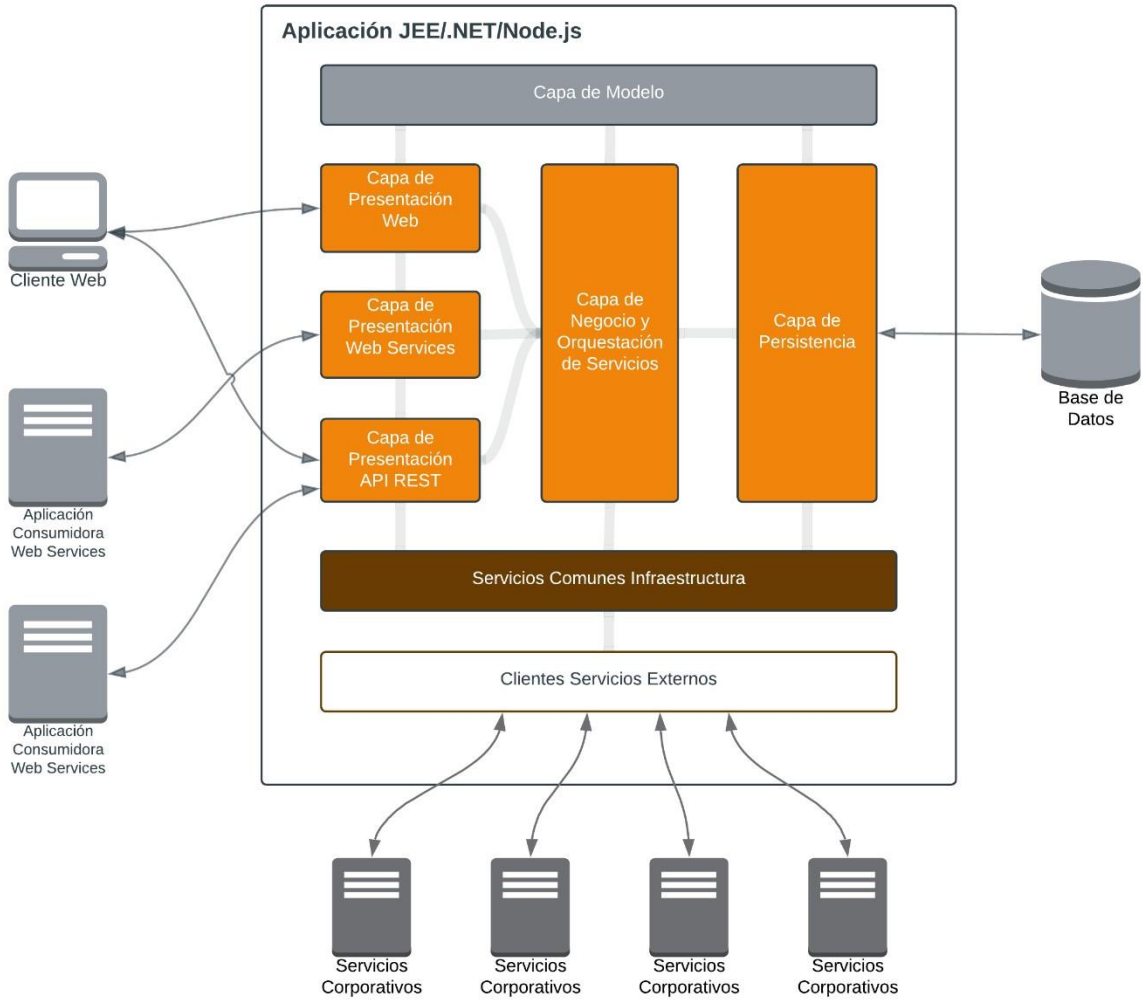


Figura 1. Vista lógica de la arquitectura de referencia

12.1.2 Directrices de Desarrollo

Se contemplan como directrices de desarrollo, las **normativas**, **estándares** y **recomendaciones** para la elaboración de un **código fuente homogéneo**, **estandarizado** y de **calidad**, con el objeto de **minimizar las tareas de mantenimiento** y la **deuda tecnológica**.

También se incorporan las especificaciones para la obtención de **sistemas de información seguros**, con un **rendimiento óptimo** y adaptado a las necesidades de las tecnologías definidas por las Arquitecturas de Software que se tomarán como referencia en los desarrollos.

A continuación, se enumeran algunas de las **directrices de desarrollo** que deberán cumplir las soluciones desarrolladas para Osakidetza:

12.1.2.1 Construcción de Aplicaciones por Capas

El principal objetivo de la construcción de aplicaciones por capas es la **separación de la lógica de negocio, de la lógica de diseño**. Su principal ventaja, es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sea necesario llevar a cabo algún cambio, sólo se necesita intervenir al nivel requerido.
El ejemplo clásico de este tipo de práctica y el que se contempla a nivel de directriz, es el consistente en separar las aplicaciones en tres capas: **Presentación**, **Negocio** y **Persistencia**.

Por lo tanto**, las aplicaciones serán desarrolladas en capas**; es decir, se diseñará las aplicaciones **separando entre los modelos de datos, la lógica de aplicación, y la interfaz de usuario**. Si optamos por esta separación de responsabilidades, podremos compartir totalmente el modelo de datos y sólo tendremos que adaptar la interfaz de usuario.

Por cada capa se consideran una serie de **directrices de desarrollo basadas en el modelo de tres capas propuesto por la Arquitectura de Referencia** de Osakidetza.

12.1.2.2 Capa de Presentación

Mediante la capa de presentación se separa la interacción del usuario respecto a la lógica de negocio.

El uso de la arquitectura en tres capas en el desarrollo de aplicaciones ha favorecido la aparición de tecnologías que facilitan la implantación de esta capa, además de un conjunto de buenas prácticas que mejoran el proceso de su implementación.

Directrices de Desarrollo para la Capa de Presentación:

Finalidad: *Presentar la información al usuario*.

La capa de presentación, también llamada "capa de usuario", deberá presentar el sistema al usuario, comunicar la información necesaria y capturar la misma en un proceso que realiza un filtrado para validar los datos respecto al formato. **No deberá procesar datos ni tomar decisiones**.

Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (comprensible y fácil de usar) para el usuario.

Modularidad: *Controlar la relación entre las vistas.*

En una vista sólo debe hacerse referencia a otras vistas relacionadas con su funcionalidad. En ejecución, se recomienda integrar la vista con un **sistema de plantillas**, para poder aumentar de esta manera la **reusabilidad** y el **encapsulamiento de funcionalidades**.

Internacionalización y localización: *Preparar las aplicaciones para diferentes idiomas y convenciones.*

Las aplicaciones estarán preparadas para que puedan adaptarse a diferentes idiomas

y convenciones (formatos de fecha, moneda, etc.), sin necesidad de realizar cambios de relevancia en el código.

Validaciones: *Realizar validaciones sobre los datos introducidos por los usuarios*.

La vista será la responsable de la validación inicial de los datos introducidos por el usuario. La validación debe ser obligatoria sólo para el caso de campos y formato. **Nunca deberá realizarse una validación de negocio desde esta capa**.

Catálogo de controles: *Elaborar un catálogo con la lista de controles.*

Se especificará, si es posible, la función del control, los posibles validadores/conversores que se le puedan asociar, los eventos que pueda lanzar y los atributos que puedan cambiarse para manipular su aspecto externo.

Los controles deberían permitir el tratamiento de componentes y eventos que posibiliten extraer de la vista toda la lógica de interfaz. La consecuencia principal de este enfoque es que **desde la vista nunca se manipularán los controles**. El objetivo es que **la construcción de la vista sea una tarea totalmente autónoma del resto de componentes de la función de negocio**.

Uso de plantillas: *Facilitar el mantenimiento haciendo uso de plantillas.*

Se debe utilizar un **framework de plantillas (templates) en la capa de presentación**, ya que de esta manera se facilita el mantenimiento del diseño de los sistemas de información. Existen frameworks que posibilitan aislar el diseño de la aplicación en un único fichero de plantilla (o más, si se necesitan). **Esto facilita la tarea de hacer modificaciones en el diseño**.

Controles de interfaz: *Utilizar un identificador único para los controles de interfaz.*

Todos los controles de interfaz deben mantener **un identificador único** para facilitar su **compresión**, **reutilización** y **batería de pruebas** del mismo.

Independencia entre capas: *Evitar el acoplamiento entre distintas capas.*

Cada capa debe tomar la responsabilidad que le corresponde. En este caso, debe evitarse que la capa de presentación realice atribuciones que le correspondan a otras capas.

12.1.2.3 Capa de Negocio

La capa de negocio expone la lógica necesaria a la capa de presentación para que el usuario, a través de la interfaz, interactúe con las funcionalidades de la aplicación.
Se utilizarán **componentes de negocio** para abstraer la lógica de negocio de otros problemas generales de las aplicaciones, como la concurrencia, transacciones, persistencia, seguridad, etc.

Directrices de Desarrollo para la Capa de Negocio:

Reglas de negocio: *Implementar las reglas de negocio en esta capa.*

Los módulos que requieran una funcionalidad deberán encontrarla en un solo lugar y única versión. No debe haber réplica de funcionalidades en ninguna de las otras capas (Persistencia y/o Presentación)

Validaciones de datos: *Garantizar el valor correcto de los datos.*

Esta capa debe garantizar que los datos requeridos para procesarla hayan sido debidamente validados, y sólo se puede iniciar el flujo del proceso de negocio si la validación resultó correcta.

Comunicación entre capas: *Definir la estrategia de comunicación entre capas.*

La comunicación entre capas se debe establecer mediante objetos del modelo. Se crearán entidades sin métodos, sólo tendrán propiedades, campos y atributos, que nos servirán para almacenar información, como puede ser la asociación de las propiedades a las columnas de la base de datos.

Transacciones: *Tratar convenientemente las transacciones de los datos.*

La capa de modelo de datos proporciona los datos al sistema, pero **las transacciones** que involucren a dichos datos **se deben manejar desde la capa de negocio.**

Excepciones: *Encapsular en la capa de negocio las excepciones y trasladarlas de forma correcta a la capa de presentación.*

El objetivo es **simplificar el manejo de excepciones** de la aplicación. Se recomienda tener bien definida una **política para el tratamiento de excepciones y disponer de una jerarquía propia de excepciones**. Como norma general, **las aplicaciones no deberán nunca extender las excepciones del sistema**, sino las de su propia jerarquía de clases.

Servicios encapsulados: Encapsular la capa de negocio en servicios.

Se debe encapsular la capa de negocio en servicios, para de esta forma **aislar la base de datos respecto de una interacción directa con el usuario**. Estos servicios de negocio conforman un "puente" entre el usuario y los datos. Responden a peticiones de usuarios u otros servicios de negocio aplicando procedimientos formales y reglas de negocio a datos relevantes.

Acceso a servicios: *Controlar el acceso a los servicios en la capa de negocio.*

El control de acceso al servicio de negocio debe hacerse en la capa de negocio, puesto que podemos tener distintas capas de presentación, como, por ejemplo: una capa de presentación web y una capa de presentación Web Services.
El control de acceso puede realizarse tanto a nivel de método, dentro de cada servicio, como a nivel de servicio vertical.

Inyección de Dependencias: Uso del patrón de diseño ‘Inyección de Dependencias’

(DI), como forma de 'Inversión de control' (IoC)

La **Inversión de Control (Inversion of Control en inglés, IoC)** es un **método de programación** en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales. En la Inversión de Control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. Es el principio subyacente a la técnica de Inyección de Dependencias, siendo términos frecuentemente confundidos.

La **Inyección de Dependencias (Dependency Injection en inglés, DI)** es **un patrón de diseño orientado a objetos**, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. La forma habitual de implementar este patrón es mediante un "Contenedor DI" y objetos planos o simples, por ejemplo, los llamados POJO. El contenedor inyecta a cada objeto, los objetos necesarios según las relaciones plasmadas en un fichero de configuración. Típicamente este contenedor es implementado por un framework externo a la aplicación.

La implementación de un patrón de diseño como la 'Inyección de Dependencias' permite gestionar de una manera eficiente las dependencias entre los diferentes componentes de una aplicación, sin tener que recurrir a un acoplamiento no deseable entre estos.

12.1.2.4 Capa de Persistencia

La necesidad de vincular los datos guardados en una base de datos relacional, con los objetos de una aplicación orientada a objetos, determina la aparición del concepto de **persistencia de objetos**. Siguiendo el paradigma de desarrollo en tres capas, **la persistencia queda recogida en su propia capa, separada de la lógica de negocio y de la interfaz de usuario**.

La persistencia de la información es probablemente la parte más crítica en una aplicación de software y permite ejecutar operaciones en el almacenamiento persistente (base de datos). El modelo de objetos de un sistema difiere en muchos aspectos del modelo relacional. La interfaz que une esos dos modelos se llama asociación objeto-relacional (ORM en inglés). La capa de persistencia encapsula el comportamiento necesario para mantener los objetos.

Directrices de Desarrollo para la Capa de Persistencia:

Asociación Objeto-Relacional: Usar un mapeador Objeto Relacional para implementar la capa de persistencia.

Se debe utilizar un mapeador Objeto-Relacional para implementar la capa de persistencia, ya que es una técnica que permite convertir los tipos de datos utilizados en un lenguaje de programación orientado a objetos y los utilizados en una base de datos relacional, lo que posibilita el uso de las características propias de la orientación a objetos.

Uso del patrón DAO: Crear una clase DAO por cada objeto de negocio del sistema

El **patrón CRUD**, reconocido como el patrón más importante del acceso a datos, indica que **cada objeto debe ser creado en base de datos para que sea persistente**. Para esto es necesario asegurar que existen operaciones que permiten a la capa inferior (de acceso a datos) leerlo, actualizarlo o borrarlo.

Mediante la implementación de DAO's pueden proporcionarse las operaciones CRUD necesarias para cada aplicación. No es obligatorio que cada DAO implemente todas las operaciones CRUD (puede no ser necesario en la lógica funcional del sistema). Por lo tanto, **se recomienda crear un DAO distinto por cada objeto de negocio en el sistema**.

Manejo de la caché: Usar la caché en las aplicaciones para reducir tiempos de lectura.

Se recomienda **utilizar un sistema de caché** en las aplicaciones, para reducir tiempos de lectura, ya que la mayoría de los accesos de lectura acceden a una pequeña parte de los datos de la aplicación. Normalmente, hay un conjunto de datos que son relevantes a todos los usuarios y que, por lo tanto, son accedidos con más frecuencia.

Concurrencia de usuarios: Permitir la concurrencia de usuarios.

Se debe permitir que varios usuarios trabajen en la misma base de datos, **protegiendo los datos de ser escritos erróneamente**. En la mayoría de los casos, se utilizará el bloqueo optimista, soportado por la mayoría de los motores de persistencia como la opción por defecto.

Referencias circulares entre objetos: Evitar las referencias circulares entre objetos de la capa de persistencia.

Se deben evitar las referencias circulares, facilitando la localización de los objetos. De esta manera, se devuelve el objeto solicitado sin necesidad de realizar un recorrido para localizarlo, lo que supone un acceso más efectivo.

Actualización en cascada: Utilizar la actualización en cascada siempre que sea posible.

Utilizar la posibilidad que ofrecen algunos frameworks para la actualización en cascada de objetos. Una actualización en cascada **permite que las modificaciones hechas a un objeto se repliquen en los objetos relacionados**. De esta manera se mejora el mantenimiento de los objetos, se asegura que los cambios introducidos se replican de manera eficiente y se mantiene la **integridad de los datos**.

12.2 Modelo de aplicaciones modernas

Osakidetza se encuentra en un proceso de renovación de la arquitectura tecnológica, con el objeto de adaptarla a las prácticas más actualizadas.

La arquitectura de aplicación tradicional de tres niveles (que comprende un navegador, una aplicación monolítica implementada en un servidor de aplicaciones y una base de datos) es inadecuada para respaldar nuevas demandas comerciales digitales. Las aplicaciones monolíticas inflexibles, la complejidad arquitectónica y la deuda técnica

sobrecargan a los equipos de desarrollo e impiden la agilidad llegando a frustrar a los usuarios. La modernización de la cartera de aplicaciones facilitará la transformación del negocio digital, este esfuerzo de modernización comienza con una arquitectura de aplicación moderna.

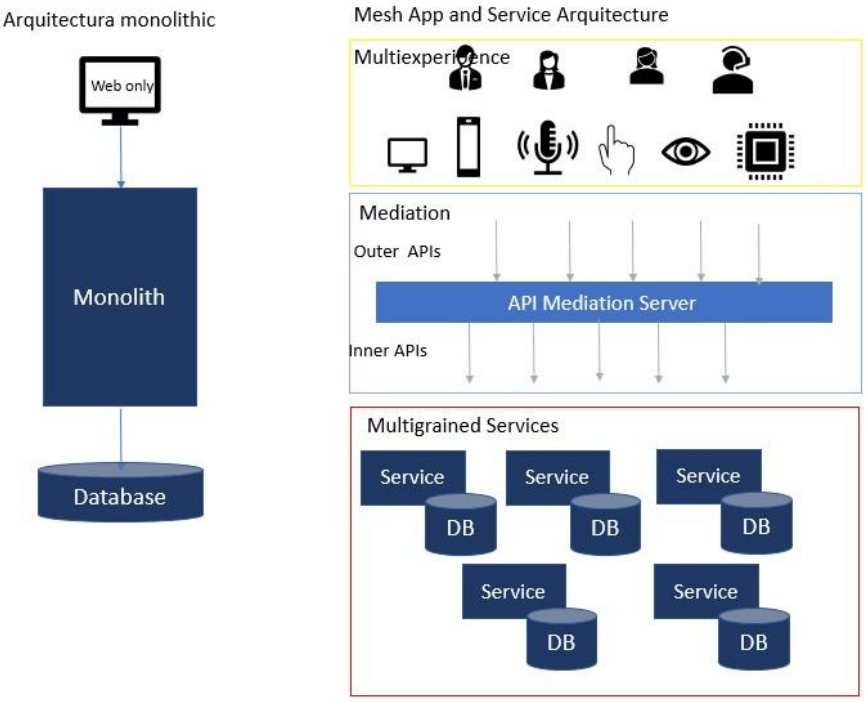
Las demandas de “negocios digitales” conducen hacia una rápida transformación de aplicaciones, sistemas y procesos, así como la necesidad de aplicaciones innovadoras de múltiples experiencias y obligan a brindar soluciones flexibles que puedan adaptarse rápidamente a los cambios en los requisitos y los avances tecnológicos.
Una arquitectura ágil basada en tecnologías modernas de interfaz de usuario (UI), API y servicios poco acoplados brindan la agilidad y flexibilidad para satisfacer estos requisitos. Este enfoque de arquitectura es lo que el mercado llama MASA (Mesh App and Service Arquitecture), arquitectura de aplicaciones y servicios en malla.
MASA (Mesh App and Service Arquitecture) proporciona tres capacidades arquitectónicas para potenciar el negocio digital:

- multiexperiencia (múltiples experiencias de usuario), este enfoque brinda la oportunidad de crear aplicaciones específicas más pequeñas que son más fáciles de diseñar, desarrollar e implementar. También permite que la aplicación se concentre en cualquier canal que prefiera el usuario al realizar tareas, ya sea una aplicación basada en la web que se ejecuta en una computadora portátil, una aplicación móvil en un teléfono o una interfaz conversacional.
- mediación de API (manejo de las API programáticas), un principio crítico en MASA es garantizar que las partes individuales de la aplicación puedan cambiar según sea necesario sin afectar otras partes de la aplicación. Debido a que las API pueden causar dependencias estrechas entre los componentes de la aplicación, todas las interacciones de la API deben abstraerse para garantizar un acoplamiento flexible.

  Monitorización de interacciones: la mediación de API puede monitorear interacciones desde perspectivas técnicas y comerciales.
  Protección de las interacciones: la mediación de la API puede hacer cumplir las políticas de autenticación y autorización y evitar las intrusiones de malware.
  Gestión del flujo de tráfico: la mediación de API puede admitir el enrutamiento de solicitudes, el equilibrio de carga, la limitación, la redirección y la recuperación automática de interacciones fallidas.
  Gestión de transformaciones: la mediación de API puede traducir entre diferentes protocolos: Puede transformar mensajes, por lo que admite formatos personalizados para diferentes consumidores y asignaciones entre diferentes versiones de API.

- servicios multigrano (microservicios, mini servicios o web services), cada servicio encapsula una capacidad comercial y expone esa capacidad a los consumidores a través de una API. El tipo de API más común utilizado en MASA es una API REST, pero MASA también admite otros tipos de API de solicitud/respuesta (como SOAP y gRPC), así como API basadas en mensajes y eventos.

  Macroservicios granulares que encapsulan y exponen algunas o todas las capacidades de una aplicación monolítica sin refactorización.
  Miniservicios de implementación independiente y granulares que implementan la funcionalidad a nivel de dominio.
  Microservicios detallados y de implementación independiente que implementan características individuales.



Aplicaciones multiexperiencia

La estrategia de aplicaciones front-end en MASA aplica un enfoque modular y granular para crear aplicaciones de clientes especializadas. Este es un cambio que se aleja del enfoque tradicional donde la intención es poner tanta funcionalidad como sea posible en las aplicaciones de clientes individuales. En lugar de una sola aplicación de cliente compleja, la experiencia del usuario de una aplicación de malla se compone de aplicaciones enfocadas estrechamente alineadas con flujos de trabajo específicos requeridos por diferentes personas que trabajan a través de varias interfaces. Esto brinda la oportunidad de crear aplicaciones más pequeñas y con un enfoque limitado que pueden usar múltiples servicios desde el back-end. Esto también le permite enfocar las aplicaciones en los canales preferidos del usuario para realizar sus tareas, ya sea una aplicación web en una computadora portátil, una aplicación móvil en un teléfono o una aplicación en una interfaz conversacional.

En una aplicación grande y monolítica, los usuarios deben navegar para encontrar la parte de la funcionalidad que necesitan para realizar las tareas específicas para las que están usando la aplicación. Sin embargo, en una aplicación multiexperiencia adecuada, la aplicación debe ser intuitiva, hacer que las tareas del usuario sean lo más eficientes posible y admitir el canal en el que el usuario desea utilizar la aplicación.

El diseño de la aplicación utilizando personas y escenarios implica un proceso de cuatro pasos:

1. Crear personajes de usuario para establecer una comprensión precisa de sus usuarios. Use las personas para identificar cosas como preferencias de dispositivos, nivel de experiencia, comportamientos y limitaciones de tiempo.
2. Identifique escenarios para comprender mejor las tareas que requieren los usuarios. Los escenarios deben revelar el orden de las tareas requeridas, los

pasos para realizar las tareas y las expectativas específicas que los usuarios pueden tener al realizar las tareas en una aplicación.

3. Diseñe flujos de trabajo basados en las personas y los escenarios. Estos flujos de trabajo deben identificar el flujo natural de trabajo, así como las posibles desviaciones que pueden experimentar los usuarios. Tenga en cuenta que la aplicación puede admitir varias personas y flujos de trabajo.

4. Utilice los flujos de trabajo para impulsar el diseño de la aplicación. Implementación de diseño de páginas web/móviles u otras interfaces, así como la navegación entre ellas, para soportar de manera óptima los flujos de trabajo definidos.

Además de diseñar las aplicaciones para que proporcionen una apariencia coherente en varias aplicaciones, la apariencia debe ser coherente en varios canales. Se utilizarán marcos, plataformas y herramientas comunes que garanticen la coherencia entre las aplicaciones y simplificar su implementación:

Frameworks de JavaScript como Angular, React, Vue.js y Polymer para aplicaciones web personalizadas

Frameworks móviles como Xamarin, React Native e Ionic para aplicaciones móviles personalizadas

Plataformas de experiencia digital (DXP)

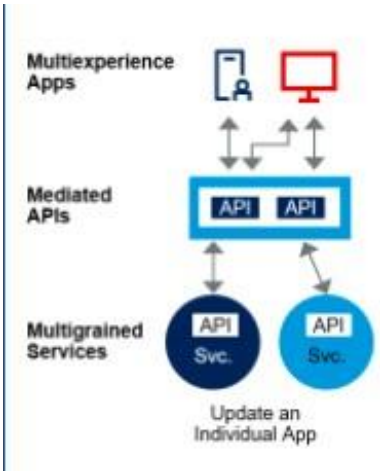
Plataformas de aplicaciones de código bajo (LCAP) y plataformas de desarrollo multiexperiencia (MXDP)

Mediación de servicios, API y eventos

Las API mediadas permiten ofrecer API optimizadas para satisfacer las necesidades de consumidores específicos. El enfoque de API mediado permitirá abstraer las complejidades en la implementación del servicio, así como para implementar un control de acceso, seguridad y administración consistentes.

Un enfoque de API mediada implica la implementación de tres capas distintas:

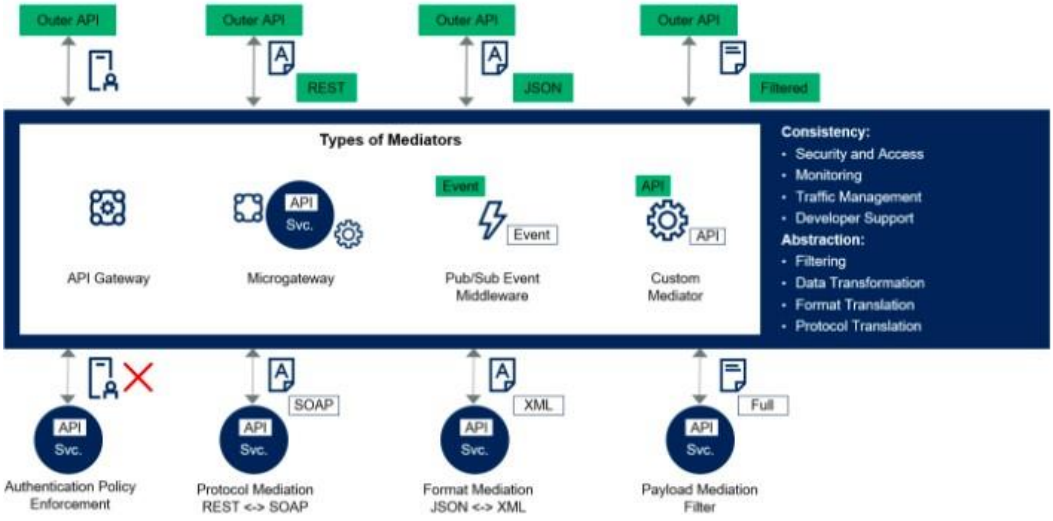
- API internas habilitan el dominio nativo del servicio y los modelos de interacción.
- API externas proporcionan una capa uniforme para que las aplicaciones accedan a la funcionalidad de back-end. Las aplicaciones pueden compartir una API externa o tener una API única. Las aplicaciones usan API externas como parte de su plataforma y, por lo tanto, las tratan como productos de consumo individuales.
- La mediación de API asigna las API externas a las API internas, monitorea y registra las interacciones y aplica políticas de tiempo de ejecución para la seguridad y la gestión del tráfico. Esto incluye actuar como intermediario para traducir protocolos, formatos, estructuras de datos y otras transformaciones a medida que las solicitudes pasan entre las API internas y las API externas.



Implementar una capa de mediación para administrar las API

La capa de mediación permite abstraer las complejidades de los servicios de las preocupaciones del cliente de la API externa. La capa de mediación puede usar uno o más mediadores, tecnologías que se ubican entre las API externas que usan las aplicaciones cliente y los servicios. Los contratos técnicos definido en la API externa y el diseño de la API interna ayudan al diseño y la implementación de la capa de mediación. Mapee los requisitos técnicos de la API externa con las capacidades actualmente expuestas por el servicio. Por ejemplo, la API externa puede requerir una interfaz REST mientras que el servicio expone una interfaz SOAP.

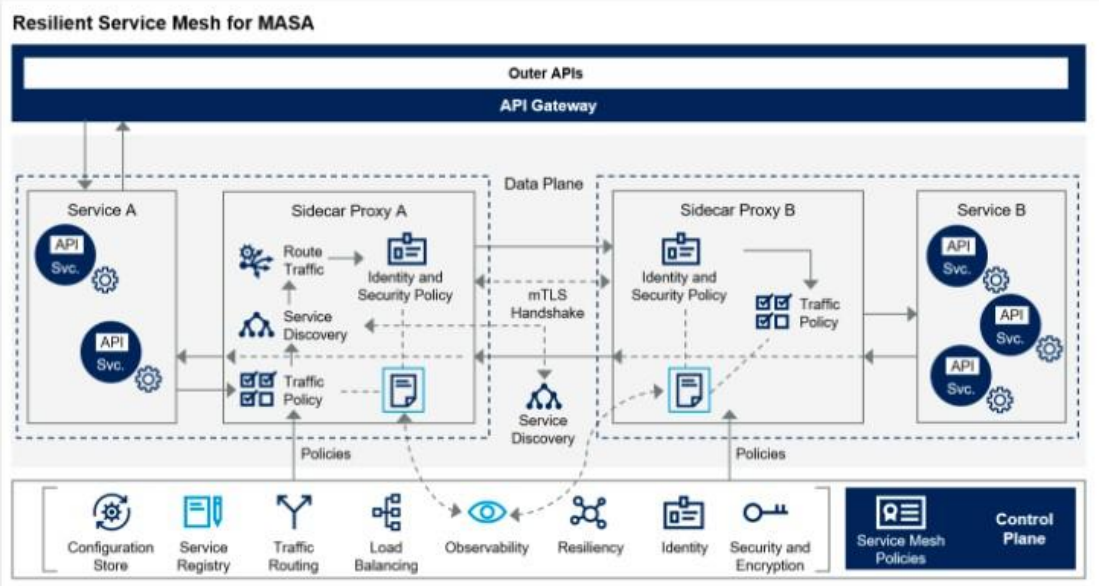
Una puerta de enlace API (API gateway) es el lugar óptimo para comenzar con la mediación porque proporciona de manera efectiva una capa consistente para aplicar políticas de administración de seguridad, identidad, monitoreo y tráfico. Además, algunas puertas de enlace API pueden usar políticas para filtrar o transformar datos.



Implementar capacidades de Service Mesh para admitir servicios multigrano

Construir un MASA con servicios multigrano requiere capacidades para habilitar y gobernar las interacciones entre servicios. Una red de servicios (service mesh) es una combinación de componentes de infraestructura, arquitectura y configuración, como

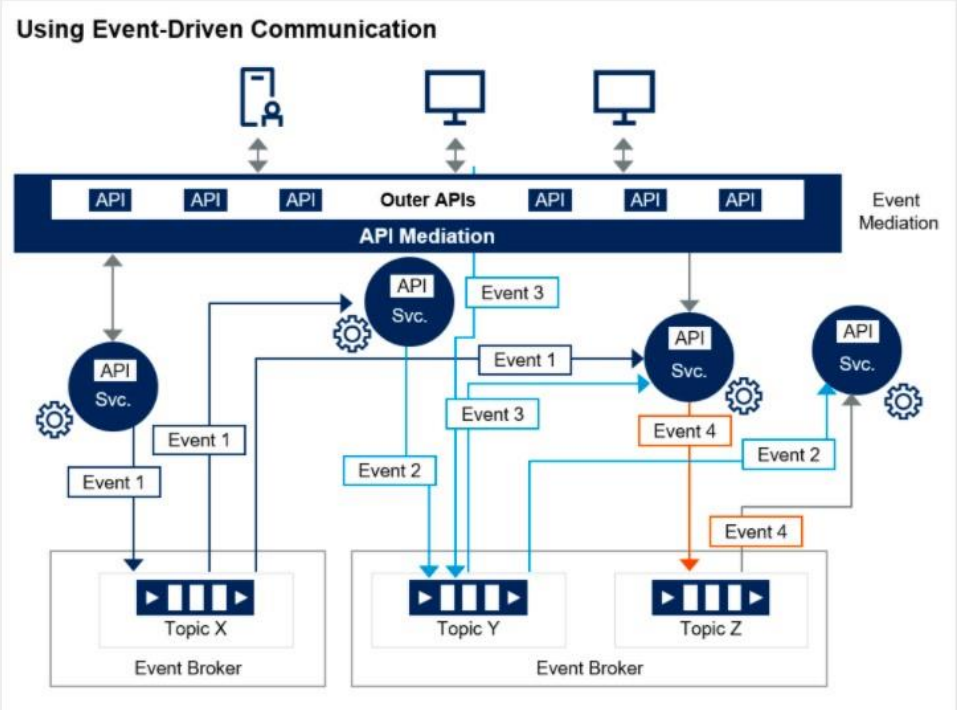
puertas de enlace internas o proxies y mediación ligera que proporciona un modelo de comunicación coherente entre servicios débilmente acoplados. Una red de servicios (service mesh) es parte de la capa de mediación de API y es un componente importante de MASA que permite un modelo consistente para definir, implementar y administrar las complejidades de la comunicación entre múltiples servicios desacoplados.



Modelo basado en eventos para la comunicación desacoplada

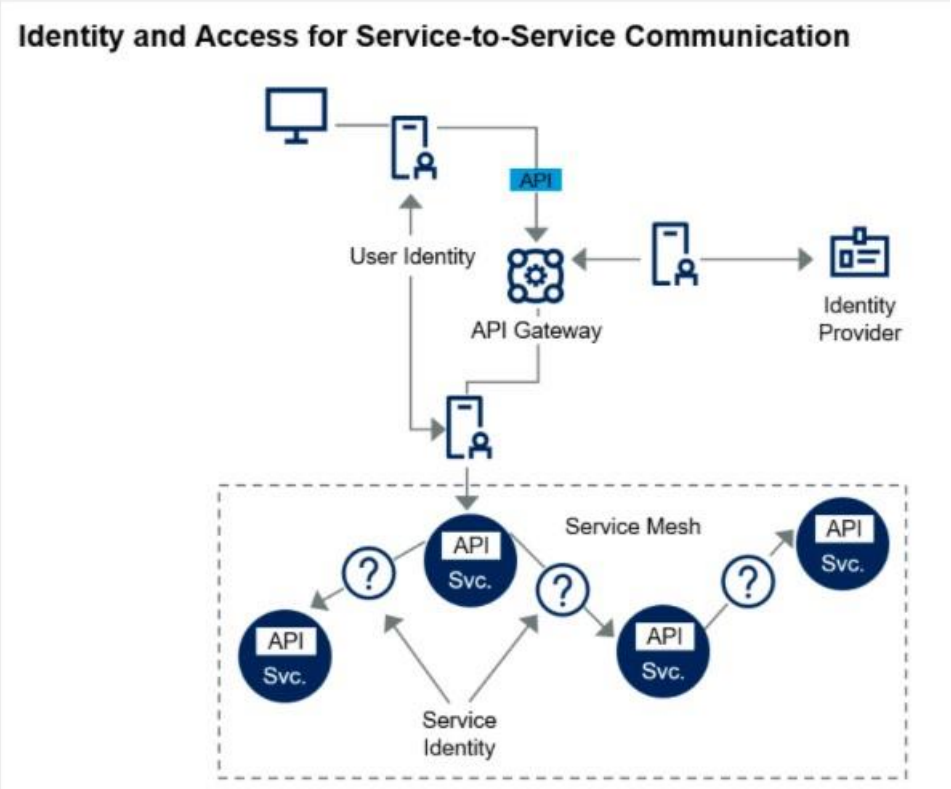
Un modelo de comunicación basado en eventos proporciona un método altamente desacoplado y eficiente para comunicar cambios de estado y otros eventos entre servicios. En un modelo de comunicación basado en eventos, los editores de eventos envían eventos a un tema y los consumidores de eventos se suscriben a esos temas y ejecutan la funcionalidad cuando reciben un evento.

Un modelo de comunicación basado en eventos proporciona un mayor grado de desacoplamiento que un modelo de comunicación tradicional de solicitud/respuesta. Los productores de eventos no tienen conocimiento directo de los consumidores intermedios y no dependen del procesamiento de eventos: simplemente notifican al corredor que se ha producido un evento. Los consumidores de eventos tampoco tienen dependencias directas de los productores de eventos: simplemente ejecutan la funcionalidad cuando reciben un evento.



Aplicar una estrategia de identidad y acceso para API y servicios multigrano

Aprovechar la mediación para externalizar la autenticación y la autorización de los componentes de la aplicación. Una aplicación de malla que utiliza la comunicación de servicio a servicio agrega una dimensión adicional a la identidad y el acceso. La protección solo en la puerta de enlace es un antipatrón. Los servicios necesitan protegerse a sí mismos; no asuma que una red es segura.



La estrategia de identidad y acceso admite la autenticación para las solicitudes de servicio a servicio, además de las solicitudes de las aplicaciones. Usar un servicio de puerta de enlace para convertir los tokens de autenticación de los usuarios (como contraseñas) en tokens de seguridad y páselos entre los servicios. Utilizar la mediación en la red de servicios para administrar la identidad y el acceso de la solicitud en la comunicación de servicio a servicio.

Implementar servicios multigrano

Los servicios son un componente clave para implementar una aplicación utilizando MASA. Los servicios proporcionan la lógica comercial, el acceso a datos y recursos, y otras funciones de back-end requeridas por las aplicaciones. Los servicios exponen esta funcionalidad a través de las API internas que reflejan su modelo de dominio interno y su modelo de interacción nativo, es decir, sus API internas.

El propósito de las API internas es exponer la funcionalidad de los servicios para que la use la capa de mediación. Las API internas desacoplan y abstraen la funcionalidad del servicio para proporcionar flexibilidad, agilidad y escalabilidad en la implementación y despliegue de los servicios. Las API internas deben centrarse en el dominio del servicio para el consumo general de otros servicios y mediadores. Esto es diferente de las API externas, que se enfocan en las necesidades particulares de las aplicaciones de los clientes.

La funcionalidad de aplicaciones empresariales, las aplicaciones heredadas, SaaS y los servicios de integración habilitados a través de API son ejemplos de implementaciones de servicios potencialmente disponibles para las aplicaciones MASA.

Los patrones comunes de integración de servicios personalizados incluyen:

- Orquestación y agregación: las aplicaciones desacopladas en MASA a menudo requieren interacciones con múltiples sistemas de back-end. Los servicios de orquestación personalizados coordinan las solicitudes a múltiples sistemas de back-end para proporcionar un acceso

unificado a los datos y el procesamiento de esas aplicaciones. Esto reduce la cantidad de solicitudes que las aplicaciones deben enviar al back-end. Un uso común de la orquestación son los servicios de agregación, que combinan datos de varios sistemas back-end y pueden aplicar proyecciones, filtros y otras operaciones para optimizar los datos.

- Coreografía: el patrón de coreografía y orquestación le permite realizar una única solicitud a un servicio personalizado que inicia el procesamiento en varios servicios de back-end en un orden específico o a un conjunto específico de sistemas de back-end.

- Backend para frontend: las aplicaciones multiexperiencia a veces requieren una API de back-end especializada que se aplique específicamente a sus necesidades. En lugar de diseñar e implementar un back-end único y completo, un servicio personalizado que abstrae las necesidades específicas de la aplicación de la implementación del backend le permitirá utilizar las capacidades del back-end existentes.

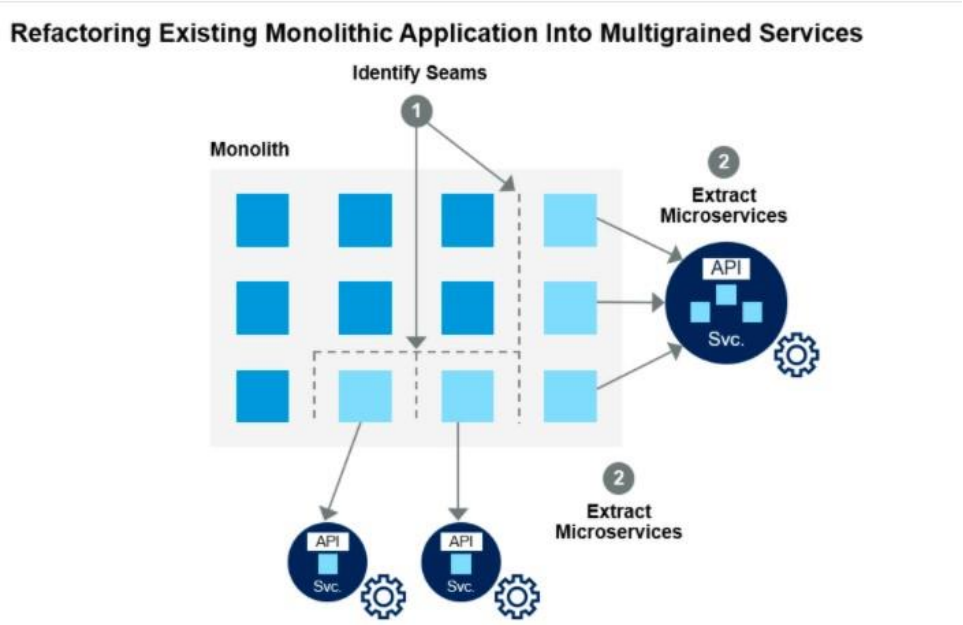
- Procesamiento por lotes: los servicios de integración personalizados proporcionan un método adaptable para implementar el procesamiento por lotes de datos de una variedad de fuentes de datos en un sistema de back-end.

- Integración de servicio a servicio: los servicios de integración personalizados le permiten conectar servicios de back-end de varias maneras, como sincronizar cambios de estado entre dos sistemas de back-end diferentes o ampliar el procesamiento comercial al iniciar procesos comerciales adicionales en otros sistemas.

- Integración basada en eventos: los eventos proporcionan un mecanismo efectivo para integrar los cambios de estado que ocurren en un servicio para procesarlos en un servicio separado con un acoplamiento mínimo.

Refactorizar los sistemas existentes:

Refactorice los sistemas existentes que limitan el rendimiento, la escalabilidad o la cadencia de entrega requerida por la aplicación MASA. Esto evita la fragilidad posterior, las limitaciones de flexibilidad y los problemas de escalabilidad. No necesita refactorizar toda la aplicación, solo las partes que necesita para la aplicación de malla. La refactorización debe ser un proceso iterativo en el que refactorice el monolito extrayendo piezas.



Implemente nuevos servicios con la granularidad adecuada:

Implemente nuevos servicios para agregar funcionalidades que no existen y que no se pueden refactorizar. Los nuevos servicios también incluyen funciones que pueden orquestar o agregar funciones de otros servicios existentes para admitir nuevas funciones requeridas por la aplicación.
Considere la granularidad de la funcionalidad del servicio MASA como un espectro con macroservicios en un extremo, microservicios en el otro y miniservicios en el medio:

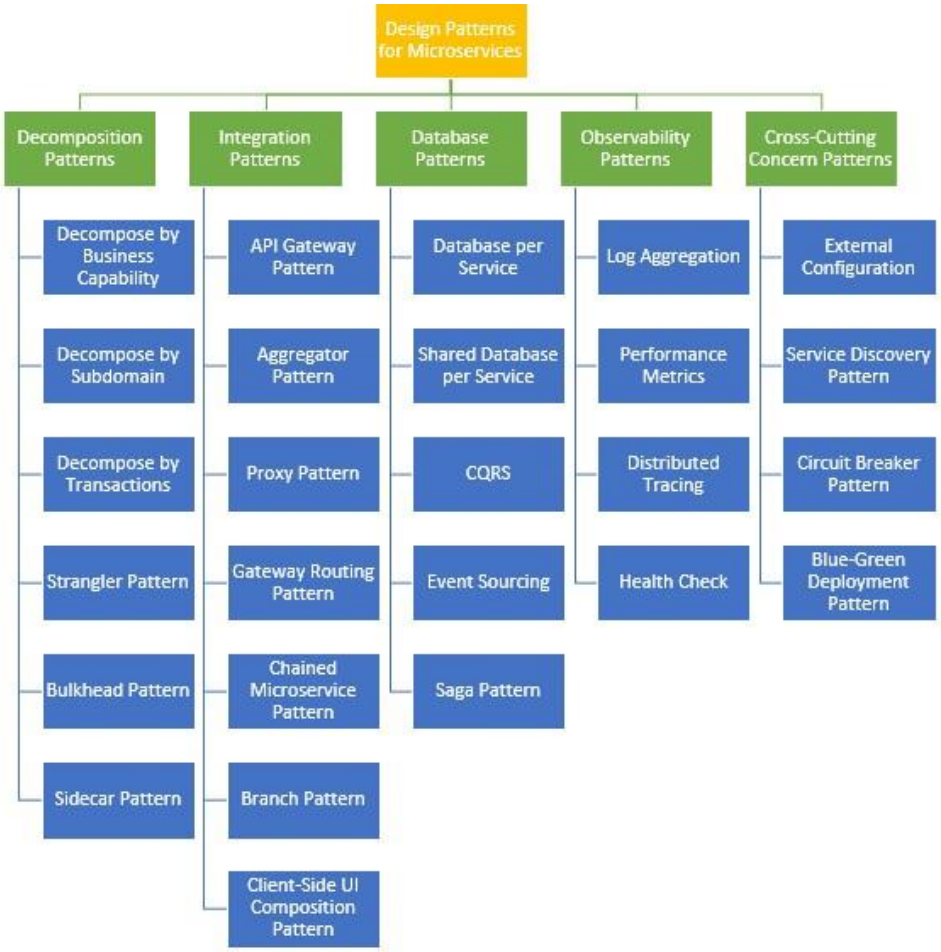
- Macroservicios: servicios detallados que habilitan algunas o todas las funciones de una aplicación monolítica; los cambios requieren reconstruir y volver a implementar todo el monolito.
- Miniservicios: servicios de granularidad gruesa que brindan funcionalidad a nivel de dominio, lo que le permite realizar e implementar cambios en un solo servicio de dominio en la aplicación de malla sin afectar necesariamente a los otros servicios de dominio.
- Microservicios: servicios detallados con capacidades específicas que se implementan de forma independiente y brindan la mayor cantidad de flexibilidad.

Con esta renovación se persiguen los siguientes objetivos:

- Contar con una base tecnológica plenamente actualizada
- Profundizar en la modularización del sistema, donde cada módulo mantenga una total independencia, tanto en su versionado, como en cuanto al equipo responsable de desarrollo y mantenimiento
- Mejorar el Time to Market de los nuevos desarrollos
- Posibilitar una correcta gobernanza del sistema en Producción, optimizando la relación infraestructuras / rendimiento
- Maximizar la reutilización de código
- Automatizar al máximo las tareas de integración, pruebas, despliegue y control de calidad.

12.2.1 Especificaciones de desarrollo

Microservicios se refiere a servicios independientes e implementables adjuntos a un dominio particular. Los microservicios se comunican a través de redes. La arquitectura de microservicio consta de múltiples microservicios que colaboran y es una forma de arquitectura orientada a servicios (SOA).
https://learncsdesign.medium.com/microservices-design-patterns-91fe56a33a47



Patrones de descomposición:

- Descomponer por capacidad empresarial

Los microservicios tienen que ver con hacer que los servicios estén poco acoplados, aplicando el principio de responsabilidad única. Se descompone por capacidad empresarial. Definir los servicios correspondientes a las capacidades del negocio. Una capacidad empresarial es un concepto del modelado de arquitectura empresarial, es algo que hace una empresa para generar valor.

- Descomponer por subdominio

Descomponer una aplicación usando capacidades comerciales puede ser un buen comienzo, pero hay algunas clases que no serán fáciles de descomponer. Estas clases serán comunes entre múltiples servicios. Defina los servicios correspondientes a los subdominios de diseño controlado por dominio (DDD). DDD dominio consta de varios subdominios. Cada subdominio corresponde a una parte diferente del negocio.

- Patrón de descomposición por transacciones

Puede descomponer los servicios sobre las transacciones. Entonces habrá múltiples transacciones en el sistema.

- Patrón de estrangulador

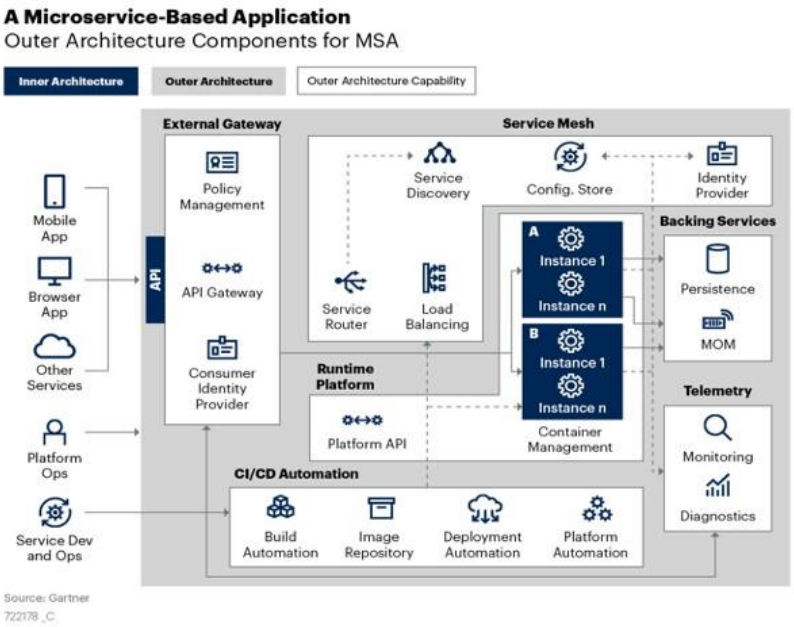
Los patrones de diseño por los que pasa descomponen aplicaciones para Greenfield, pero el 80 % del trabajo que realiza es con aplicaciones brownfield, que son aplicaciones grandes y monolíticas (base de código heredada). El patrón Strangler viene al rescate o solución. Esto crea dos aplicaciones separadas que viven una al lado de la otra en el mismo espacio URI. Con el tiempo, la aplicación recién refactorizada "estrangula" o reemplaza la aplicación original hasta que finalmente puede cerrar la aplicación monolítica.

- La malla de servicio (Service Mesh) patrón informático distribuido, implementado como middleware, que optimiza las comunicaciones entre servicios de aplicaciones.

Proporciona mediación de proxy y/o liviana para comunicaciones de servicio a servicio y admite funciones como autenticación, autorización, encriptación, descubrimiento de servicios, enrutamiento de solicitudes, equilibrio de carga e instrumentación de servicios.

  - ➢ Sidecar proxies

- API gateways
- OAuth 2.0
- Microservices orchestration



12.2.2 Entorno tecnológico

Existen 2 estrategias a nivel de utilización de imágenes base:

12.2.2.1 Contenedores

Existen 2 estrategias a nivel de utilización de imágenes base que se explicarán en siguientes puntos:

- Modo no gestionado (CaaS)
- Modo gestionado (PaaS)

En paralelo, hay 3 estrategias de persistencia en contenedores:

- **PVC RWO**: Volúmenes accesibles como lectura-escritura por un solo contenedor.
- **PVC RWX**: Volúmenes accesibles como lectura-escritura por múltiples contenedores (requiere validación de rendimiento).
- **S3**: Almacenamiento basado en objetos fuera de kubernetes y accesible en modo lectura-escritura por múltiples contenedores.

Desde la arquitectura de aplicación y sus necesidades de persistencia hay que tener en cuenta los siguientes aspectos:

- PVC RWO dificulta la resiliencia multi-réplica de contenedores en kubernetes.
- PVC RWX penaliza el rendimiento en accesos con volumen elevado de archivos grandes.
- El uso de S3 requiere el alta del servicio con el código de las APPs.

12.2.2.1.1 Modo no gestionado (CaaS)

En este modo todo el peso del contenedor recae en el desarrollo. Se proveen flujos de

CI/CD para el despliegue de la imagen, pero ésta no es gestionada desde Osakidetza. A pesar de eso puede ser exigible el cumplimiento de ciertos requisitos de seguridad durante el despliegue.

12.2.2.1.2 Modo gestionado (PaaS)

En este modo desde Osakidetza se proveen imágenes base de plataformas, asegurando el ciclo de vida de estas, y los parcheos, tanto evolutivos, como de seguridad.

Desde la aplicación a su vez existen 2 opciones para adaptarse al ciclo de vida de estas imágenes base.

- Utilizar la version "**latest"** de la misma, de forma que en cada despliegue siempre se utilice la última versión publicada de la imagen base
- Utilizar una versión concreta de la misma, de forma que el control de la versión a utilizar en cada despliegue dependa de desarrollo.

A nivel tecnológico existen imágenes base de las siguientes plataformas:

- Tomcat
- Nginx (basada en la imagen oficial)
- NodeJS (basada en la imagen oficial)
- Spring boot
- MongoDB
- Redis

12.2.2.2 Orquestación de contenedores

La aplicación funcionará sobre una plataforma para aplicaciones modernas (*cloudnative application platform*) basada en contenedores orquestados por Kubernetes.

Esta plataforma se apoya en el portfolio de soluciones de VMware Tanzu y aporta:

- El entorno de ejecución nativo e integrado en el cloud privado sobre el que está desplegada el resto de la infraestructura de Osakidetza.
- Capacidades de gestión, operación y observabilidad para los contenedores.
- Prestaciones de seguridad de la información, pudiéndose establecer además políticas transversales con las partes basadas en máquinas virtuales.
- Gestión del ciclo de vida del software.
- Soluciones de continuidad como backup y recuperación y posibilidad de despliegue en CPDs de respaldo.
- Capacidad para que la aplicación pueda funcionar en diferentes proveedores de cloud pública del mercado sin precisar de adaptaciones e hibridada también con el entorno on-premise estando federados entre sí.

En cada una de esas ubicaciones se podrá disponer a su vez de entornos de no -

producción y producción según se precise por criterios de calidad del software, testing, conveniencia o agilidad.

La plataforma está operada en dos niveles:

1. **Operación automática** kubernetes (autoremediaciones).
2. **Operación manual** según responsabilidades por el equipo que da servicio a Osakidetza de Operación , Plataforma (PlatOps), Infraestructura (CloudOps) y Seguridad (SecOps).

El equipo de desarrollo (DevOps) consume la plataforma **definiendo como código** sondas de operación automática, despliegues por entorno y arquitecturas modernas.

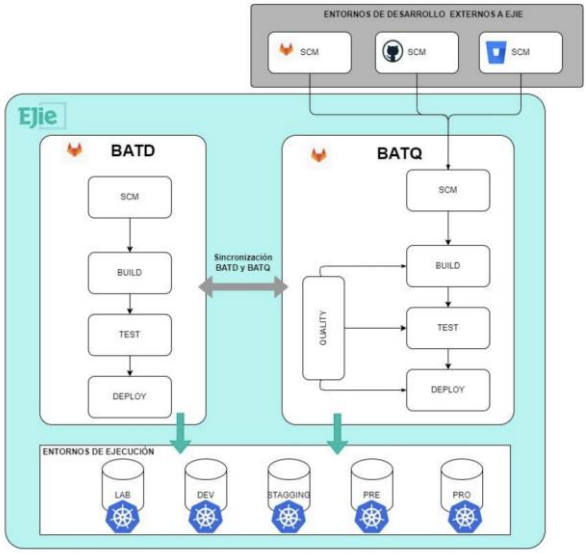
En un ciclo de **mejora continua y colaborativa devsecops**, la orquestación va adquiriendo madurez en agilidad, seguridad y resiliencia.

12.2.2.3 Ciclo de vida

Apoyándose en componentes de la plataforma descrita en el apartado anterior, se plantean como un modelo de ciclo de vida para aplicaciones modernas orientado mejorar la eficiencia, la seguridad y la competitividad buscando ofrecer una experiencia de usuario mejorada, con tiempos de respuesta más rápidos y menos interrupciones. Para la consecución de estos objetivos se plantean dos modelos de ciclo de vida:

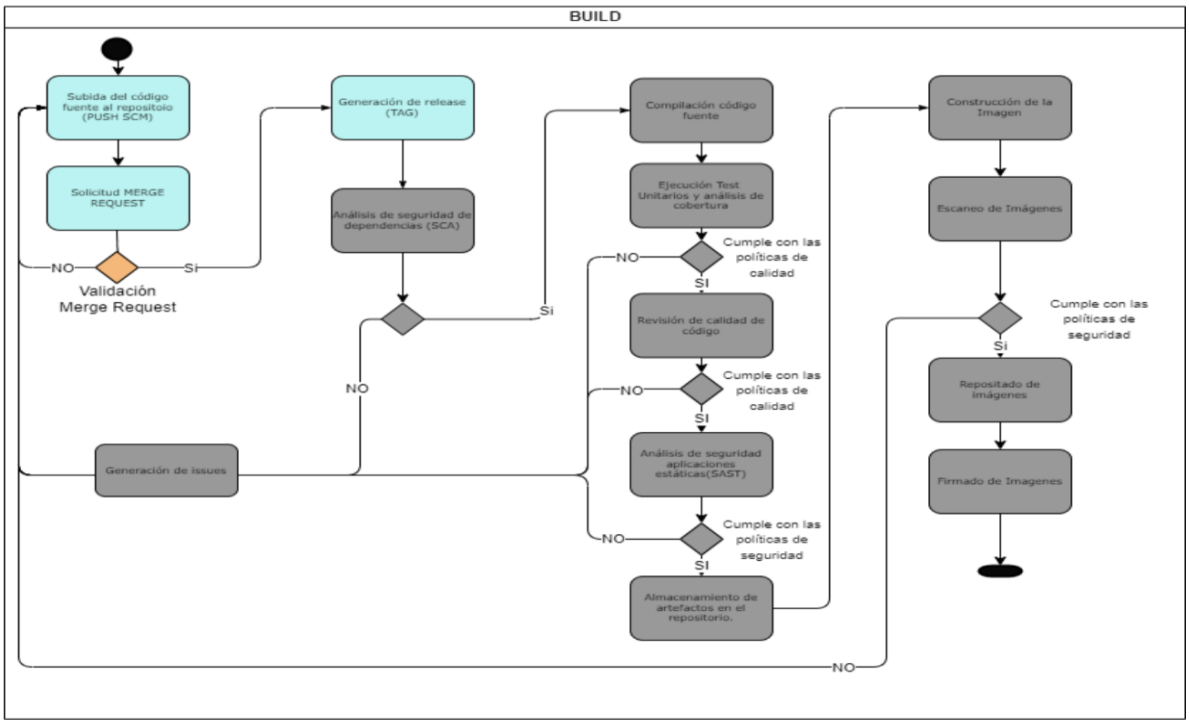
- Básico: Modelo de ciclo de vida cuyo objetivo es disponer de entornos orientados principalmente a los equipos de desarrollo:
  - Basado en un ciclo de vida orientados al CI/CD.
  - Maximizando la automatización de los procesos de integración y despliegue continuos.
  - Incorporando calidad mínimos de caja no personalizables.

- Calidad y Seguridad: Modelo de ciclo de vida orientado a asegurar la calidad (calidad y seguridad) en aplicaciones modernas garantizando el cumplimiento de estándares de calidad y seguridad desde su concepción hasta su despliegue y mantenimiento continuo.
  - Basado en un ciclo de vida orientados al CI/CD.
  - Maximizando la automatización de los procesos de integración continua, pruebas y despliegue continuo.
  - Orientado a la prevención de vulnerabilidades y al maximizar la calidad desde las primeras etapas del desarrollo.
  - Este modelo se basa en una única suite de herramientas (Gitlab) que proporciona el ciclo de vida completo de las aplicaciones modernas



El proceso de BUILD para ambos modelos parte de las siguientes premisas:

- Establecer un modelo de trabajo basado en la premisa “Build Once, Deploy Anywhere” que permite desacoplar el proceso de construcción del proceso de despliegue.
- Incorporar un modelo de construcción orientado a asegurar que los desarrollos se llevan a cabo en base a versiones de frameworks y runtimessoportados(Tanzu Buildservices).
- Incorporación de seguridad y calidad en base a análisis de calidad de código, test unitarios, análisis SAST, SCA (dependencias de terceros), escaneo de imágenes base. (En base a herramientas opensource en el primer modelo en base a herramientas soportadas en el segundo modelo)
- Incorporación de políticas de calidad y seguridad base en función de la clasificación de las aplicaciones: de caja en el primero y personalizables en segundo.

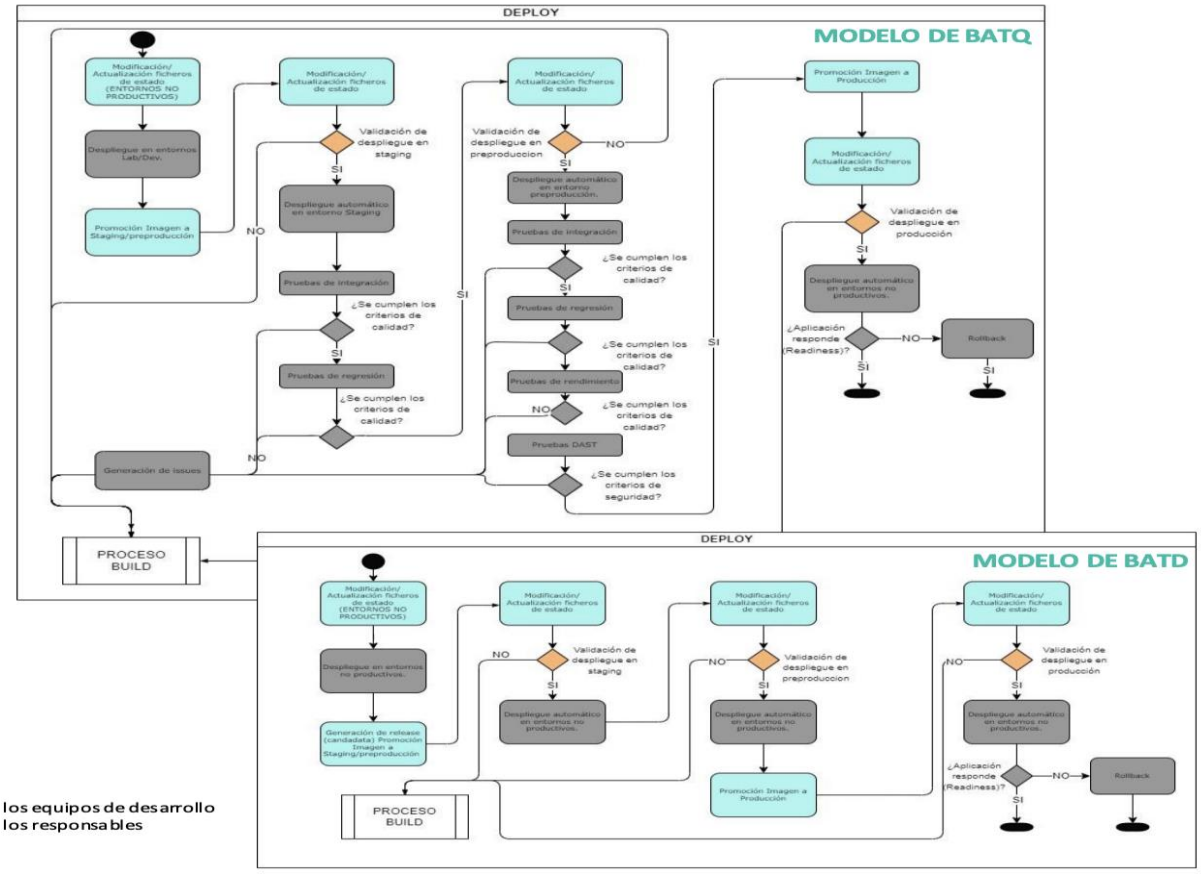


El proceso de DEPLOY parte de las siguientes premisas:

- Incorporación del paradigma GitOps cuyo objetivo es gestionar la configuración de las aplicaciones en kubernetes utilizando Git como única fuente de verdad manteniendo un espejo entre el estado de las aplicaciones en el repositorio y el estado de las aplicaciones en los clusters de kubernetes permitiendo aplicar el principio de “You own it, you ship it!".
- Incorporación de aprobación de despliegues en los diferentes entornos basados en la aceptación (merge request) de las modificaciones de los ficheros de estado de los componentes.
- Incorporación de rollbacks automatizados basados en la respuesta a los readiness probes de las iniciativas. En caso de no cumplirse la política establecida, se llevará a cabo el rollback mediante la reversión de los cambios de los ficheros de estado.

El segundo modelo se centra en el deploy en asegurar la calidad y seguridad de las aplicaciones proporcionando capacidades adicionales con respecto al primero:

- Automatización de la ejecución de los procesos de pruebas integradas, regresión/funcionales, rendimiento y DAST en función de los entornos sobre los que se despliegan.
- Incorporación de políticas de calidad y seguridad personalizables en base a la clasificación de las aplicaciones.



los equipos de desarrollo
los responsables

12.2.2.4 Plataformas horizontales

Plataformas distribuidas de transmisión de datos, como Kafka, almacenamiento de objetos, etc.

En múltiples ocasiones existen escenarios mixtos en los que la aplicación se evoluciona a un modelo sobre contenedores, pero el dato sigue persistiendo en BBDD's "tradicionales".

En estos casos las versiones estandarizadas son:

- Oracle RAC 19c (para réplicas se usa Active Data Guard)*
- Oracle 19c Single instance
- Microsoft SQL Server 2019 y 2022(*)
- MongoDB 7.0
- PostreSQL 15.6
- MySQL 8.0.30

(*) Continuidad de negocio y Alta disponibilidad: Oracle RAC, SQL Server Always On sólo para aplicaciones críticas

12.2.2.5 Cloud

En general Osakidetza se acogerá al Servicio Cloud dentro del Catálogo de Servicios

Convergentes, que tiene un enfoque híbrido y multicloud para dotarse de capacidades. Existen a su vez tres modelos de entrega IaaS, PaaS y SaaS.

El suministro de estos servicios Cloud se gestiona a través de un marco común de contratación de proveedores que es gestionado desde EJIE. EJIE garantizará el cumplimiento de principios y requisitos técnicos:

- En relación con Gestión de Identidad de Accesos y autentificación:
  - Las personas usuarias de la herramienta existirán en el directorio corporativo del Gobierno Vasco y la herramienta se conectará con este directorio para autenticar a la persona usuaria.
  - La integración con el directorio corporativo se hará utilizando mecanismos estándar: SAML / OAUTH.
  - El sistema se integrará con el proveedor de identidad que determine la Dirección competente en materia TIC.
  - Los permisos / roles que una persona usuaria tenga en la herramienta se configurarán en la propia herramienta, aunque se podrá analizar la viabilidad de gestionar estas autorizaciones en sistemas de EJIE
- Cumplir con la certificación ENS Esquena Nacional de Seguridad con Nivel de Seguridad Medio.
- Cumplir con normativa (RGPD) Reglamento General de Protección de Datos y mecanismos de confianza de UE.
- Disponer de como mínimo dos zonas de disponibilidad diferenciadas, presencia en el territorio de la Unión Europea y/o por normativa legal presencia en el territorio del Estado Español dependiendo del servicio. Se tenderá de forma preferente a desplegar en esta última región.
- Disponer de medios que garanticen el tratamiento de los datos para avalar que no son manipulados ni usados por algoritmos, por terceros o proveedores.
- Poseer las certificaciones requeridas de cumplimiento en ámbito de seguridad de la información, protección de identificación personal en Cloud públicas, seguridad para los servicios Cloud.
- Garantizar el acceso y control de sus datos a los usuarios.
- Garantizar la interoperabilidad a nivel de API para los servicios que lo requieran.

12.2.2.6 Modelos de servicio/entrega

Existen dos tipos de modelos de servicio/entrega:

- **Cloud público** que son los servicios prestados por los **CSPs** (Hiperescalares) homologados en Batera.
  - Modalidad SaaS
  - Modalidad PaaS (plataformas del portfolio del hiperescalar) o IaaS
- **Cloud privado** son los servicios prestados por **Cloud Privado Batera**.
  - Modalidad PaaS(hosting) o IaaS

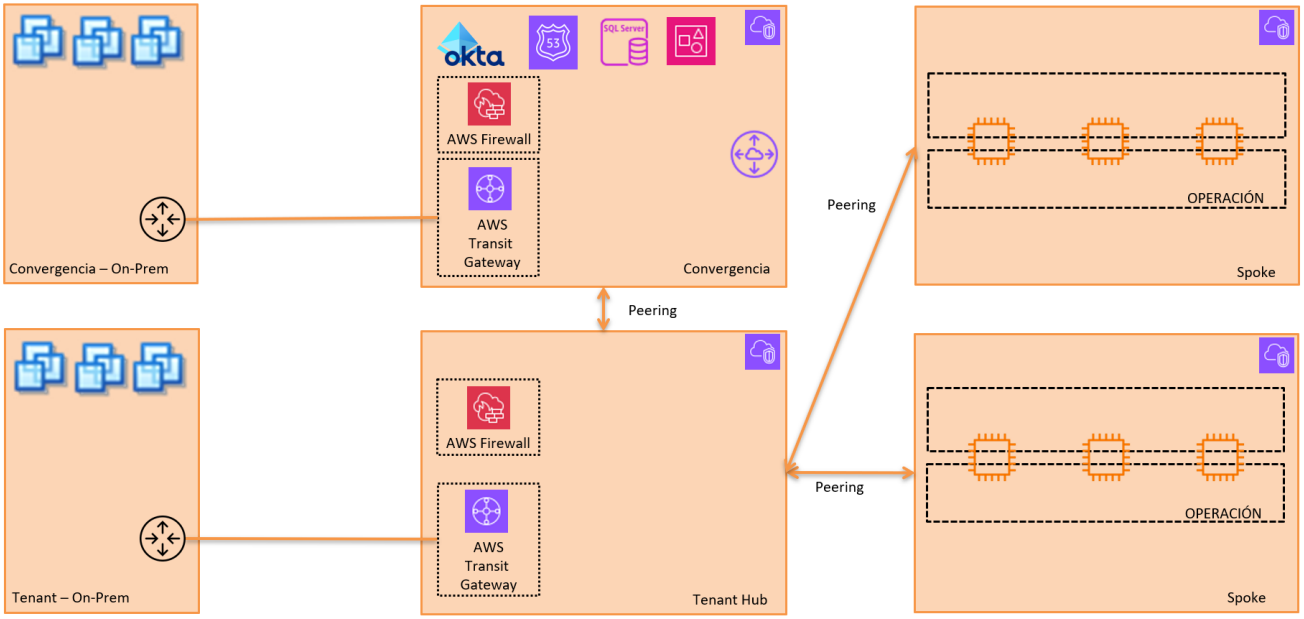
Los recursos desplegados para Osakidetza están encajados dentro de la Landing Zone de Batera en cada Hiperescalar, en la que se organizan los recursos según una estructura jerárquica de cuentas/suscripciones que permite aplicar políticas de seguridad y/o configuración, autenticar usuarios y vincular con roles según el contexto y las responsabilidades. Además, se prestan de forma transversal en cada CSP a Osakidetza los servicios de bajo nivel de registro de eventos, monitorización, auditoría

y networking tanto interno a una cuenta/suscripción, entre cuentas, hacia los CPDs on-premises o hacia el exterior/Internet según corresponda.

12.2.2.7 Modelos de conectividad

Para ser más ilustrativo se muestra un ejemplo con AWS, pero los conceptos son trasladables a otros hiperescalares homologados.

Se sigue en general una arquitectura "hub&spoke". Todas las conexiones hacia Internet o el entorno on-premises de Osakidetza se centralizan en cuentas de networking específicas, no pudiéndose realizar desde cada cuenta.



Estas conexiones al exterior siempre se realizan a través de una VPN entre los dos extremos, en el caso de que el extremo del CSP no soporte este mecanismo (un SaaS albergado en el CSP por ejemplo, que no lo contempla) se hará una conexión privada entre la cuenta implicada y la cuenta networking, para establecer desde ahí la VPN. Para el caso en el que se precise una conexión con QoS, tras un análisis es posible habilitar una conexión directa por MPLS.

12.3 Directrices tecnológicas de IOC

Osakidetza dispone de múltiples aplicaciones desarrolladas en diversas tecnologías (principalmente Java y .Net), y albergadas en diferentes plataformas (principalmente Apache HTTP Server, Microsoft IIS y VMware Horizon como soluciones de presentación, Oracle WebLogic, Apache Tomcat y Microsoft IIS como servidores de aplicación y Oracle RDBMS, Microsoft SQL Server, MongoDB como servidores de BBDD).

El modelo de referencia de las aplicaciones es una solución Web (sin plug-in's), con **arquitectura** de n-capas preferentemente **3 capas**: presentación, negocio y persistencia, compatible con tecnología de virtualización y almacenamiento SAN y NAS.

Dicha arquitectura es **escalable horizontalmente** en cuanto a las capas de negocio y persistencia (reparte la carga equitativamente entre los diversos servidores que ejecutan la lógica de negocio y estos a su vez, mediante un pool de conexiones acceden a la capa de persistencia), y es **tolerante a fallos** garantizando la continuidad de negocio.

Los servidores que atienden a las capas de negocio y persistencia están ubicados en los dos CPDs de la Dirección General de Osakidetza siendo el reparto de los mismos proporcional en un CPD u otro; la capa de presentación está distribuida en los puestos de trabajo fijos, portátiles, móviles y tabletas en los diversos centros de Osakidetza (Hospitales, Centros de salud, etc..).

De modo que la solución (producto/aplicación/desarrollo) a suministrar deberá ser compatible con el modelo de arquitectura indicado en el párrafo anterior y con la relación de productos y tecnologías que se indican a continuación (las que apliquen).

12.3.1 Entorno tecnológico

En conjunto, el entorno tecnológico de las soluciones actualmente en producción es:

- Plataformas de desarrollo:
  - .NET sobre Windows 2019 y 2022(*)
  - Java sobre Red Hat Enterprise Linux 8 y Red Hat Enterprise Linux 9(*)
  - Aplicaciones móviles:
    - Node.js, Express, AngularJS, MongoDB sobre Red Hat Enterprise Linux 8
    - Ionic, Cordova sobre Android, iOS

(*)En la actualidad Windows 2022 Y RHEL 9 en construcción

- Infraestructura:
  - On-premises:
    - Cloud Privado:
      - VMware Cloud Foundation 9
      - VMWare vSphere 8.0 U3
      - vSAN
      - NSX Networking 4.x
      - Aria Suite Enterprise 8
      - Avi Load Balancer
      - Almacenamiento fichero y objeto:
        - PowerScale (CIFS y NFS)
        - Scality (S3)
  - Cloud Pública:
    - CSPs, homologados en Batera.

- Sistemas Operativos:
  - Servidores:
    - Windows 2019/2022
    - Red Hat Enterprise Linux 8 y Red Hat Enterprise Linux 9
  - Puesto de trabajo fijo: Windows 10, Enterprise x64

- Puesto de trabajo móvil: Android, iOS

- Servidores Web:
  - Apache Webserver 2.4.37 sobre RHEL 8
  - MS IIS 10 (S.O. Windows Server 2019 y Windows Server 2022)
  - SAP Web Dispatcher 7.73 (actual 7.20 obsoleta)
  - Nginx 1.18.0

- Servidores de Aplicación:
  - Apache Tomcat 9.0.64 + JDK 8, 10.1.13 + OpenJDK 11.0.18 o 17.0.7
  - Oracle Weblogic Server 12.1.3 + JDK 7, 12.2.1.4 +JDK8 y 14c + JDK 11
  - MS IIS 10 + .NET Framework 4.7.2 + Windows Server 2019
  - MS IIS 10 + .NET Framework 4.8 + Windows Server 2022
  - Application Server de SAP Netweaver 7.5
  - Node.js 12.17.0

- Sistemas de Gestión de Bases de Datos (*)
  - Oracle RAC 19c (para réplicas se usa Active Data Guard)
  - Oracle 19c Single instance
  - Microsoft SQL Server 2019 y 2022 en construcción
  - MongoDB 7.0
  - PostreSQL 15.6
  - MySQL 8.0.30

  (*) Continuidad de negocio y Alta disponibilidad: Oracle RAC, SQL Server Always ON sólo para aplicaciones críticas.

- Gestión de contenidos:
  - MS SharePoint Online (O365)
  - Documentum 21.1

- Infraestructura SOA:
  - Oracle Service Bus 12c (12.2.1.4) + JDK 8.0

- Plataforma de firma (PKI):
  - Izenpe

- Autenticación:
  - Directorio Activo de MS (AD)

- Modo de ejecución/escritorio:
  - local: puesto fijo o móvil
  - Cliente de Horizon 2212.1

- Dispositivos Inalámbricos:
  - Seguridad:
    - Utilización del estándar IEEE 802.11 b/g/n
    - Capacidad de asociarse a un SSID de la red WIFI correspondiente
    - Soporte SSID oculto
    - Soporte WPA2 Enterprise
    - Cifrado de datos AES
    - Autenticación EAP basada en certificado de máquina
    - Soportar certificados:
      - Los certificados emitidos son de 1024 bits
      - El certificado de la CA raíz es de 4096
      - Soporten certificados con hash SHA-2
      - El tipo de certificado: x.509
      - Protocolo: RSA
  - Señal:
    - Ganancia de la antena: Mínimo 3 dBi
    - Sensibilidad <= -70 dBm a 11Mbps

Consideraciones en relación con los productos y tecnologías indicados anteriormente:

- En cuanto al navegador, se debe **evitar** el uso de Applets (Java), componentes ActiveX (.NET), Flash u otras tecnologías que impliquen la **descarga y ejecución de software embebido en el navegador**.
- Al respecto de **Office 2016**, no está permitido el uso, dependencia o interrelación con Access.
- Al respecto de los dispositivos inalámbricos se han especificado los **requisitos mínimos** que deben cumplir.

12.3.2 Especificaciones del cliente (maqueta)

Actualmente, la versión de maqueta que se instala en los equipos de **Osakidetza** es la *5.0(W10 22H2),* aunque convive con versiones anteriores instaladas.

- Hardware (mínimo existente):
  - PC arquitectura Intel
  - Procesador INTEL i5-10500T a 2,3GHz hasta 3,8GHz, 6 núcleos 12MB yPro.
  - Memoria RAM 16 GB DDR4
  - Disco duro 256 GB SSD
  - Monitor 24 pulgadas con resolución FHD 1920 x 1080

- Restricciones de escritura en disco C: para usuarios standard, excepto en su perfil de usuario (C:\users\<usuario>)

o En cuanto al navegador, se debe evitar el uso de Applets (Java), componentes ActiveX (.NET), Flash u otras tecnologías que impliquen la descarga y ejecución de software embebido en el navegador.

o Al respecto de Office 2016, no está permitido el uso, dependencia o interrelación con Access.

Sistema Operativo que instala esta imagen		
X	Actual - Windows 10, Enterprise x64 22H2	
Idiomas Disponibles:	Castellano (es-ES); Euskera (eu-ES)	
Hotfixes:		
Descripción		Clasificación
Última versión de Windows 10 soportada por Microsoft, donde se incluyen los **parches y actualizaciones de seguridad hasta julio de 2024** (Windows y Office). **Internet Explorer ya no se encuentra en la maqueta**.		Updates Críticas y de Seguridad

- SOFTWARE BASE EN MAQUETA

Producto	Fabricante	Versión	Idioma
7-Zip (x64)	Igor Pavlov	22.01.00.0	ESP
Adobe Acrobat DC (64-bit)	Adobe	2.200.320.282	MUI
Adobe Shockwave Player	Adobe Inc.	12.3.5.20282	ESP
Crystal rteports Basic Runtime for Visual Studio 2008	Business Objects	10.5.0.0	ESP
Epson Printer Software	_____	_____	_____
Epson Universal Print Drivers Printer Uninstall	Seiko Epson Corporation	2.69.0	Multi
Google Chrome	Google LLC	126.0.6478.127	ESP
Idazki Desktop	Izenpe S.A.	3.5.1	ESP
Instalable DNIe	Cuerpo Nacional de Policia	14.1.0	ESP
Instalable TC-FNMT	FNMT-RCM	7.2.0	ESP
Izenpe Middleware	Izenpe S.A.	4.0.2.528	ESP;EU
Java 8 Update 151	Oracle Corporation	8,0,1510,12 (x86)	ESP
Java 8 Update 151 (64 bit)	Oracle Corporation	8,0,1510,12 (x64)	ESP
Libre Office Suite	The Document Foundation	7.5.1.2	Multi
LibUsDotNet	Travis Robinson	2.2.8	Multi
LAPS (Admin Local Password Solution)	Microsoft	6.2.0.0	ESP
Microsoft Access Runtime 2016	Microsoft	16.0.4288.1001	ESP;EU
Microsoft Edge	Microsoft	126.0.2592.1001	Multi

Microsoft Office Language Interface Pack 2016 - Euskara	Microsoft	16.0.4266.1001 (x64)	EUSK
Microsoft Office 2016 Standard	Microsoft	16.0.4266.1001 (x64)	ESP;EU
Microsoft OneDrive	Microsoft	19.043.03	ESP
Microsoft Visual C++ 2005	Microsoft	8.0.56336	ESP
Microsoft Visual C++ 2010 x64	Microsoft	10.0.40219	ESP
Microsoft Visual C++ 2012 x64	Microsoft	11.0.61030.0	ESP
Microsoft Visual C++ 2012 x86	Microsoft	11.0.61030.0	ESP
Microsoft Visual C++ 2013 x64	Microsoft	12.0.40664.0	ESP
Microsoft Visual C++ 2013 x86	Microsoft	12.0.40664.0	ESP
Microsoft Visual C++ 2015-2022 x86	Microsoft	14.32.31332.0	ESP
Microsoft Visual C++ 2015-2022 x64	Microsoft	14.32.31332.0	ESP
Microsoft Windows Desktop Runtime - 6.0.12 (x64)	Microsoft	6.0.12.31928	ENG
Mozilla Firefox (x64-es)	Mozilla	128.0	Multi
Mozilla Maintenance Service	Mozilla	128.0	Multi
NDDigital DCS Server	NDDigital	2.0.0.6	ESP
NDDigital nddPrint Agent	NDDigital	5.23.22	Multi
Paint	Microsoft		
Paquete Controladores de Windows-Lexmark International Printer	Lexmark International	10/01/2015 2,10,0,0	Multi
Paquete Controladores de Windows-Lexmark International Printer	Lexmark International	10/02/2015 2,10,0,0	Multi
Paquete Controladores de Windows-Lexmark International Printer	Lexmark International	10/03/2015 2,10,0,0	Multi
SAP Crystal Reports runtime engine for .NET Framework 32 bit	SAP	13.0.22.2668	Multi
SIAVAL SafeCert CSP Client	SIA	3.2.0	ESP
Team Developer 5,2 Deployment	Unify	5.2.0000	Multi
Vmware Horizon Client	Vmware	8.8.1.34412	Multi
WebView2 Runtime de Microsoft Edge	Microsoft	126.0.2592.87	
Windows Driver Package-Direccion General de la Policía (UMPass) SmartCard	Direccion General de la Policía	1.0.2.9	Multi
Windows Driver Package-Fabrica Nacional de Moneda y Timbre (UMPass) SmartCard	Fabrica Nacional de Moneda y Timbre	1.0.0.11	Multi
Xuxen 5,1 Euskarako Zuzentzailea	Elhuyar	5.4.10	ESP;EU
Zoom	Zoom	5.17.34827	
Zoom Outlook Plugin	Zoom	6.0.0	

- SOFTWARE DE POST-INSTALACIÓN

Bitdefender Endpoint Security Tools	Bitdefender	7.9.8.350	ENG

CrowdStrike Windows Sensor	CrowdStrike INC.	7.5.17706.0	ENG
FlexxAgent	Flexxible	24.5.1.2	ENG
Configuration Manager Client	Microsoft	5.00.9128.1007	ENG

12.3.3 Dispositivos de entrada/salida

La solución deberá ser capaz de interactuar con diversos dispositivos de entrada y salida de diversos fabricantes; siendo dichos dispositivos: pantallas táctiles, impresoras térmicas, monitores, por tanto, la interfaz con los mismos será en base a estándares del mercado.

Así mismo la solución deberá adaptarse a la resolución de las diversas pantallas.

En relación con dispositivos móviles se deben tener en cuenta ciertas particularidades en relación con las características de movilidad, tamaño, comunicación inalámbrica e interacción con las personas:

- Son aparatos pequeños.
- La mayoría de estos aparatos se pueden transportar en el bolsillo del propietario o en un pequeño bolso.
- Tienen capacidad de procesamiento.
- Tienen conexión permanente o intermitente a una red.
- Tienen memoria (RAM, tarjetas MicroSD, flash, etc.).
- Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, de modo que puede adaptarlos a su gusto.
- Tienen una alta capacidad de interacción mediante la pantalla o el teclado.

En lo que se refiere a dispositivos sobre los que tienen que ejecutar soluciones de movilidad hay dos ámbitos que son las soluciones orientadas a profesionales de Osakidetza y soluciones orientadas al ciudadano.

Las aplicaciones orientadas al ciudadano se ejecutarán en dispositivos de consumo ya sean móviles, tabletas o cualquier otro dispositivo al que vaya destinada la solución. En este caso el desarrollo deberá ejecutarse en dispositivos con los diferentes sistemas operativos que haya en el mercado.

En las aplicaciones orientadas a los profesionales de Osakidetza las aplicaciones deberán de ejecutarse en el dispositivo elegido y homologado para soportar la solución. Habrá que tener en cuenta los siguientes aspectos:

- Ergonomía, tamaño y usabilidad del dispositivo, se refiere a que cumpla las expectativas del usuario en ámbito en el que se va a desarrollar el proyecto.
- Especificaciones de limpieza y resistencia a los golpes.
- Características de comunicaciones como pueden ser conectividad Wifi y 4G.

12.3.4 Seguridad, acceso, autenticación y autorización

El tratamiento de los datos cumplirá lo establecido por la legislación vigente en materia de **seguridad** y protección de datos.

En el ámbito intranet se permitirá el acceso mediante los **transportes** RMI, JMS, HTTP y HTTPS.

El proveedor de **servicios de certificación** de Osakidetza es Izenpe.

Osakidetza ha dotado a todos sus profesionales de certificados corporativos reconocidos, en soporte tarjeta criptográfica, con capacidad de firma de documentos y mails, cifrado y autenticación en el AD.

Además, para los nuevos SI que necesitan firma electrónica ágil, Osakidetza ha implantado una solución horizontal de firma, basada en certificados alojados en HSMs, llamada "firma centralizada". Son certificados con las mismas características y garantías que los alojados en tarjeta criptográfica. Estos HSM son de última generación y cumplen con las más exigentes normas internacionales (FIPS 140-2 Level 2 and Level 3, Common Criteria at EAL 4+ y RoHS compliant). Permiten la firma a través de CSP propio (applets) y la firma directa basada en una Pasarela de firma propietaria.

Izenpe es también el proveedor de la infraestructura de autenticación, validación y firma basada en certificados electrónicos. Osakidetza dispone de 2 Zain en HA (equipos TrustedX de Safelayer) en propiedad y su backup, en modo servicio, en los Zain de Izenpe. La integración con estos equipos se puede realizar mediante su librería Smartwrapper. Provee diferentes servicios web (validación, evolución de firmas, firma en servidor, etc.) alojados en los OSB de Osakidetza.

Así mismo, Osakidetza dispone de su propia infraestructura de Custodia de documentos firmados, para asegurar la validación de cualquier firma realizada en sus sistemas de información con certificados electrónicos. Está constituida por 2 equipos Siaval en HA, de la empresa SIA, y HSM Ncipher propios.

Para la firma electrónica de documentos in-situ por parte de sus pacientes, Osakidetza dispone de una solución denominada "firma biométrica", que asegura la validez de todas las firmas implicadas. Se apoya en la solución SmartAccess de Telefónica.

Osakidetza está inmerso en la adecuación de toda su infraestructura y sistemas de información al reglamento europeo eIDAS. Tanto en cuestión de certificados como de niveles de aseguramiento (autenticación).

Los sistemas de **autenticación** validos son:

- **Autenticación** basada en **Directorio Activo (Microsoft Active Directory) corporativo,** que permite identificar un empleado de Osakidetza mediante usuario/contraseña o la tarjeta profesional.
- **Token de Kerberos**, que habrá sido emitido por el Directorio Activo corporativo.
- **Autenticación** basada en **infraestructura de certificación digital (PKI)**. Esta autenticación permite la identificación de personas (certificados personales) o de sistemas de información (certificados de servidor)

- **Certificados X.509 de servidor**, que permitirán identificar sistemas (máquinas) y establecer confianza entre ellas.
- **Certificados X.509 de cliente**, que permitirán identificar individuos (personas) - Para los servicios web, el estándar **WS-Security** permitirá aplicar políticas de autenticación.
- El bus de servicios sólo se utilizará para la autenticación basada en **certificados X.509** de servidor.

El sistema de **autorización** se implementará por medio de la definición de roles de usuario en base a los cuales se gestionarán los permisos y acciones sobre los distintos procesos por los que estará compuesta la solución. De modo que cada usuario en función del rol que tenga asignado podrá acceder a una serie de funcionalidades, así como a los datos relativos a su organización de servicio, centro de trabajo, servicio/área de atención.

Se crearán grupos de autorización del Directorio Activo. Por cada perfil que tenga la aplicación (medico, enfermera, administrativo, administrador de la aplicación…) se creará un grupo de autorización en el DA. Se mapeará el perfil al grupo.

12.3.5 Monitorización

Osakidetza dispone de una **plataforma de monitorización** a través de la cual ya tiene establecida una línea de monitorización base de su infraestructura y SI.

En este sentido la solución deberá incluir un apartado en el que se describan los elementos a monitorizar para asegurar un correcto funcionamiento del sistema, servicio o producto suministrado de forma integral dentro de la mencionada plataforma de monitorización.

La monitorización se podrá llevar a cabo a través de alguna combinación de los siguientes productos o métodos que Osakidetza integra dentro de su plataforma de monitorización:

- SCOM mediante Management Pack para sistemas Microsoft.
- Nagios XI para sistemas Linux.
- Enterprise Manager Cloud Control para SGBD Oracle / EM 13 y servidor de aplicaciones Weblogic para rendimiento.
- Dynatrace versión actual 1.300.27(SaaS) con instalación de agente Dynatrace (1.219.153) para aplicaciones web/Horizon

Las herramientas de monitorización se integran con Evanios/ITOM (gestión de eventos) para las plataformas Batera.

Con el objetivo de monitorizar la solución que vaya a ser suministrada, deberán detallarse los objetos concretos a monitorizar para garantizar su correcto funcionamiento más allá de los aspectos generales de la infraestructura de sistemas que los alberguen; dicha relación de objetos podrá incluir, aunque no exclusivamente,

algunos como:

- Procesos en ejecución
- Puertos de comunicaciones a la escucha
- Eventos concretos en registros del sistema
- Unidades de disco o File Systems

Para cada uno de ellos deberán incluirse los umbrales de consumo que se consideren anormales, especificándose 2 valores a partir de los cuales establecer el correspondiente nivel de alerta preventiva o alerta crítica según proceda.

Adicionalmente podrán suministrarse scripts o procesos para llevar a cabo:

- Pruebas sintéticas de los sistemas cuyo resultado determine el correcto funcionamiento del sistema o el correspondiente nivel de alerta preventiva
- Operativas sobre el sistema que puedan ayudar a determinar o especificar las condiciones concretas de los errores o alertas producidos.
- Operativas sobre el sistema para intentar remediar automáticamente la condición de la alerta o error, como reinicio de procesos o servicios, etc.

Todos estos procesos deberán entregarse con su correspondiente documentación, y su operativa deberá resultar sencilla de modo que pueda ser realizada por un operador.

12.3.6 Backup/Restore

Osakidetza dispone de una infraestructura para la realización de copias de seguridad y restauración a través de red basada en CommVault.

De modo que la solución deberá especificar los procedimientos necesarios a llevar a cabo para la recuperación del servicio ante la pérdida de este, corrupción de sus datos o alguno de sus componentes. En este sentido se plantearán escenarios de recuperación independientes para cada uno de los posibles componentes impactados y el orden de ejecución en caso de que deban ejecutarse secuencialmente para la recuperación de varios componentes o del sistema completo en su conjunto.

A partir de los procedimientos anteriormente descritos y en cada uno de ellos, se inferirá **los elementos concretos** sobre los **que se requiere hacer Backup**, en qué **orden**, con qué **frecuencia** y **política de persistencia** deben llevarse a cabo **para poder satisfacer** los distintos **escenarios de recuperación** del servicio contemplados. Así mismo deberán indicarse las restricciones que pudieran existir para la realización del Backup: posible impacto en el servicio, restricciones horarias, etc.

Los distintos procedimientos de recuperación suministrados con la solución deberán ser validados por Osakidetza para garantizar su compatibilidad con la infraestructura de copia de seguridad y restauración existente.

12.4 Interoperabilidad

Los requisitos de integración de diversos sistemas de información de Osakidetza se realizarán de acuerdo con una arquitectura orientada a servicios (SOA) sobre la plataforma Oracle SOA Suite (Oracle BPEL Process Manager, Oracle Service Bus (OSB), Oracle Business Rules, ...).

Dicha arquitectura proporciona una forma bien definida de exposición e invocación de Servicios Web, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

Sobre la misma arquitectura SOA, Osakidetza implementa una solución de integración orientada a Eventos; es un diseño a medida que gestiona un conjunto de sistemas que publican eventos y un conjunto de aplicaciones que se suscriben a determinados eventos. Dicha solución se denomina Gestor de Eventos-Event Manager y es responsable de recibir los eventos publicados, ejecutar las validaciones adecuadas, enrrutar y almacenar los eventos para su envío a los subscriptores que estén asociados a cada uno de los elementos recibidos.

Relación de estándares de comunicación soportados:

- Protocolos a nivel de mensaje: ○ SOAP 1.1 y SOAP 1.2 ○ WSDL 1.1 y WSDL 1.2 Binding ○ SOAP con Attachments ○ JSON – REST
- Protocolos de seguridad a nivel de mensaje:
  - ○ WS-Security 1.0/1.1 ○ WS-SecurityPolicy ○ WS-Policy ○ WSPolicyAttachment
  - ○ WS-Security: Username Token Profile 1.0/1.1 ○ WS-Security: X.509 Token Profile 1.0/1.1 ○ WSSecurity: SAML Token Profile 1.0/1.1 ○ WS-Security: KerberosToken Profile 1.1 ○ WS-Reliable Messaging 1.0 ○ WS-Addressing ○ WS-I Basic Profile 1.1
  - ○ WSSecurity: JWT Token (aplicaciones móviles en internet) ▯ Protocolos a nivel de transporte: ○ HTTP 1.0, HTTP 1.1 ○ TLS v1.2 y TLS v1.3, SSL ○ Interoperabilidad con registros UDDI v3-compliant ○ Sistemas middleware basados en JMS/MQ
  - ○ HTTPS

12.4.1 Servicios Web

Los Servicios Web se implementarán de acuerdo con las especificaciones WSDL v1.1, SOAP v1.1, v1.2, JSON-REST (en el caso de aplicaciones móviles), UDDI v2.XX y XML v1.0, con el objetivo de incorporar las recomendaciones de la WS-I definidas en la especificación Basic Profile v1.0, v2.0 y de esta manera asegurar la interoperabilidad entre los sistemas.

El estándar de codificación que se utiliza en los mensajes XML es UTF-8.
La seguridad aplicada a los servicios web cubrirán los siguientes aspectos:

- Autenticación: Verificar que el cliente (usuario o aplicación) es quien dice ser. La identidad de un usuario se realizará en base a la

información presentada por el usuario (usuario/contraseña, certificado, token SAML)

- Autorización: Otorgar acceso a los servicios en base a la identidad del cliente o a los roles asignados.
- Confidencialidad, privacidad: Mantener la información secreta mediante el uso de algoritmos de encriptación estándar de elementos XML.
- Integridad, no repudio: Asegurar que un mensaje permanece inalterado durante la transmisión mediante la firma digital. La firma también valida la identidad del remitente y proporciona una marca de tiempo para garantizar que una transacción no puede ser repudiada más tarde ni por el remitente ni por el destinatario.

Osakidetza usa Oracle Web Service Manager (OWSM) para gestionar y aplicar políticas a los servicios web publicados en la plataforma SOA.

La política estándar que Osakidetza ha definido para los servicios proporciona:

- Autenticación mediante certificado x509
- Protección del mensaje mediante firma (sin encriptado)

Existen dos versiones de la política en OWSM, una para servicios y otra para clientes. Para garantizar la interoperabilidad, cada política tiene su versión compatible con tecnología .NET y Java.

- oracle_wss10_x509_token_with_message_sign_service_policy
- oracle_wss10_x509_token_with_message_sign_service_policy_net
- oracle_wss10_x509_token_with_message_sign_client_policy
- oracle_wss10_x509_token_with_message_sign_client_policy_net
- oracle_http_jwt_token_server_policy

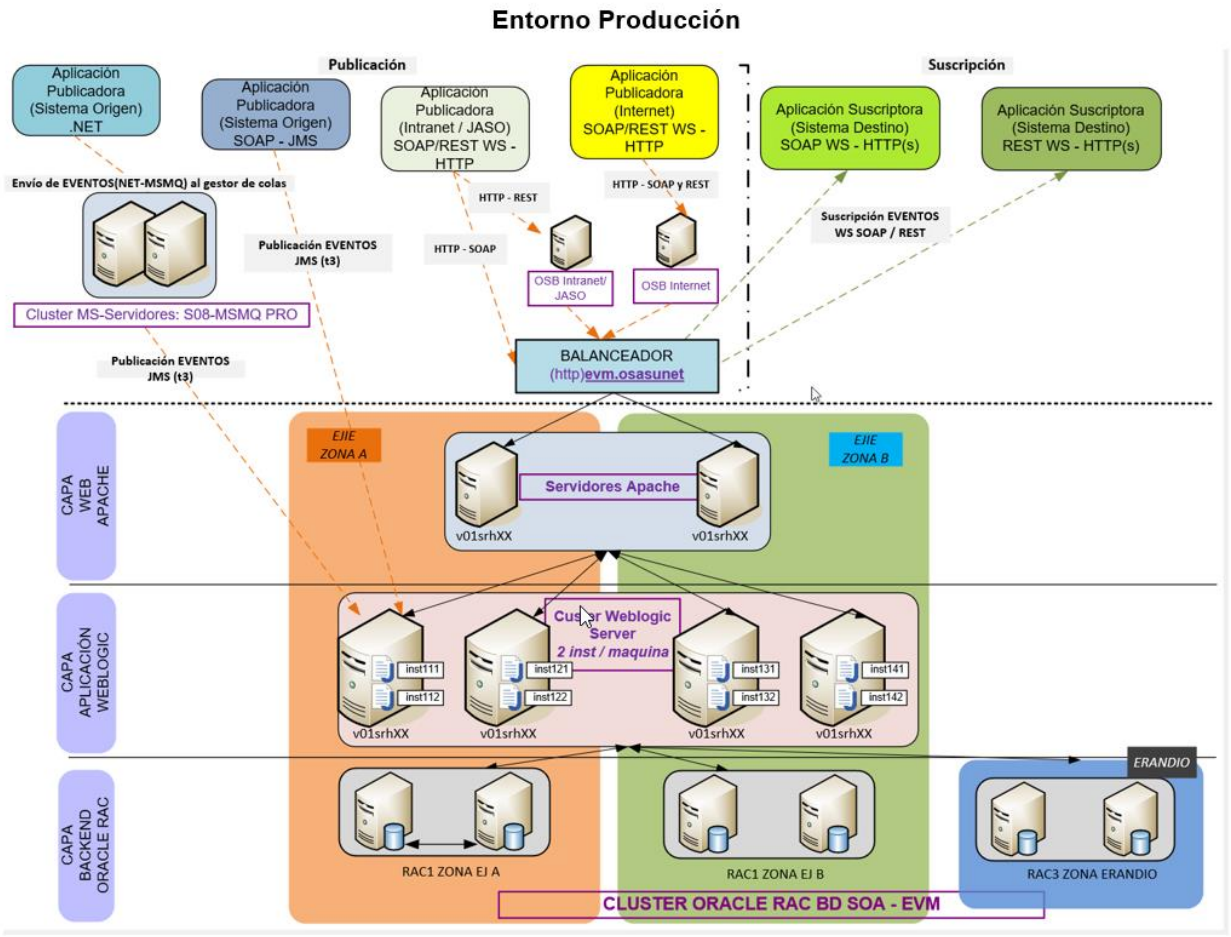
Los clientes y aplicaciones que acceden a través de internet entran a la DMZ a través del firewall de aplicaciones (WAF). El WAF aplica reglas contra ataques y define patrones de seguridad.

- Es responsabilidad de cada aplicación publicada en la DMZ controlar el acceso y autorizar a los usuarios.
- El OSB de internet publica los servicios a los que pueden acceder las aplicaciones de internet.
- El OSB de internet audita todas las llamadas a los Servicios Web mediante una política propietaria de Osakidetza gestionada por OWSM.
- En el OSB de internet se protegen todos los servicios con la política oracle_wss10_x509_token_with_message_sign_service_policy. Esta política autentica a las aplicaciones mediante certificado x509 y firma el mensaje de petición y respuesta.
- El OSB de internet delega la ejecución a servicios publicados en la Intranet.

- En la intranet, se despliegan instancias independientes de servicios web para dar servicio a las peticiones que llegan desde el OSB de Internet.

12.4.2 Gestor de Eventos

A continuación, se describen los requisitos técnicos que tienen que cumplir las aplicaciones para publicar y/o recibir eventos.



12.4.2.1 Estándares de comunicación

La mensajería del Servicio Osakidetza se implementará de acuerdo con las especificaciones del estándar HL7 a partir de la versión 2.5 o con cualquier otro formato propio de Osakidetza; de esta manera se asegura la interoperabilidad entre los sistemas de información.

Relación de tecnologías soportadas para la publicación y subscripción a eventos.

Tecnologías que soporta el gestor de eventos para la publicación y subscripción a eventos, así como si la modalidad soporta transaccionalidad y las opciones de seguridad disponibles:

Modalidad	Tecnología	Transaccional	Orden	Seguridad
Publicación	Mensajería JMS	Sí	Sí, si el publicador establece el parámetro UnitOfOrder	Autenticación (user/pass)
Publicación	Servicio web	No	No	Ninguna, Autenticación (user/pass) y WS-Security
Subscripción	Mensajería MS	Sí	Sí. Event Manager garantiza la entrega en el mismo orden que ha recibido los mensajes	Autenticación (user/pass)
Subscripción	Servicio OSB	Sí	Sí. Event Manager garantiza la entrega en el mismo orden que ha recibido los mensajes	Ninguna
Subscripción	Servicio web	Sí	Sí. Event Manager garantiza la entrega en el mismo orden que ha recibido los mensajes	Ninguna, Autenticación (user/pass) y WS-Security

Por cada modalidad y tecnología se deben cumplir unos requisitos técnicos que serán indicados al adjudicatario para la implementación de esta forma de integración.

12.4.2.2 Mensajería. Definición de Evento

Un evento es un documento XML definido mediante un XSD, donde:

- ✓ **id**: Es el identificador del tipo de evento. Se genera durante el proceso de alta del evento en el sistema de administración de Event Manager. Durante el procesamiento de un evento se verifica que el id sea válido.

- ✓ **correlation**: Es un campo libre en el que el publicador del evento indica un número correlativo relativo a su sistema.

- ✓ **source**: Es el identificador del publicador. Se genera durante el proceso de alta de un publicador en el sistema de administración de Event Manager. Durante el procesamiento de un evento se verifica que el source sea válido.

- ✓ **timestamp**: Lo establece el publicador del evento en el momento del envío.

- ✓ **metadata**: Puede contener un xml que ayude a describir el contenido del evento. Event Manager puede utilizar esta información para tomar decisiones de enrutado.

- ✓ **payload:** Es el contenido del evento. Puede ser cualquier cadena de texto o XML.

El resultado devuelto cuando se publica un evento en Event Manager es un XML definido por un XSD, donde:

- ✓ **uuid**: Es un identificador único que se asigna a cada evento procesado por Event Manager.

- ✓ **processed**: true o false, si el evento se ha procesado correctamente o con errores.

- ✓ **errorCode**: Si se ha producido un error, aqui se informa el código del error.

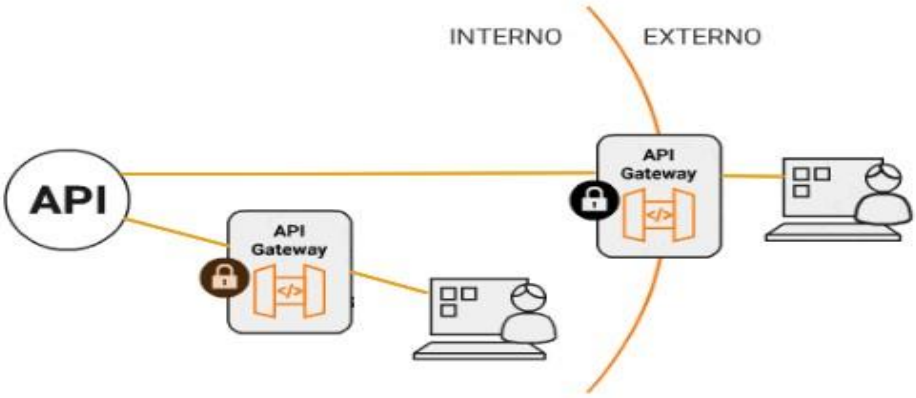
- ✓ **errorDescription**: Si se ha producido un error contiene la descripción de éste.

Se realiza gestión de errores: codificación de errores y descripción de estos.

12.4.3 Arquitectura API REST

De forma adicional a la Arquitectura SOA, Osakidetza permita la interoperabilidad mediante API REST.

El componente centrar de la arquitectura API REST de Osakidetza el API Manager.



Las funciones del API Manager son:

- Exponer las APIs de manera fácil y segura a consumidores internos y externos.
- Implementar flujos de autorización estándar, como OAuth, OpenID Connect y JWT.
- Gestionar el correcto diseño y el tiempo de ejecución para todo el ciclo de vida de las APIs
- Monitorización y estadísticas de uso.

Osakidetza requiere que las API se ajusten al documento de buenas prácticas de desarrollo distribuido para tal fin.

Las API deben tener documentación elaborada mediante Swagger/Open API 3.0.

12.4.3.1 Requisitos alta de API de forma automática

- Disponer de tarea jenkins para proceder con el alta automática mediante ficheros en Git.

- Disponer de un fichero de parámetros, en el mismo sitio que Api.yaml, el cual nos valga para definir los entornos y sus endpoints en cada caso. El archivo se llamará params.yaml, y tendrá una estructura predefinida.

- Tener creada la categoría que este definida en el api.yaml

12.4.3.2 Requisitos para traspaso entre entornos

La herramienta para poder lanzar comandos contra el actual apimanager de WSO2 es apictl.

La descarga se realiza desde: https://apim.docs.wso2.com/en/latest/install-andhttps://apim.docs.wso2.com/en/latest/install-and-setup/setup/api-controller/getting-started-with-wso2-api-controller/setup/setup/api-controller/getting-started-with-wso2-api-

controller/

12.4.3.3 Requisitos suscripción API

Para suscribirse a una API concreta, se requerirá la utilización y configuración de una clave APIKey. Esta clave la generará Osakidetza en el API Manager para todas las suscripciones de una misma aplicación.

12.5 Explotación Información- Business Intelligence

La herramienta usada para desarrollo de tecnología Business Intelligence es Oracle Analytics Server (OAS).

Esta herramienta es una herramienta web, que está soportada por una cada de presentación en Apache, y un servidor Web (Weblogic). En esta capa web de Weblogic se define la integración con el proveedor de identidades mediante ROLES de usuario (grupos y perfiles de usuario) que se usan para la autorización en el Acceso a Datos y a partes de la aplicación.

En este servidor también se alberga el RPD, que es donde se desarrolla todos los metadatos y las distintas Áreas Temáticas accesibles por usuarios.

Todos los datos accesibles desde OAS están almacenados en una única BBDD de Producción albergada en Exadata Cloud@Customer.

Para realizar la carga de datos (Incremental), utilizamos la herramienta de extracción (ETL – Extract, Transform and Load) en ODI (Oracle Data Integrator).

Para poder realizar las cargas incrementales necesitamos que los orígenes de datos tengan alguna fecha o característica que indique que el dato se ha modificado el día anterior, y así poder hacer cargas de sólo aquello que no existe en destino o se ha modificado. (Un LAST_MODIFIED por ejemplo). Las cargas son normalmente diarias, aunque existen cargas anuales o trimestrales.

Versiones utilizadas:

- Analítica Descriptiva – Business Intelligence (BI)
  - OAS 2024 (Oracle Analytics Server) 2024
  - Power BI
- ETL
  - ODI 12.2.1.4.0 (Oracle Data Integrator)

Además de lo expuesto en los párrafos anteriores, actualmente hay dos servicios de Analítica Avanzada:

- **Big Data as a Service (BdaaS)** – Configurable en base a las necesidades del proyecto, basada en Cloudera y On Premise. Servicio para extinguir, actualmente hay configuradas iniciativas que ya se consumen, aunque ya no se dan de alta nuevas en espera del nuevo servicio de plataforma analítica en cloud.

- **Nueva plataforma analítica en cloud** - Planteamiento de evolución al cloud como premisa para toda la estrategia del dato. Actualmente servido sobre AWS (Amazon Web Services). Este servicio está en fase de diseño y construcción hoy en día (2024).

12.6 Soluciones Móviles

Este apartado refleja las tecnologías para el desarrollo de aplicaciones móviles, definida por Osakidetza.

Comprende dos capas:

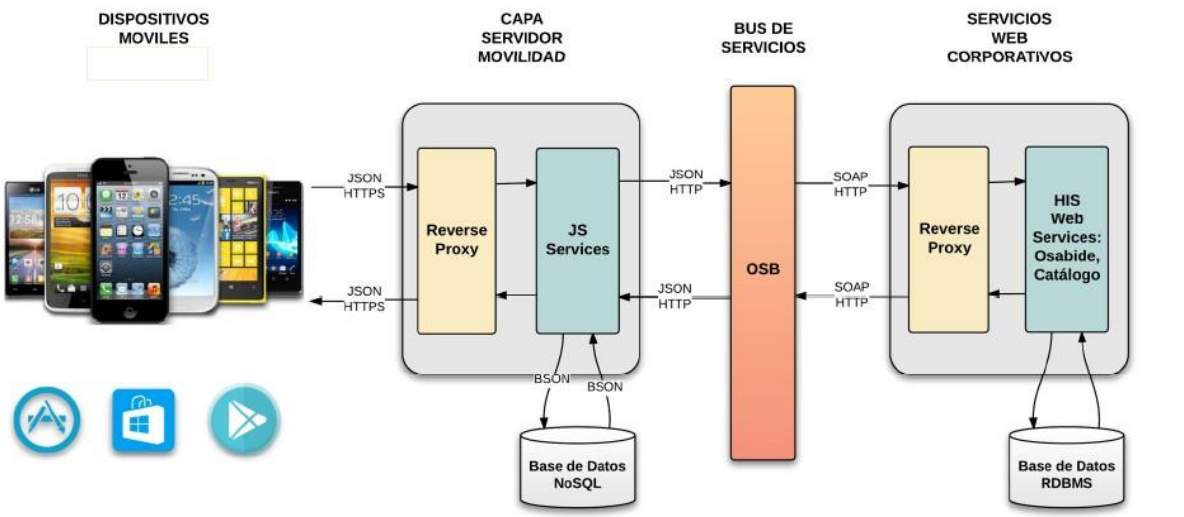
- Tecnologías para el desarrollo en cliente: aplicaciones para dispositivos móviles tales como tablets, smartphones (iOS, Android)

- Tecnologías para el desarrollo en servidor: servicios para la integración con servicios backend, seguridad y lógica de negocio movilidad.

Las características comunes a todas las tecnologías de movilidad son:

- Licencia de código abierto (open source)
- Permiten el desarrollo de aplicaciones multiplataforma
- Lenguajes de programación JavaScript, HTML5, CSS3

En la siguiente figura se muestra la **vista lógica** de la arquitectura de referencia para aplicaciones en movilidad.



12.6.1 Tecnologías Cliente

12.6.1.1 Angular

AngularJS es un framework de JavaScript de código abierto, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.
La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticas o dinámicas.

12.6.1.2 Ionic Framework 5 o superior

Ionic es un SDK completo de código abierto para el desarrollo de aplicaciones móviles híbridas. Construido sobre AngularJS y Apache Cordova, Ionic ofrece herramientas y servicios para el desarrollo de aplicaciones móviles híbridos utilizando tecnologías web como CSS, HTML5, y Sass.
Las aplicaciones pueden ser construidas con estas tecnologías Web y luego distribuidas a través de las tiendas de aplicaciones nativas para ser instalado en los dispositivos mediante el aprovechamiento de Capacitor.

12.6.2 Tecnologías Servidor

12.6.2.1 Nginx

Nginx es un servidor web/proxy inverso ligero de alto rendimiento.

Actualmente Nginx se usa en más de 100 millones de sitios web.

Es un software moderno y su adopción ha sido muy rápida debido a su implementación ligera, que proporciona unos tiempos de respuesta muy bajos con un consumo de recursos hardware mínimo.

12.6.2.2 Node.js

Node.js es un entorno de programación en la capa del servidor basado en el lenguaje de programación ECMAScript (JavaScript), asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Node.js es ligero y eficiente ejecutando aplicaciones que hacen uso intensivo de grandes cantidades de datos en tiempo real y corren en dispositivos distribuidos.

12.6.2.3 MongoDB

Mongodb es una base de datos No SQL open source. Mongodb está orientada a documentos, en vez de guardar los datos en tablas como se hace en las bases de datos

relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico.

12.7 Calidad de uso

Rendimiento

Se exigirá que el rendimiento de la aplicación en uso (producción) cumpla y se ajuste a los requisitos no funcionales y al plan de pruebas diseñado en etapas tempranas. En cualquier caso, el comportamiento de la aplicación deberá cumplir de forma obligatoria los siguientes criterios en la ejecución de las pruebas técnicas.

INDICADOR	DESCRIPCIÓN	DESVIACIÓN
Prueba de escalabilidad	El tiempo de respuesta de la aplicación deberá mantenerse invariable con el incremento de usuarios concurrentes(a menos que exista alguna saturación de recursos del sistema), al menos hasta alcanzar la carga pico estimado.	+/- 10%
Prueba de estrés	Tras someter al sistema al doble de carga nominal, tras la disminución de dicha carga a la carga normal de trabajo (80%), la aplicación deberá garantizar su estabilidad y recuperación sin necesidad de reinicio.	+/- 10%
Prueba de estabilidad	La aplicación deberá garantizar un tiempo de respuesta estable manteniendo una carga nominal normal (80 %) durante al menos 12 h	+/- 10%

Usabilidad

Se exigirá que la aplicación mantenga unos criterios de usabilidad que permitan:

- Facilitar el aprendizaje de la aplicación sin requerir conocimientos técnicos en su uso. El sistema deberá ser fácil de usar, asegurando unos bajos costes de aprendizaje y de asistencia o ayuda al usuario.
- Asegurar la velocidad de uso de la aplicación permitiendo el acceso a la información en el menor número posible de clics de ratón.

- El interfaz de usuario deberá ser amigable de forma que se incremente la satisfacción y la productividad de los usuarios.
- Optimizar los costes de diseño, rediseño y mantenimiento.

La ISO 9241-11:1998 “Guidance on usability”, define la usabilidad como: " *La medida con la que un producto se puede usar por usuarios determinados para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso concreto*"

Osakidetza exigirá el cumplimiento de las directrices de la ISO 9241-151 en cuanto al cumplimiento del grado de usabilidad a exigir en el desarrollo de la aplicación

Accesibilidad

El cumplimiento de la accesibilidad de la aplicación incorporará posibilidades de navegación que garanticen el acceso a la información y a los servicios proporcionados minimizando al máximo las limitaciones y/o restricciones por razón de discapacidad de cualquier carácter o condicionantes técnicos, atendiendo así a la normativa existente.

Se deberán aplicar todas las pautas correspondientes al nivel de accesibilidad AA, llegando a un nivel AAA en las secciones que la configuración de los contenidos y diseño lo permita (Sede electrónica). La accesibilidad se regirá por las pautas WAI indicadas por el Word Wide Web Consortium (W3C).

La accesibilidad no se refiere únicamente a personas discapacitadas. Un sitio web accesible y que se adapte a diferentes tipos de usuario podrá ser visitado por más personas; esta flexibilidad también da beneficios a personas con discapacidades temporales (una convalecencia o enfermedad, problemas de visión), personas de edad avanzada, con una conexión lenta a Internet, que utilizan un dispositivo móvil, etc.

Compatibilidad y portabilidad

Osakidetza exigirá la compatibilidad de la aplicación con los siguientes navegadores web al menos en su última versión estable del fabricante y al menos dos anteriores: Internet Explorer, Firefox, Google Crome, Safari y Mozilla.

La empresa adjudicataria garantizará el funcionamiento de la aplicación con estos navegadores y se responsabilizará de la actualización de la aplicación en caso de necesidad para garantizar la compatibilidad de la misma con estos navegadores durante toda la ejecución del contrato y su periodo de mantenimiento.

El interfaz debe visualizarse independientemente del sistema operativo utilizado por el cliente.

Si procede el diseño deberá ser multi-dispositivo, es decir, accesible y visible desde cualquier dispositivo tipo teléfono móvil, tablet, etc.

13 ANEXO III. CUESTIONARIO DE PERSONAL

Este formulario se deberá cumplimentar por cada profesional propuesto:

Empresa licitante:									
Profesional:									
Categoría ofertada:									
Antigüedad	**Empresa**	**Categoría**	**F. Alta**	**F. Baja**	**Actividad Informática**				
Titulación Académica	**Título académico**	**Centro**	**Años**	**Fecha Expedición**	**TIC S/N**				
Actividades desarrolladas	**Nombre proyecto**	**Perfil (*1)**	**F. Inicio**	**F. Fin**	**Categoría**	**Entidad Usuaria**	**Funcionalidad (*2)**	**Tecnología (*2)**	**Contenido del Trabajo**

(*1) En un mismo proyecto, puede haber trabajado en diferentes perfiles.
(*2) Funcionalidad y Tecnología se refiere a los módulos funcionales y tecnologías con los que ha trabajado por cada proyecto.

Documento firmado por: SUSANA
IGLESIAS TAMAYO
En la fecha 10.11.2025 09:20:47 CET