



Eusko Jaurlaritzaren
Informatika Elkarte

Sociedad Informática
del Gobierno Vasco



Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

Fecha: 10/07/2025

Este documento es propiedad de Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, S.A. (EJIE). Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Versión	Fecha	Resumen de cambios	Elaborado por:	Aprobado por:
1.0	Julio 2025	Versión Inicial	Javier Pérez Toral	Ander Martinez Juan Jose Villa

Índice

1	Introducción	5
2	Objeto del contrato.....	6
2.1	Objeto del contrato.....	6
2.2	Requisitos de los servicios de Asistencia técnica.....	6
3	Definición del Servicio	8
3.1	Fases del Servicio.....	8
4	Configuración del Servicio	11
4.1	Organización del Equipo de Trabajo	11
4.2	Asignación de recursos a fases de proyecto	13
4.3	Composición del equipo de trabajo	14
4.4	Formación	14
4.5	Transferencia Tecnológica.....	15
4.6	Control y Seguimiento.....	15
5	Metodología aplicable y Entorno Tecnológico	16
5.1	Especificaciones para Desarrollo	16
5.1.1	<i>Arquitectura de Referencia</i>	<i>16</i>
5.1.1.1	<i>Vista lógica</i>	<i>17</i>
5.1.1.2	<i>Directrices de Desarrollo</i>	<i>18</i>
5.1.1.3	<i>Construcción de Aplicaciones por Capas.....</i>	<i>18</i>
5.2	Modelo de aplicaciones modernas	22
5.2.1	Especificaciones de desarrollo.....	31
5.2.2	Entorno tecnológico.....	33
5.2.2.1	Contenedores	33
5.2.2.2	Orquestación de contenedores	34
5.2.2.3	Ciclo de vida	35
5.2.2.4	Plataformas horizontales.....	37
5.2.2.5	Cloud	38
5.3	Directrices tecnológicas de Infraestructuras	39
5.3.1	Entorno tecnológico.....	40
5.3.2	Especificaciones del Cliente (Maqueta)	42
5.3.3	Dispositivos de entrada/salida.....	45
5.3.4	Seguridad, acceso, autenticación y autorización	45
5.3.5	Monitorización.....	46

5.3.6	<i>Backup/Restore</i>	47
5.4	Interoperabilidad	48
5.4.1	<i>Servicios Web</i>	49
5.4.2	<i>Gestor de Eventos</i>	50
Anexo 1. Asistencia Técnica Calidad Osakidetza		¡Error! Marcador no definido.

1 Introducción

EJIE, Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, es la Empresa pública de servicios de las tecnologías de la información y las comunicaciones (TIC), cuya razón de existir es contribuir a la consecución de un Sector Público Vasco, moderno y eficiente, en el Marco Legal establecido por el Gobierno, con la seguridad y calidad necesarias y con el debido respeto al medio ambiente.

EJIE tiene como misión ser el órgano gestor tecnológico del Gobierno Vasco que facilita la digitalización de los servicios que el sector público vasco presta a la ciudadanía, garantizando la calidad, seguridad y continuidad de las tecnologías de la información y la comunicación que los soportan, alcanzando resultados satisfactorios en relación a las entidades cliente de los servicios y sus personas usuarias.

EJIE tiene como meta final la consecución de la satisfacción de sus clientes, siendo el instrumento común de prestación de servicios TIC en el Sector Público Vasco, y comprometiéndose en:

- Construir y mantener con eficiencia y calidad la infraestructura de los Sistemas de Información, posibilitando su continuidad y seguridad.
- Garantizar la interoperabilidad entre las distintas administraciones.
- Servir de apoyo a las necesidades de planificación y realización de la función informática del sector público del Gobierno Vasco, asegurando la cobertura de sus demandas con el compromiso y profesionalidad adecuados a las relaciones contractuales que se establezcan.

Se puede obtener información más detallada y extensa en nuestra dirección de Internet <http://www.ejie.eus>

Por tanto, EJIE es la referencia para la prestación de servicios TIC en el Sector Público Vasco:

- Aportando valor añadido.
- Proporcionando soluciones competitivas.
- Transmitiendo confianza a sus clientes.
- Contando con personas cualificadas y comprometidas.

Con fecha 16 de agosto de 2016 se firmó el Acuerdo entre el Ente Público de Derecho Privado Osakidetza y Eusko Jaurlaritzaren Informatika Elkartea—Sociedad Informática del Gobierno Vasco, S.A. (EJIE) para la integración del servicio de infraestructuras y operaciones (I.O.) de OSAKIDETZA con el análogo servicio (I.O.) en EJIE, en el marco del acuerdo adoptado por el Consejo de Gobierno sobre el proceso de convergencia en materia de tecnología de la información y comunicación –TIC/ICPS el 27 de julio de 2015 y del Acuerdo de 21 de junio de 2016 relativo a dicho proceso de convergencia por el que se aprueba el documento ejecutivo “plan general de actuación en materia de convergencia TIC/ICPS”.

La actividad objeto de contratación, es necesaria para la prestación de los servicios, competencia de EJIE S.A., y está incluida en los servicios que pueden ser objeto de contratación (según Instrucción Sexta de la Instrucciones sobre buenas prácticas en la contratación de servicios de Gobierno Vasco). Su prestación no puede ser asumida con los recursos humanos y técnicos de que dispone EJIE S.A., siendo además inviable su reorganización.

El presente pliego, como el resto de los documentos de la fase de preparación del contrato, incorporan las debidas garantías para dar cumplimiento a la Instrucción octava de las Instrucciones sobre buenas prácticas en la contratación de servicios de Gobierno Vasco.

En este ámbito, el Osakidetza encomienda a EJIE los **servicios de asistencia técnica profesional necesaria** para asegurar el correcto funcionamiento, **calidad** y evolución de los sistemas de información que dan soporte a los procesos de gestión dentro de su ámbito competencial.

2 Objeto del contrato

2.1 Objeto del contrato

La finalidad de la contratación es dotar a EJIE de los servicios de asistencia técnica profesional necesarios para asegurar **la calidad** de los sistemas de información que dan soporte a los procesos de Osakidetza.

La necesidad de contratación se enmarca en el servicio de Asistencia Técnica prestado por EJIE a dichos Departamentos, Organismos Autónomos y Entes Públicos de derecho privado del Gobierno Vasco, gestionados por el área técnica Educativa Socio-Sanitaria de EJIE que tiene como objetivo proporcionarles la asesoría y el apoyo técnico especializado preciso para la definición de planes y proyectos informáticos, así como para el desarrollo, mantenimiento e implantación de los Sistemas de Información. A través de este servicio, EJIE también ayuda a la Dirección de Tecnologías de la Información y la Comunicación para velar por el cumplimiento de los estándares y la adecuada implantación de las aplicaciones.

Para la prestación de los servicios de Asistencia Técnica, EJIE requiere contratar servicios profesionales de soporte técnico especializado en tecnologías de la información.

El **entorno de referencia** para los que se requiere el servicio de asistencia técnica se establece en el **Anexo 1** recogido en este documento.

2.2 Requisitos de los servicios de Asistencia técnica

Para la prestación del servicio de Asistencia Técnica EJIE requiere de la contratación de servicios profesionales de proveedores externos, aunque siempre sujeto a la decisión, control y seguimiento de personal de EJIE. Los servicios requeridos contemplan la realización de las siguientes tareas:

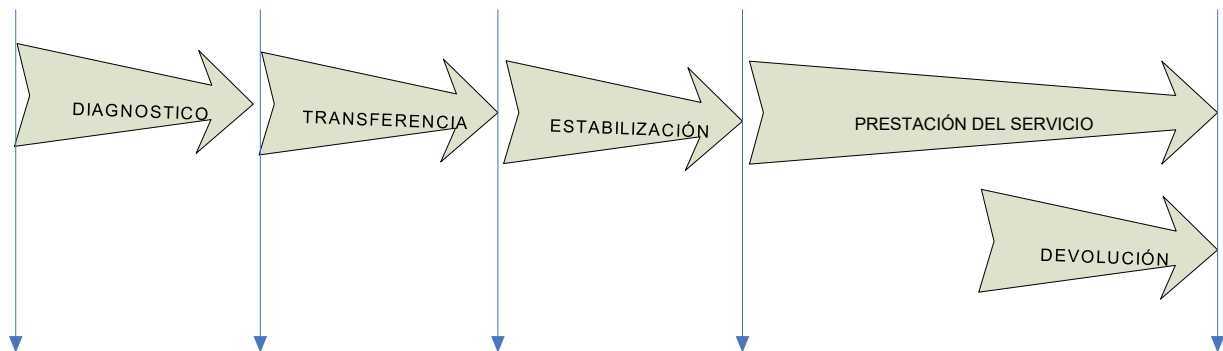
- **Soporte técnico informático especializado “bajo demanda”** en las actividades de aseguramiento de calidad del software. Este modelo bajo demanda aplica al resto de tareas definidas.
- **Soporte técnico a pruebas de calidad de software**, teniendo en cuenta los siguientes aspectos:
 - Alcance de las pruebas: cual es el objeto de la prueba, es decir, aplicaciones, módulos, funcionalidad, sistemas, subsistemas, interoperabilidad, alta disponibilidad, que serán objeto de prueba, según el caso que se plantee: correctivo, nueva funcionalidad, evolución tecnológica de plataformas/productos comerciales de software, evolución/sustitución de infraestructuras o puesto de trabajo. Cuando sea necesario, la empresa adjudicataria deberá **concertar** el alcance de las pruebas con el desarrollador de los módulos correspondientes o el titular de los derechos de propiedad intelectual del producto.
 - Tipos de Pruebas: la diversidad de software requiere, según el caso, realizar auditorias de código, así como la realización completa o en parte de los siguientes tipos de pruebas:
 - Pruebas funcionales. para probar que el software desarrollado cumple con las funciones específicas para las que ha sido creado.
 - Pruebas no funcionales. referentes a rendimiento, escalabilidad, seguridad, alta disponibilidad y continuidad de negocio, usabilidad, etc.
 - Pruebas de integración: para comprobar la integración del software globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de los sistemas de información con los que se comunica.
 - Pruebas de aceptación: pruebas para asegurar que un sistema cumple con el funcionamiento esperado y permitan al usuario final determinar su aceptación, desde el punto de vista de funcionalidad y de rendimiento.
 - Pruebas de regresión: para evaluar que el sistema continúa operando de manera correcta y de acuerdo con lo especificado después de introducir una nueva funcionalidad o de realizar un cambio al código.
 - Pruebas de seguridad. son un conjunto de actividades y técnicas destinadas a identificar, evaluar y mitigar vulnerabilidades, amenazas y riesgos en un sistema o aplicación. Su objetivo es garantizar que el software proteja la confidencialidad, integridad y disponibilidad de los datos, y que resista ataques o accesos no autorizados durante su ciclo de vida.

- **Análisis, diseño y planificación de las pruebas:** Detalle de los niveles de pruebas y tipos de pruebas a realizar, métricas, herramientas que se van a utilizar, qué se automatiza y por qué, qué y cómo serán tratados aquellos elementos que no se pueden probar; definición del entorno de ejecución de las pruebas detallando los recursos hardware y software necesarios, criterios de inicio, suspensión y finalización de las pruebas, posibles riesgos, equipo de pruebas con sus roles y responsabilidades, estimación de todas las actividades a realizar acotadas en tiempo y recursos asociados a las mismas, relación de entregables, ...
 - **Ejecución y evaluación de resultados:** actividades para la ejecución de las pruebas (preparación del entorno, juegos de datos, relación de métricas, ...); detección y registro de defectos o incidencias encontrados, así como la evaluación del software en base a los resultados y posibles recomendaciones de mejoras. Seguimiento y control de estos consensuado con EJIE.
 - **Estandarización y automatización de los procesos y tareas** fruto de la aplicación de la metodología y estrategia de pruebas, de modo que el trabajo realizado sea fácilmente reutilizable en otros casos o repeticiones de estas.
 - **Desarrollo de métricas de calidad** para hacer el seguimiento de la calidad del software, generales y específicas de cada proyecto.
 - **Coordinar y mantener los controles de seguridad de las aplicaciones**, realizando como mínimo, el mantenimiento de seguridad en los pipelines de DevOps, formar y dar soporte a los perfiles de seguridad de los equipos de desarrollo, gestionar y documentar la resolución de vulnerabilidades, elaborar informes de seguimiento y evolución de la seguridad, y promover la mejora continua de la arquitectura de seguridad conforme a los estándares OWASP SAMM y DSOMM, en alineación con los estándares de calidad y seguridad del software reconocidos (ISO/IEC 25000, ISO 5055).
 - **Actualización del catálogo de servicios** para ser consumido por los clientes con entradas y salidas perfectamente definidas.
 - **Proporcionar las herramientas** y su gestión, necesarias para realizar todas las tareas que se especifiquen en la metodología y estrategia de pruebas, así como las requeridas para soportar la propia Calidad de Software, objeto del expediente, siendo preferente el uso de software libre y la valoración de las herramientas que usa EJIE para otros Departamentos. La decisión final de la mismas debe ser consensuada con EJIE. El servicio vigente utiliza **Testmo** como plataforma para la gestión de calidad del software. De manera inicial es requisito su mantenimiento mientras no se proponga un cambio consensuado.
 - **Revisión y mejora continua:** Se deberá contemplar una revisión periódica tanto de la metodología como de la estrategia de pruebas para determinar mejoras sobre las mismas, adaptarse y adoptar evoluciones tecnológicas futuras de todos aquellos elementos que las componen, así como las del software de entorno de negocio/funcional de referencia.
- Osakidetza dispone de un Modelo de Pruebas de Osakidetza del servicio vigente. Es necesario revisar y actualizar el mismo de manera **semestral** para adaptarse a las necesidades tan cambiantes y por lo tanto para asegurar que el software de Osakidetza cumple los requerimientos especificados, y las necesidades y expectativas del usuario, es decir, asegurar la calidad del software.
 - Soporte al personal de EJIE para las actividades de **gestión de configuración, cambios y despliegue de aplicaciones** en los distintos entornos de despliegue de aplicaciones. El personal de EJIE recibe de los equipos de mantenimiento o proyectos de sus clientes las modificaciones o necesidades de rediseño del software de las distintas aplicaciones de negocio. Para su despliegue se requieren servicios de asistencia técnica para su correcta configuración y parametrización en producción. Asimismo, es habitual hacer modificaciones en dicha configuración y despliegue para la actualización del software.
 - Soporte al personal de EJIE para las actividades de **pruebas de carga** de aplicaciones. El personal de EJIE recibe de los equipos de mantenimiento o proyectos de sus clientes solicitudes de pruebas de carga de las distintas aplicaciones de negocio. Para su preparación puede ser necesario reutilizar los casos de uso definidos dentro del servicio de calidad. En caso de que este servicio no exista en EJIE **puede ser solicitado como un tipo de prueba más** dentro del servicio de calidad.

3 Definición del Servicio

3.1 Fases del Servicio

El modelo de prestación del servicio se ha dividido en varias fases consecutivas, estableciendo un período inicial de adquisición progresiva de conocimientos, una fase central de prestación normal del servicio, y una fase final orientada a la transición de responsabilidades hacia a EJIE o hacia un nuevo adjudicatario.



Secuencia de fases del servicio

Aunque para cada una de las fases se especifica un conjunto salidas, estas deben considerarse como orientativas. Previamente al inicio del servicio se concretará y acordará entre el adjudicatario y el responsable del servicio de EJIE cuáles de estos productos de salida deben ser obtenidos, o incluso otros nuevos que se consideren necesarias dada las características particulares del servicio.

FASE 1: DIAGNÓSTICO

Esta fase inicial del servicio facilitará al adjudicatario una primera toma de contacto en la cual podrá obtener datos actuales sobre las características del proyecto/servicio. En colaboración directa con el Gestor de EJIE actuando como facilitador y proveedor de toda la documentación e información que se requiera, el adjudicatario podrá planificar y preparar la fase de transferencia, obtener información detallada del servicio, así como las necesidades derivadas de gestión y organización que garanticen, en definitiva, el éxito de la prestación del servicio para el que ha sido contratado.

Con objeto de disponer de una visión global sobre el estado inicial de las aplicaciones objeto del contrato, el adjudicatario elaborará un informe resumen en el que se recojan los aspectos más relevantes del diagnóstico realizado. El índice y contenido de dicho informe será propuesto por el adjudicatario, que deberá ser aprobado por los Gestores de EJIE y Osakidetza.

Salidas:

- Informe de diagnóstico de situación actual

FASE 2: TRANSFERENCIA

En esta fase, la empresa adjudicataria después de completar el acopio de todos los elementos necesarios para la prestación del servicio iniciado en la fase anterior, deberá asegurarse el conocimiento detallado de todos los proyectos de calidad implementados en el servicio anterior.

Se elaborará y ejecutará un plan detallado de reuniones de transferencia del conocimiento con los responsables técnicos de EJIE, asegurando así el dominio detallado de la operativa de los sistemas, y el de sus características técnicas.

Será responsabilidad de la empresa adjudicataria reinstalar en sus propias dependencias, si fuese necesario, el conjunto de herramientas y aplicaciones proveyéndose así de la autonomía suficiente para realizar las actividades 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

que le sean requeridas. Se contempla la opción de conexión VPN, aunque no se descartan otras posibles alternativas que deberán ser valoradas.

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

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

Como en la fase anterior, el adjudicatario deberá documentar adecuadamente todas las actividades del proceso de transferencia.

Salidas:

- Informe de resultados de la fase de transferencia

FASE 3: ESTABILIZACIÓN

Una vez adquiridos los conocimientos necesarios, y montadas las infraestructuras de soporte, el adjudicatario podrá ejecutar las peticiones de trabajo que se le requieran por parte de los Gestor de EJIE y Osakidetza. No obstante, y con el fin de garantizar que el servicio de mantenimiento se resolverá adecuadamente, se incluye un período de estabilización y acompañamiento (voluntario, a decisión del adjudicatario) del personal técnico de EJIE que permita realizar los ajustes finales, y consolide por fin los flujos de trabajo que se aplicarán en la fase de prestación del servicio.

En esta fase el adjudicatario consolidará definitivamente 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.

El plazo máximo de ejecución de estas tres primeras fases (diagnóstico, transferencia y estabilización) será de 1 meses, y los servicios incluidos en ellas NO SERÁN FACTURABLES. Cualquier modificación a este respecto deberá ser justificada por la empresa adjudicataria y aprobada por EJIE.

Salidas:

- Organización final del servicio
- Flujos de trabajo consolidados

FASE4: PRESTACIÓN DEL SERVICIO

Se trata de la fase operativa del servicio. Una vez superada la fase de estabilización, el adjudicatario deberá funcionar ya de manera autónoma, resolviendo adecuadamente y en los plazos marcados las peticiones de trabajo que se le encomienden, gestionando los recursos dedicados, medición de los indicadores acordados, realización de las reuniones de seguimiento, obtención de informes de actividad, etc., garantizando en definitiva la calidad del servicio para el que ha sido contratado.

Salidas:

- Las propias de las actividades objeto de la contratación

FASE 5: DEVOLUCIÓN

La fase de devolución asegurará que el adjudicatario pondrá a disposición de EJIE o de un tercero, de manera ordenada, todos aquellos artefactos software, documentación, inventarios, repositorios, etc. que han sido utilizados durante la prestación del servicio. Se garantizará así que EJIE o un tercero pueda asumir las mismas actividades de mantenimiento en un tiempo suficiente.

En esta fase de devolución a EJIE o a un tercero se fija en **1 meses**, coincidiendo por tanto con dichas fases iniciales para quien asuma entonces el servicio de mantenimiento. Se entiende entonces que será tarea del

adjudicatario colaborar activamente en este período de transición. En cualquier caso, EJIE podrá reducir el tiempo estimado para la fase de devolución del servicio si lo considerara necesario.

Para garantizar este objetivo, a lo largo del periodo que dure el contrato, se realizará un Plan de Devolución que será revisado y aprobado progresivamente por los Gestores de EJIE y Osakidetza.

Salidas:

- Plan de devolución, realizado

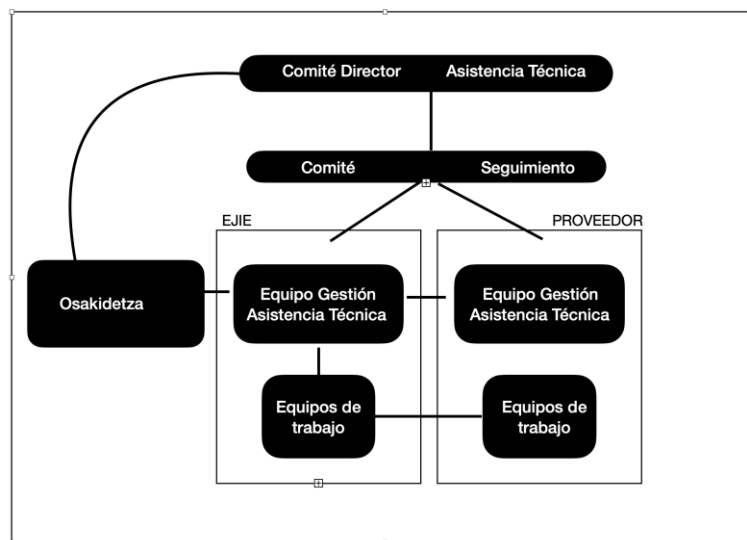
4 Configuración del Servicio

4.1 Organización del Equipo de Trabajo

La empresa licitadora deberá describir en su Documento de Propuesta Técnica:

- La organización (perfiles) del equipo de proyecto asignado a la realización de las actividades resultantes del presente pliego, así como
- Las funciones de estos.

No obstante, se detalla a continuación una posible aproximación de modelo organizativo:



COMITÉ DIRECTOR ASISTENCIA TÉCNICA

Ostenta la máxima autoridad en el proyecto y representa a las organizaciones involucradas en el mismo, EJIE y la empresa adjudicataria. A este comité se incorporarán las personas que en cada momento estime oportuno el propio Comité, principalmente de las áreas representadas en el servicio.

Participantes:

- EJIE:
 - Dirección de Área de EJIE o director técnico
 - Jefe/a de proyecto responsable Osakidetza
- OSAKIDETZA:
 - Jefe/a de servicio responsable
- Empresa adjudicataria:
 - Director/a de Proyecto de la empresa adjudicataria en representación de la Dirección
 - Responsable del proyecto/servicio

Periodicidad de las reuniones del Comité:

- Trimestral. Deberá quedar constancia de un acta de reunión y de los informes facilitados por la empresa adjudicataria.

Roles y responsabilidades:

- Liderar y apoyar la consecución de los objetivos fundamentales.
- Resolver los conflictos o problemas que, por cualquier razón, no puedan hacerse desde el comité de Seguimiento.
- Definir o aprobar los indicadores de control cuyo seguimiento se va a realizar.
- Evaluar y aprobar, en su caso, las posibles variaciones de alcance, plazos, coste y requerimientos que pudiesen surgir
- Asegurar la disponibilidad de los medios y recursos necesarios para la ejecución del proyecto.
- Seguimiento exhaustivo de la prestación del servicio de Asistencia Técnica, evaluando y aprobando las mejoras a aplicar.

COMITÉ DE SEGUIMIENTO

Representa la máxima autoridad en relación a la operativa del proyecto, realizando un seguimiento del trabajo (planificaciones y recursos), de las problemáticas surgidas, escalando los aspectos que se consideren que sobrepasan los límites de actuación establecidos.

Participantes:

- EJIE:
 - Jefe/a de proyecto responsable Osakidetza
 - Otro personal de EJIE que, a solicitud del Jefe de Proyecto, fuera requerido incluir en este comité.
- OSAKIDETZA:
 - Jefe de proyecto responsable
- Empresa adjudicataria:
 - Responsable del proyecto/servicio

Periodicidad de las reuniones del Comité:

- Mensual. Deberá quedar constancia de un acta de reunión y de los informes facilitados por la empresa adjudicataria.

Roles y responsabilidades:

- Seguimiento permanentemente del servicio.
- Seguimiento del plan de formación establecido para el servicio.
- Gestión y seguimiento de los trabajos encargados
- Agilizar la toma de aquellas decisiones que, por su naturaleza, pudieran exceder de las competencias del Equipo de Trabajo.
- Elevar al comité de dirección las decisiones o incidencias que consideren necesarias.
- Elaborar los Informes de progreso con el fin de informar del desarrollo del servicio.

EQUIPOS DE TRABAJO DE LA EMPRESA ADJUDICATARIA

Para la realización de los servicios especificados en el Objeto del contrato, el adjudicatario dispondrá de un equipo de trabajo, que estará formado por perfiles con capacidad, experiencia, y conocimientos necesarios para el correcto desempeño de su trabajo.

Participantes:

- Con carácter obligatorio para todos los casos:
 - Responsable del servicio (Dirección).
- Con carácter opcional en función de la demanda concreta del ámbito de la aplicación, el equipo estará compuesto por los siguientes perfiles:
 - Jefe/a de proyecto / Consultor de testing (La jefatura del proyecto ha de ser asumida por el consultor de testing).
 - Técnicos de testing Experto Automatización.
 - Técnicos de testing Pruebas manuales.
 - Coordinador Seguridad asociada a la calidad de software.

NO existen necesidades iniciales de servicio, **es un servicio bajo demanda**, que se solicitaran por parte de EJIE, de acuerdo con las necesidades que se den en cada momento, siempre en el ámbito de las actividades planteadas en este pliego de condiciones técnicas y de acuerdo a las demandas Osakidetza.

Para cada lote, cada licitador deberá proponer un **responsable de servicio** (no evaluable). Este responsable de relación será **una persona** de su organización **encargada** de la gestión del contrato adjudicado al amparo del presente expediente. De modo adicional dicha persona responsable deberá ejercer las facultades de dirección, organización y disciplinarias en lo referente a las personas que presten los servicios contratados, así como de gestionar y coordinar la actividad preventiva, en base a los procedimientos establecidos por EJIE en www.ejje.eus. En caso de incumplimiento de cualquiera de estas obligaciones, EJIE se reserva el derecho de resolver el contrato.

4.2 Asignación de recursos a fases de proyecto

El licitador deberá incluir en su Documento de Propuesta Técnica, un desglose de horas y el % de dedicación total por perfil y fase del proyecto, siguiendo exactamente el siguiente modelo:

Descripción Perfil	Fase Diagnóstico		Fase Transferencia		Fase Estabilización		Fase Prest. del servicio	Fase Devolución		Total Horas
	Horas	%	Horas	%	Horas	%	%	Horas	%	
Jefatura de Proyecto/Consultor técnico										
Técnicos de testing Experto Automatización										
Técnicos de testing Pruebas manuales										
Coordinador Seguridad asociada a la calidad de software										
TOTAL										

- Este desglose de horas se considerará como orientativo y será tenido en consideración en el momento de valorar el grado de aproximación a la planificación del proyecto según la estimación de la empresa licitadora, permitiendo, de esta forma, valorar la idoneidad del dimensionamiento del equipo de trabajo propuesto y su adecuación a la consecución de los objetivos. No obstante, este desglose de horas no se considera vinculante.
- La fase de prestación, al ser bajo demanda, no se solicita desglose de horas pero sin % para la valoración de la fase de prestación del servicio.
- **Se deberán ofertar los perfiles indicados en la tabla y no otros.**

- Para cada uno de los perfiles, deberá detallarse el número de horas y % de dedicación de cada uno de los miembros del equipo propuesto en función de la categoría asignada

De manera puntual, en aquellos casos en que los trabajos deban ser realizados en las dependencias de EJIE, estos se realizarán en las siguientes condiciones:

El horario de atención telefónica, así como la atención a peticiones de mantenimiento correctivo se ajustará a la jornada de trabajo establecida por EJIE.

Las actividades derivadas de las actuaciones fuera del horario laboral para implantaciones, correcciones de software, etc. tendrán la misma consideración que aquellas incurridas en el horario del servicio a efectos de estimaciones y justificaciones de esfuerzos de cara a los procedimientos de gestión del modelo de servicio con EJIE.

Las necesidades excepcionales de ampliación del horario de disponibilidad de los entornos de EJIE en base a su categorización, o de sus servicios de soporte u operación, deberán ser acordados con antelación suficiente con el Gestor de EJIE, que será el responsable de realizar las peticiones que sean necesarias. En cualquier caso, solo se contemplarán las necesidades recogidas en el catálogo de servicios de EJIE.

4.3 Composición del equipo de trabajo

El equipo de trabajo propuesto estará formado por personal técnico con categoría profesional y nivel de especialización adecuados a la solvencia requerida, y a los servicios especificados en este pliego de condiciones técnicas.

La totalidad del equipo presentado deberá estar disponible para la incorporación de forma inmediata a la formalización del pedido/contrato por parte de EJIE.

4.4 Formación

Es objeto del Servicio de Asistencia Técnica de EJIE a su clientela, ayudar a la Dirección de Tecnologías de la Información y la Comunicación para velar por el cumplimiento de los estándares y la adecuada implantación de las aplicaciones. Para poder realizar las tareas de asistencia especificadas en el objeto de este contrato, es responsabilidad del adjudicatario garantizar la formación adecuada del personal del mismo asignado al equipo de trabajo.

En el inicio de la prestación del servicio, se establecerá con la empresa adjudicataria en el Comité de Seguimiento un **plan de formación específico**. Además, en todos los casos, de forma **obligatoria**.

En este sentido, la empresa adjudicataria deberá acreditar con carácter general la **realización de al menos 30 horas de formación anuales por cada persona miembro del equipo de trabajo**, contabilizando dentro de dichas horas aquellas que se realicen en los ámbitos establecidos de forma específica al comienzo de la prestación del servicio en el Comité de Seguimiento.

Deberá proponerse por parte del adjudicatario un **plan** de las acciones formativas. Dicho proceso deberá ser **concretado en el primer mes de la prestación del servicio** y podrá ser sometido a modificaciones a propuesta tanto del adjudicatario del contrato como de la persona responsable del servicio objeto de licitación en EJIE, teniendo en cuenta las necesidades que se plantean en cada momento en Osakidetza.

Cualquier coste que pudiera derivarse de las acciones formativas del personal asignado al servicio demandado será responsabilidad del proveedor y no serán imputables a EJIE.

4.5 Transferencia Tecnológica

Durante la ejecución de los trabajos objeto del contrato el adjudicatario se compromete, en todo momento, a facilitar a las personas de EJIE designadas por la persona responsable del servicio a prestar en EJIE (Jefe de Proyecto), y a tales efectos, la información y documentación que ésta solicite para disponer de un pleno conocimiento de los trabajos desarrollados, así como de los eventuales problemas que puedan plantearse y de las tecnologías, métodos, y herramientas utilizados para resolverlos.

Asimismo el adjudicatario y antes de la finalización del servicio, se compromete a realizar las actividades formativas, que garanticen la continuidad del servicio, al equipo de EJIE u Osakidetza que pudiera dar soporte a este servicio en cada momento, tras la finalización del servicio al que se refiere el presente pliego, según las directrices marcadas por EJIE.

4.6 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. Es en este sentido en el cual llamamos seguimiento a la evaluación rutinaria del estado, en tanto que llamamos control a la toma de los valores que definen el estado.

Los productos mínimos de esta práctica son los informes de seguimiento, un documento o artefacto, donde anotamos los resultados de la evaluación de una iteración de control; de momento sin decir las correcciones a tomar.

Una vez identificadas las desviaciones, es necesario decidir oportunamente las correcciones requeridas, llevándolas a cabo en el momento en que sea necesario. Finalmente es importante que la corrección planteada sea a su vez, objeto de seguimiento, lo que implica que la planificación debe ser actualizada para que refleje las acciones que se han determinado necesarias para corregir la desviación.

A la hora de realizar las reuniones de seguimiento es conveniente calcular previamente las medidas o métricas que aplicadas al servicio objeto de la contratación sirvan de indicador sobre el estado del mismo. Esto con el objeto de obtener una evaluación lo más completa y objetiva posible de la prestación del servicio.

Así, bajo esta perspectiva, la empresa licitadora deberá describir en su Documento de Propuesta Técnica:

- 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/las asistentes necesarios/as, 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

Previamente al inicio de la prestación del servicio se concretará por el responsable del servicio de EJIE cuáles serán las tareas de Control y Seguimiento que deben realizarse, los entregables a generar y la periodicidad de las reuniones a celebrar. Aun así, **deberá al menos existir una reunión de seguimiento con carácter mensual del Comité de Seguimiento**, quedar constancia de la misma con el consecuente informe de seguimiento y acta de reunión asociada.

5 Metodología aplicable y Entorno Tecnológico

La organización del trabajo y la prestación del servicio de **Asistencia Técnica**, tal y como se ha mencionado con anterioridad, el nivel de cumplimiento de los mismos dentro de los servicios a prestar deberá concretarse en las fases iniciales del proyecto. Aun así, las ofertas presentadas deberán tomar como referencia dichos modelos y metodologías, que se explican brevemente a continuación.

El entorno tecnológico de referencia será el establecido en los estándares de Osakidetza. Se deberán aplicar por tanto aquellos que se consideren necesarios para la ejecución del servicio, teniendo en cuenta como referencias las características específicas de las aplicaciones incluidas en el Anexo 1 del presente pliego de condiciones técnicas.

La empresa adjudicataria se compromete a adoptar y soportar en todo momento cualquier modificación o adaptación del entorno tecnológico establecido por Osakidetza e incluido dentro de sus estándares, así como todas aquellas otras actualizaciones específicas impuestas por el responsable del Área Funcional.

5.1 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 del mismo.

5.1.1 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.

5.1.1.1 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.

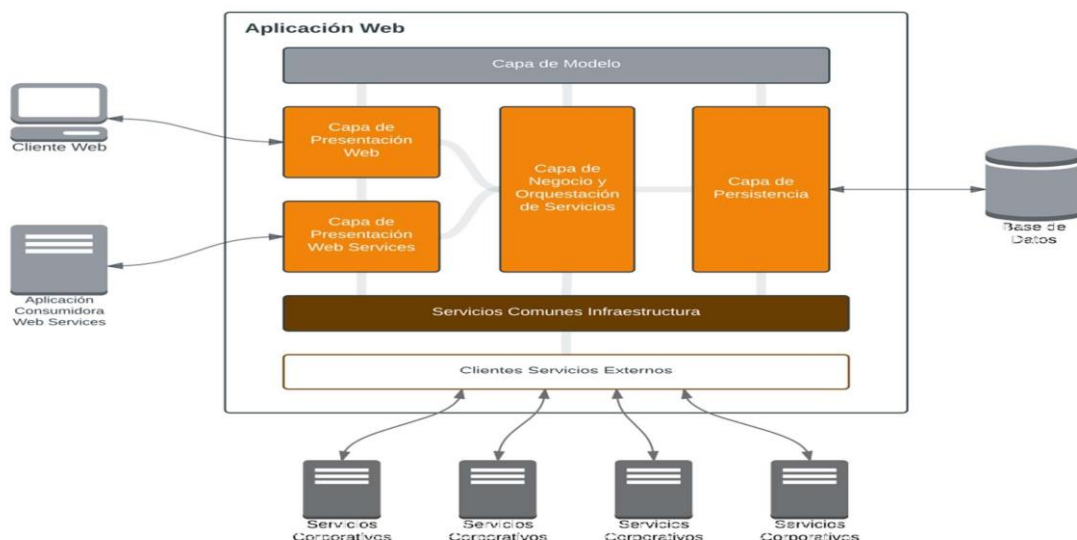


Figura 1. Vista lógica de la arquitectura de referencia

5.1.1.2 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 cumplen las soluciones desarrolladas para Osakidetza:

5.1.1.3 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.

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** de este.

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.

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.

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 objetorelacional (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 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**.

5.2 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 Architecture), arquitectura de aplicaciones y servicios en malla.

MASA (Mesh App and Service Architecture) 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.

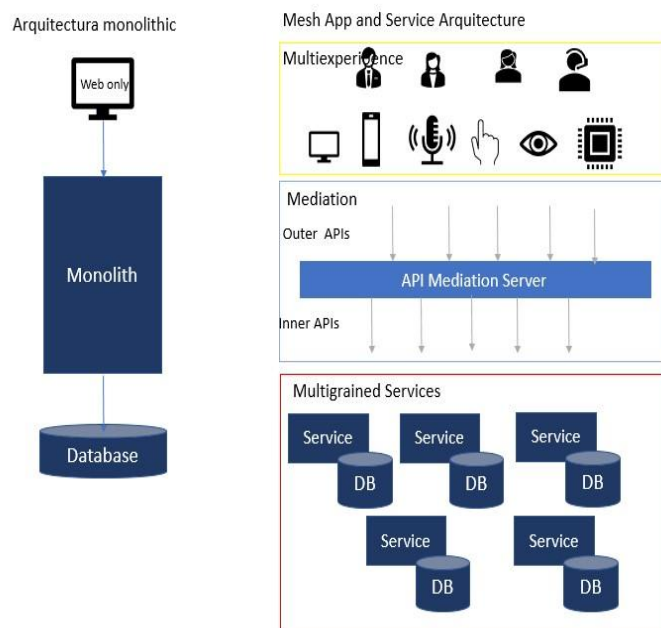
Gestión de transformaciones: la mediación de API puede traducir entre diferentes protocolos: Puede transformar mensajes, por lo que admite formatos personalizados para diferentes consumidores y asignaciones entre diferentes versiones de API.

- **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.

Macro-servicios 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)

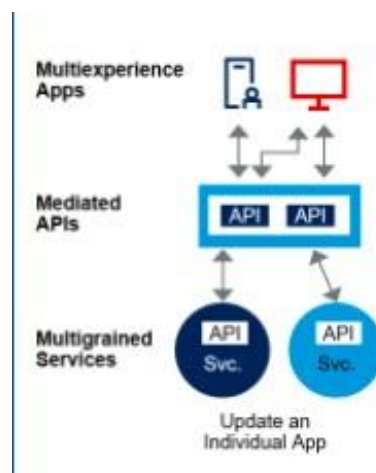
Plataformas de aplicaciones de código bajo (LCAP) y plataformas de desarrollo multiexperiencia (MXDP)

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:

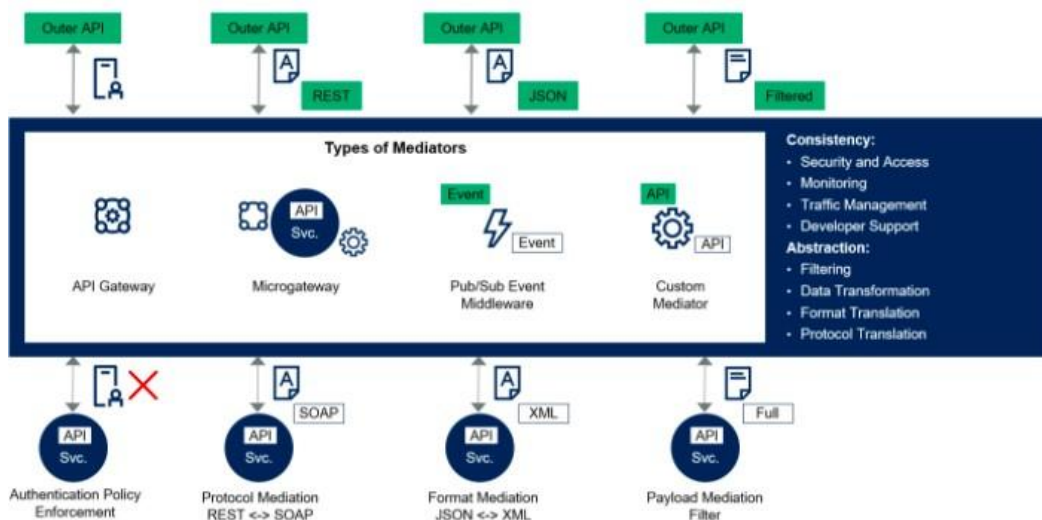
- API internas habilitan el dominio nativo del servicio y los modelos de interacción.
- API externas proporcionan una capa uniforme para que las aplicaciones accedan a la funcionalidad de back-end. Las aplicaciones pueden compartir una API externa o tener una API única. Las aplicaciones usan API externas 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

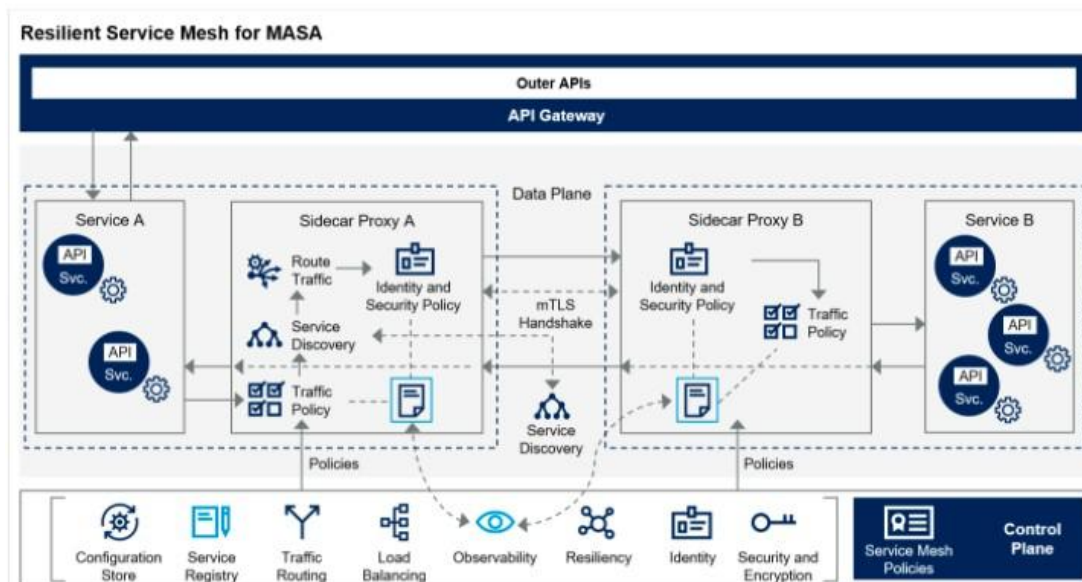
La capa de mediación permite abstraer las complejidades de los servicios de las preocupaciones del cliente de la API externa. La capa de mediación puede usar uno o más mediadores, tecnologías que se ubican entre las API externas que usan las aplicaciones cliente y los servicios. Los contratos técnicos definidos 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

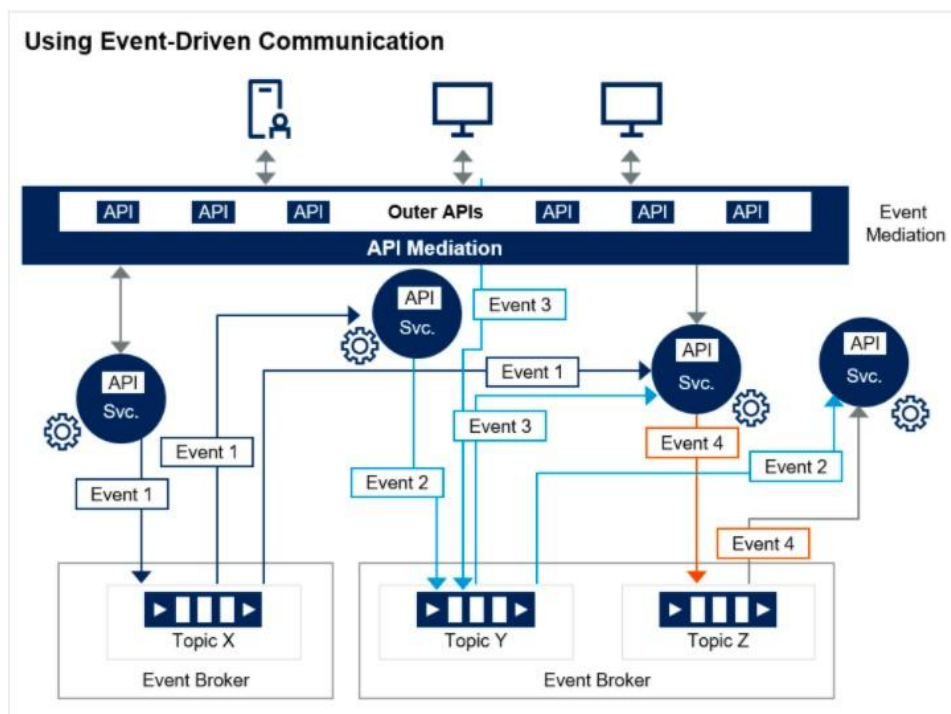
Construir un MASA con servicios multigrano requiere capacidades para habilitar y gobernar las interacciones entre servicios. Una red de servicios (service mesh) es una combinación de componentes de infraestructura, arquitectura y configuración, como puertas de enlace internas o proxies y mediación ligera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

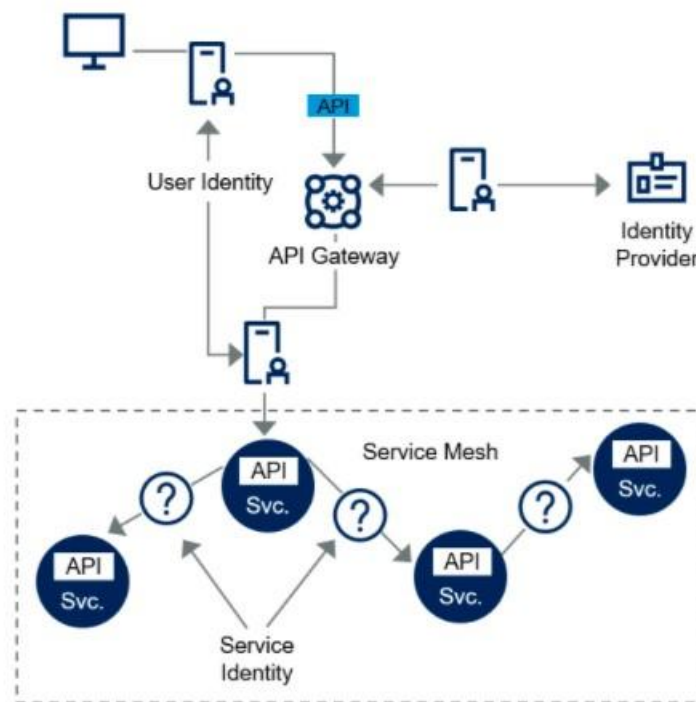
Un modelo de comunicación basado en eventos proporciona un mayor grado de desacoplamiento que un modelo de comunicación tradicional de solicitud/respuesta. Los productores de eventos no tienen conocimiento directo de los consumidores intermedios y no dependen del procesamiento de eventos: simplemente 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.

Identity and Access for Service-to-Service Communication



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

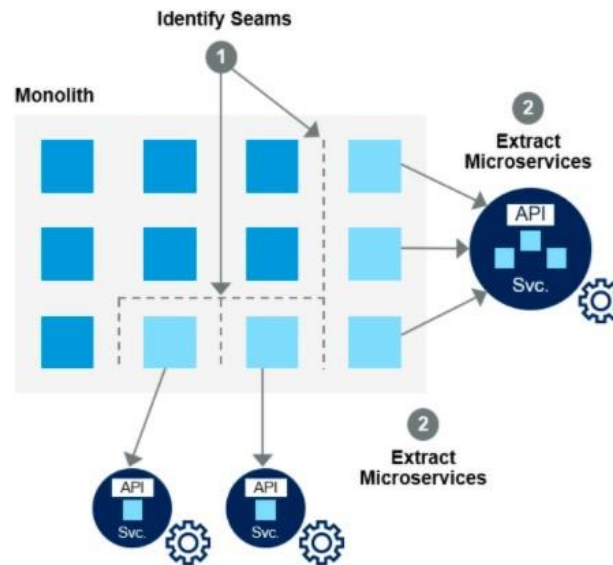
personalizados coordinan las solicitudes a múltiples sistemas de back-end para proporcionar un acceso unificado a los datos y el procesamiento de esas aplicaciones. Esto reduce la cantidad de solicitudes que las aplicaciones deben enviar al back-end. Un uso común de la orquestación son los servicios 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 abstraer 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.

Refactoring Existing Monolithic Application Into Multigrained Services



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.

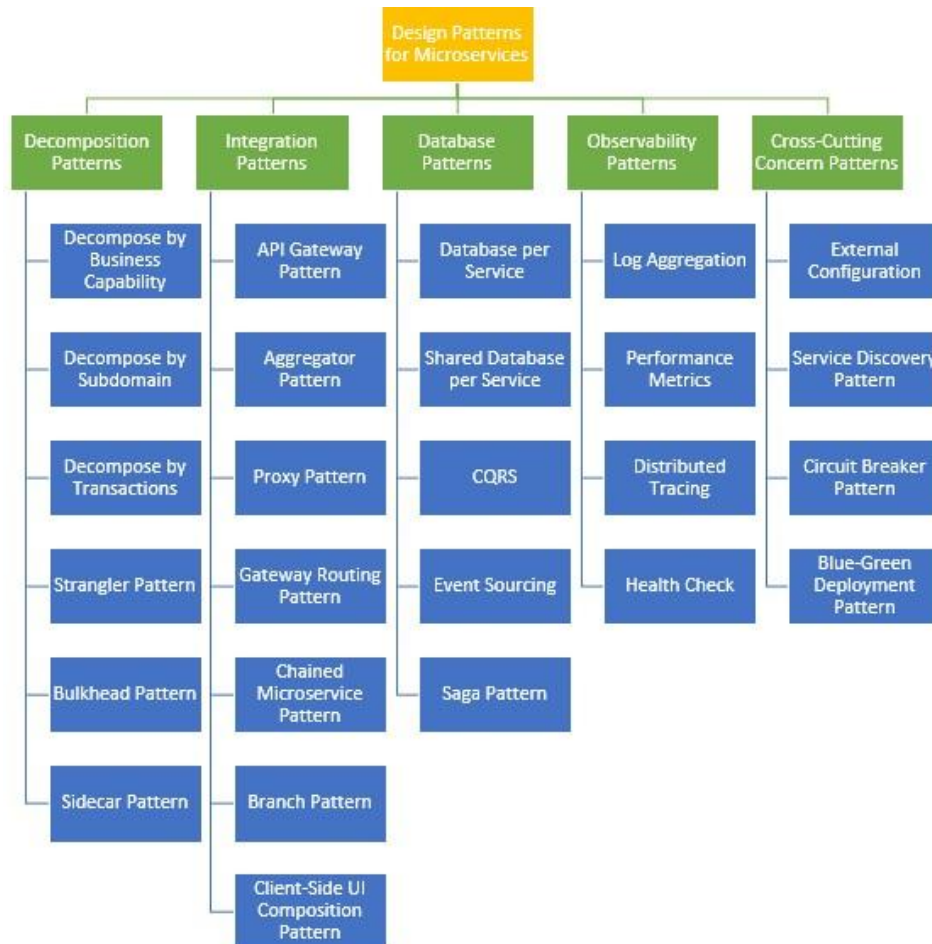
Con esta renovación se persiguen los siguientes objetivos:

- 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.

5.2.1 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://learningsdesign.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.

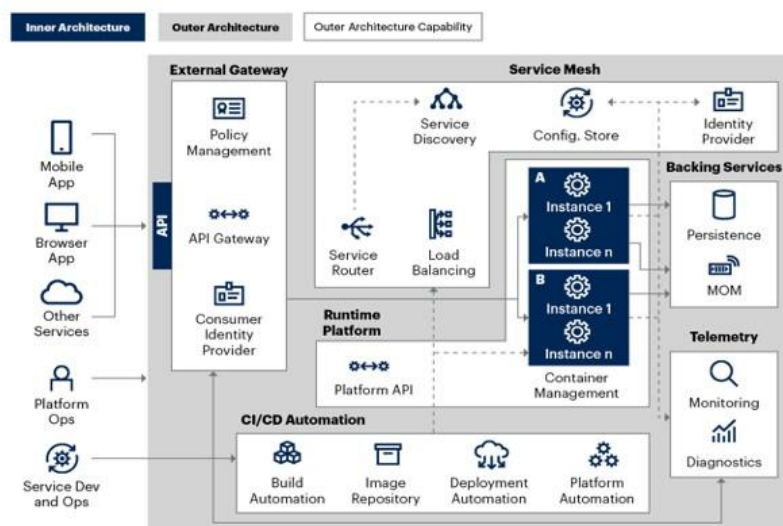
Proporciona mediación de proxy y/o liviana para comunicaciones de servicio a servicio y admite funciones como autenticación, autorización, encriptación, descubrimiento de servicios, enrutamiento de solicitudes, equilibrio de carga e instrumentación de servicios.

➤ Sidecar proxies

- O API gateways
- O OAuth 2.0
- O Microservices orchestration

A Microservice-Based Application

Outer Architecture Components for MSA



5.2.2 Entorno tecnológico

Existen 2 estrategias a nivel de utilización de imágenes base:

5.2.2.1 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.

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.

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

5.2.2.2 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).

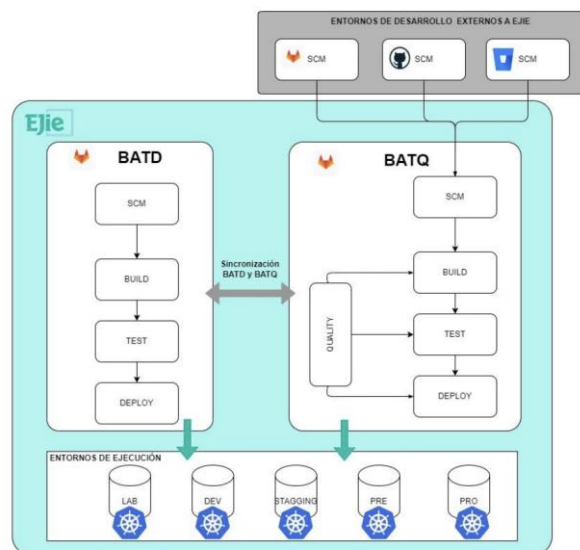
El equipo de desarrollo (DevOps) consume la plataforma **definiendo como código** sondas de operación automática, despliegues por entorno y arquitecturas modernas.

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

5.2.2.3 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:

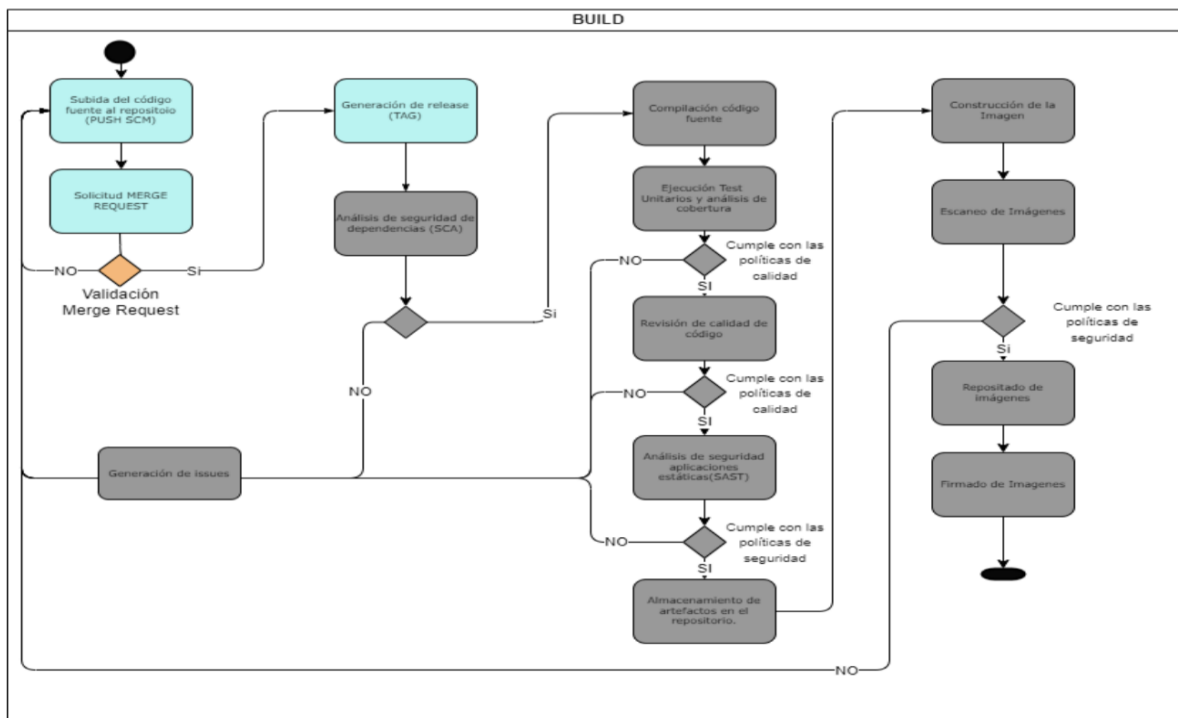
- Básico: Modelo de ciclo de vida cuyo objetivo es disponer de entornos orientados principalmente a los equipos de desarrollo:
 - Basado en un ciclo de vida orientados al CI/CD.
 - Maximizando la automatización de los procesos de integración y despliegue continuos.
 - Incorporando calidad mínimos de 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 runtimes soportados (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.



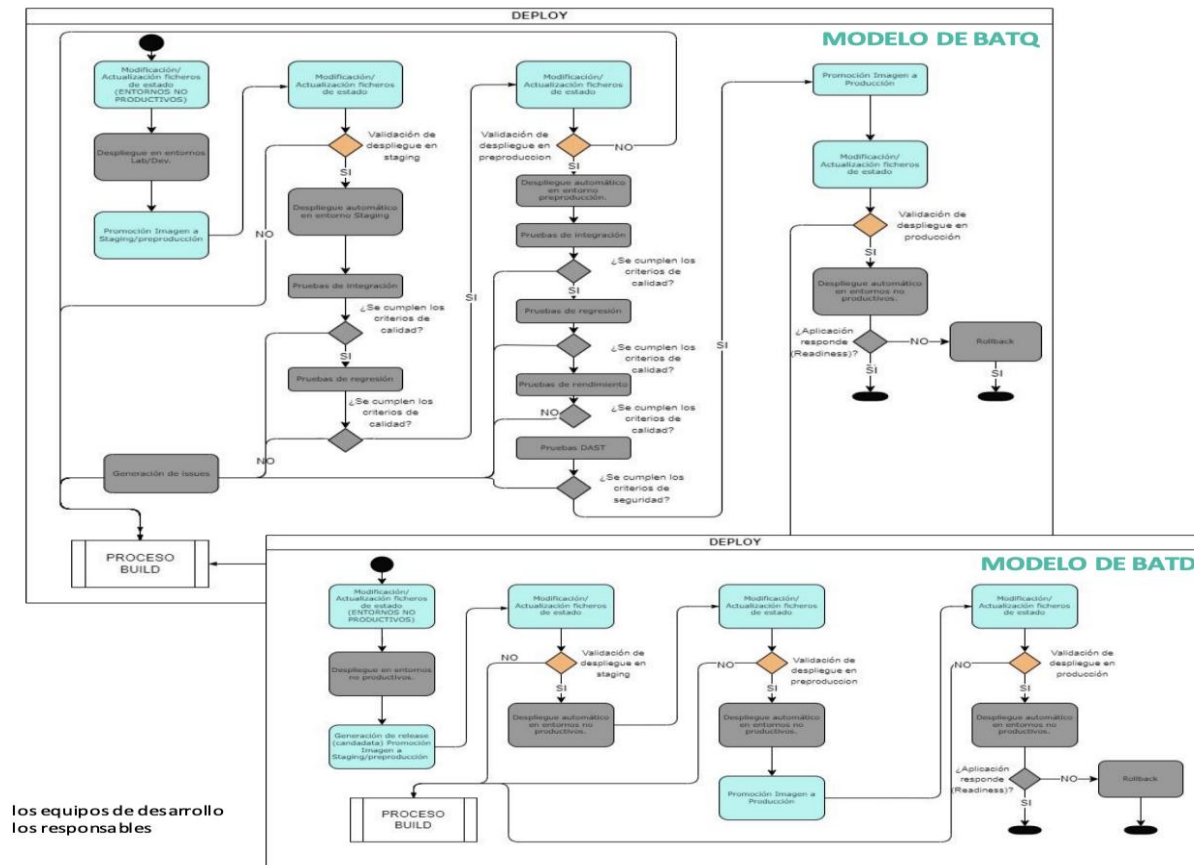
El proceso de DEPLOY parte de las siguientes premisas:

- 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.



5.2.2.4 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
- PostgreSQL 15.6
- MySQL 8.0.30

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

5.2.2.5 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 autenticación:
 - 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 Esquema 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.

Modelos de servicio/entrega

Existen dos tipos de modelos de servicio/entrega:

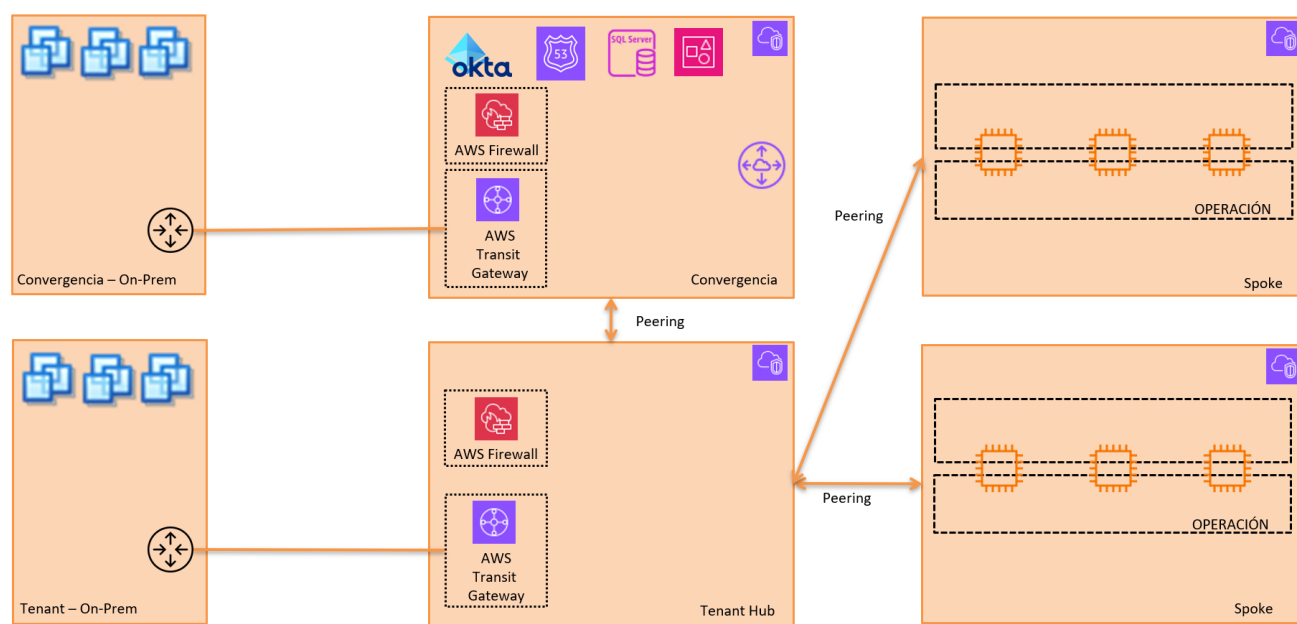
- **Cloud público** que son los servicios prestados por los **CSPs** (Hiperescalares) homologados en Batera.
 - Modalidad SaaS,
 - Modalidad PaaS (plataformas del portfolio del hiperescalar) o IaaS
- **Cloud privado** son los servicios prestados por **Cloud Privado Batera**.
 - Modalidad PaaS(hosting)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

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.

5.3 Directrices tecnológicas de Infraestructuras

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.

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).

5.3.1 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 9
 - ☐ Aplicaciones móviles:
 - Node.js, Express, AngularJS, MongoDB sobre Red Hat Enterprise Linux 8
 - ☐ Ionic, Cordova sobre Android, iOS
- 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 9
 - ☐ Puesto de trabajo fijo: Windows 10, Enterprise x64
 - ☐ Puesto de trabajo móvil: Android, iOS
- Servidores Web:
 - ☐ Apache Webserver 2.4.X
 - ☐ MS IIS 10 (S.O. Windows Server 2019 y 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
 - ☐ MongoDB 7.0
 - ☐ PostgreSQL 15.6
 - ☐ Mysql 8.0.30
- (*) Continuidad de negocio y Alta disponibilidad: Oracle RAC, SQL Server Always ON solo 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 5.3
 - 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 a los de 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**.

5.3.2 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>)
 - 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.

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 reports 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 DNle	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

5.3.3 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 interface 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 a 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:

- o 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.
- o Especificaciones de limpieza y resistencia a los golpes.
- o Características de comunicaciones como pueden ser conectividad Wifi y 4G.

5.3.4 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 válidos 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.

5.3.5 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.
- Instalación de agente Dynatrace para aplicaciones web/Contenerizadas/Virtualización de puesto (Horizon)

Las herramientas de monitorización se integran con Evanos/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:

- o Procesos en ejecución
- o Puertos de comunicaciones a la escucha
- o Eventos concretos en registros del sistema
- o 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:

- o Pruebas sintéticas de los sistemas cuyo resultado determine el correcto funcionamiento del sistema o el correspondiente nivel de alerta preventiva
- o Operativas sobre el sistema que puedan ayudar a determinar o especificar la condición concreta de los errores o alertas producidos.
- o Operativas sobre el sistema para intentar remediar automáticamente la
- o 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.

5.3.6 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 del mismo, 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.

5.4 Interoperabilidad

Los requisitos de integración de diversos sistemas de información de Osakidetza se realizarán de acuerdo a 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, enrutar y almacenar los eventos para su envío a los suscriptores 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 (para aplicaciones móviles)
- 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 o 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, SSL

- o Interoperabilidad con registros UDDI v3-compliant
- o Sistemas middleware basados en JMS/MQ
- o HTTPS

5.4.1 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 cubrirá 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:

- o Autenticación mediante certificado x509
- o 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.

- o oracle_wss10_x509_token_with_message_sign_service_policy
- o oracle_wss10_x509_token_with_message_sign_service_policy_net
- o oracle_wss10_x509_token_with_message_sign_client_policy
- o oracle_wss10_x509_token_with_message_sign_client_policy_net
- o 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.

- o Es responsabilidad de cada aplicación publicada en la DMZ controlar el acceso y autorizar a los usuarios.
- o El OSB de internet publica los servicios a los que pueden acceder las aplicaciones de internet.
- o El OSB de internet audita todas las llamadas a los Servicios Web mediante una política propietaria de Osakidetza gestionada por OWSM.

- o 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.
- o El OSB de internet delega la ejecución a servicios publicados en la Intranet.
- o En la intranet, se despliegan instancias independientes de servicios web para dar servicio a las peticiones que llegan desde el OSB de Internet.
- o Entre el OSB de intranet y el OSB de intranet, se realiza una federación de servicios con las condiciones de seguridad habituales.

Nota: OSB -> Oracle Service Bus 12c

5.4.2 Gestor de Eventos

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

Estándares de comunicación

La mensajería del Servicio Osakidetza se implementará de acuerdo con las especificaciones del estándar HL7 versión 2.XX 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 transaccionabilidad 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 JMS	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	Ninguna,
			garantiza la entrega en el mismo orden que ha recibido los mensajes	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.

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 los mismos.

Anexo 1. Asistencia Técnica Calidad Osakidetza

Alcance

Osakidetza dispone de una base instalada de aplicaciones integrada tanto por desarrollos a medida como por productos de mercado, sobre los cuales da soporte a su actividad. El mantenimiento y evolución de estas aplicaciones y sistemas de información es imprescindible para asegurar la continuidad de las funciones que proveen.

Actualmente, la creciente demanda en materia de sistemas de información hace necesario replantearse los modelos de atención al ciudadano y el servicio prestado tal como los conocemos en la actualidad, con el objeto de avanzar hacia nuevos canales de comunicación con los pacientes, definiendo los procesos integrados entre los distintos niveles de atención sanitaria, y suministrar plataformas y herramientas para dar soporte a las decisiones clínicas.

Este anexo se divide en las 2 grandes áreas funcionales que existen en Osakidetza (Asistencial y Gestión). Únicamente sirve como referencia para los servicios de Asistencia Técnica que se demandan y por lo tanto cualquier otra aplicación proveniente de las nuevas demandas en Osakidetza podrá ser alcance del servicio.

Área Asistencial

ID.	Sistema de Información	Área	Tecnología	Arquitectura	Criticidad	Descripción
B40	Osabide Global	Asistencial-Historia Clínica	.NET+ORACLE	Mixta	24x7	Estación Clínica (Primaria y Especializada).

N06	Osabide-AP	Asistencial-Primaria	.NET+ORACLE	Centralizada	24X7	El sistema de información de Osabide-AP cubre, con carácter integral, todas las necesidades administrativas en el ámbito de la Atención Primaria. El producto incluye fundamentalmente las siguientes funcionalidades: • Base de datos de pacientes de la CAPV en sincronía con la TIS de Sanidad. Encaje en la estructura organizativa de los cupos-pacientes. • Base de datos de recursos, medios y profesionales de Osakidetza, debidamente estructurados en jerarquía: • Confección de Agendas (calendarios, patrón, programa, actos,...). • Base de datos de catálogos y maestros corporativos (codificaciones, diagnósticos, Vademécum, Callejero, catálogos codificados,...). • Proceso de citación a pacientes. • Bajas o ITs. • Planificación de actividades de salud programadas • Control de usuarios y accesos.
A53	e-osabide	Asistencial-Especializada	J2EE+ORACLE	Centralizada	24x7	HIS hospitalario.
J53	OSATEGI	Asistencial	HTML5	centralizada	24x7	Gestión de Pacientes, integrada con Aseguramiento (SIABP).
N18	Detección Precoz Cáncer de Mama	Asistencial	.NET+ORACLE	Centralizada	8x5	Cribado de Cáncer de Mama.

B39 B55 B56	Servicios Universales de prescripción (Presbide)	Asistencial-Farmacia	J2EE+ORACLE	Centralizada	24X7	Sistema de información para la prescripción asistida (Receta), integrado con el sistema eRezeta del Departamento de Salud. B39-SUPRE B55-SUPRE EJIE (réplica Osakidetza-->EJIE) B56- EJIE (Réplica EJIE-->Osakidetza)
N11	Detección Precoz Cáncer Colorrectal	Asistencial	.NET+ORACLE	Centralizada	8x5	Cribado de Cáncer de Colon.

Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

J07	OsabideINTEGRA	Asistencial-Historia Clínica	HTML5+ORACLE	Centralizada	24X7	Estación clínica ligera, para movilidad y acceso entes externos.
P35	Quejas y reclamaciones (SAPU)	Asistencial	HTML5+ORACLE	Centralizada	8x5	Gestión del Servicio de Atención a Pacientes y Usuarios (SAPU).
SIB	SIB: Sistema de Informado Berria	Asistencial	.NET+ORACLE	Mixta	24X7	Herramienta para el informado de cualquier tipo de asistencia integrado con el HIS eOsabide, y con la solución de reconocimiento del habla SpeechMagic de Nuance.
B95	INOZ berria	Asistencial	HTML5	Centralizada	8x5	Gestión de la Infección Nosocomial
B61	Sistemas Externos	Transversal	.NET+ORACLE	Centralizada	24X7	Repositorio Documental, incluye Digitalización online y Subida informe vía impresora virtual
C24	AVICENA	Asistencial	HTML5+ORACLE	Centralizada	24x7	Generación automática de volantes electrónicos WEB/APP. (Autotoma de muestra y autoregistro)
B23	PADI: Salud Bucodental Infantil	Asistencial	J2EE+ORACLE	Centralizada	8x5	Aplicación que gestiona los procesos Administrativos, Historiales Dentales, Facturaciones, Perfiles Asistenciales. Tiene como objeto la comunicación entre Clientes, Usuarios y Administración.
B66	HCDSNS: Historia Clínica Digital SNS	Asistencial-Historia Clínica	J2EE+Oracle	Centralizada	24X7	Interoperabilidad con el SNS
A29	Tumores. Registro hospitalario	Asistencial	J2EE+ORACLE	Centralizada	8x5	Aplicación destinada a recoger y mantener el registro Hospitalario del Cáncer, además de posibilitar la extracción y explotación de los datos contenidos, con el objeto de dar soporte a los diferentes estudios que se llevan a cabo sobre esta enfermedad.
B21	e-osabide. Partos/Bebés	Asistencial-Especializada	J2EE Rich+ORACLE	Centralizada	24X7	Registro de los datos relativos a las pacientes que ingresan en el Hospital con un episodio de parto. recoge principalmente la siguiente información: Datos de la Embarazada, Datos del Parto, Datos de los Recién Nacidos e incluye el proceso de Extracción de datos para el Registro de Metabolopatías.
B49	e-osabide. Mto. Catálogos	Transversal	J2EE Rich+ORACLE	Mixta	8x5	Aplicación para el mantenimiento de catálogos corporativos.

B69	OSABIRT	Transversal	J2EE+ORACLE	Centralizada	24x7	Generador Informes de impresión
C14	BDCAP	Asistencial-Primaria	.NET+ORACLE	Centralizada	8x5	BDCAP - Base de Datos Clínicos de Atención Primaria para el Ministerio
I07	Propagación Fusiones	Transversal	J2EE+ORACLE+.NET	Mixta	24X7	Software de Propagación de Fusiones
J06	Trazabilidad LOPD	Transversal	HTML5+ORACLE	Centralizada	24X7	Trazabilidad LOPD: Aplicación de registro y consulta de traza de propósito general
J08	OSABAR	Asistencial	HTML5	centralizada	8x5	Barra de utilidades en euskera (OG y e-osabide)
J54	Callejero	Transversal	HTML5	centralizada	24x7	Callejero, derivada de J53, integrada con Aseguramiento (SIABP).
J55	Pacientes Berria Metamodelo	Asistencial	HTML5	centralizada	24x7	Metamodelo de J53 , derivada de J53, integrada con Aseguramiento (SIABP).
J60	Petición Historias Clínicas	Asistencial	HTML5	Centralizada	8x5	Gestión de la Petición de Historias Clínicas
J61	Hodei. Plan Contingencias	Asistencial-Historia Clínica	HTML5	Centralizada	24x7	Plan Contingencias HC
N12	Osabide-AP. Integración	Asistencial-Primaria	.NET+ORACLE	Centralizada	24X7	Osabide-AP: Capa de servicios de integración
N16	Esterilización	Asistencial	J2EE+ORACLE	Distribuida	8x5	El objeto de esta aplicación es la Gestión de Peticiones al Proceso de Esterilización de material sanitario de los hospitales de la red
N25	Kodifika	Asistencial	.NET+ORACLE	Centralizada	24X7	Kodifika on-line.

Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

N28	AURIL	Asistencial	.NET+ORACLE			Gestor de almacenamiento de muestras biológicas y tejidos humanos
P30	REKADE	Asistencial	HTML5+ORACLE	centralizada	8x5	Registro de Calidad de Donación en Euskadi
P36	Gestión encuestas	Transversal	LimeSurvey	Centralizada		Herramienta de Gestión de encuestas.
S04	Event Manager	Transversal	J2EE+ORACLE	Centralizada	24x7	Gestor de Eventos.
S05	Auditoria SOA	Transversal	OSB	Centralizada	24X7	Auditoria SOA BackOffice de Carpeta de Salud: Aplicación interna para el registro de la trazabilidad de la HC

SAP

Aplicación SAP	Descripción	Ámbito funcional	Versión SAP	Componente SAP
SAP ECC Gestión Financiera	Gestión financiera	Finanzas	ECC EHP7 6.0	FI
SAP ECC Gestión Financiera	Control de costes	Finanzas	ECC EHP7 6.0	CO
SAP ECC Gestión Financiera	Contabilidad de activos fijos	Finanzas	ECC EHP7 6.0	AA

SAP ECC Gestión Financiera	Tramitación de facturas electrónicas de proveedores	Finanzas	ECC EHP7 6.0	VIM
SAP ECC Gestión de la Cadena de Suministro	Gestión de materiales	Logística	ECC EHP7 6.0	MM
SAP ECC Gestión de Almacenes	Gestión de almacenes	Logística	ECC EHP7 6.0	WM
SAP ECC Gestión de la Cadena de Suministro	Ventas	Logística	ECC EHP5 6.0	SD
SAP ECC Gestión de Mantenimiento	Gestión de mantenimiento preventivo y correctivo	Mantenimiento	ECC EHP7 6.0	PM
SAP ECC Gastos de Viaje	Gestión de gastos de viaje	Gastos de Viaje	ECC EHP7 6.0	FI-TV
Techedge B+ electronicInvoice	Factura electrónica: recepción y emisión	Finanzas	ECC 6.0 EHP7 PI 7.31	Add-on Techedge
Techedge B+ SII	Suministro Inmediato de Información del IVA	Finanzas	ECC 6.0 EHP7 PI 7.31	Add-on Techedge
Izapide (SAP ECC Gestión de expedientes)	Tramitación de expedientes de contratación	Contratación administrativa	ECC EHP7 6.0	RMS + Fiori
SAP ECC Administración y Gestión de Personal	Administración y gestión de personal: actos administrativos, relaciones con terceros (Haciendas Forales, Seguridad Social)	Gestión de profesionales	ECC EHP7 6.0	ES-PS-HR
SAP ECC Nómina	Cálculo de nómina	Gestión de profesionales	ECC EHP7 6.0	PY-ES

Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

SAP ECC Planificación de Turnos	Cartelera: gestión de turnos de trabajo	Gestión de profesionales	ECC EHP7 6.0	PT-SP
SAP ECC Gestión de Tiempos	Cartelera: gestión del cómputo horario y la retribución variable	Gestión de profesionales	ECC EHP7 6.0	PT
SAP ECC Gestión de Organización	Gestión de plantilla y organización	Gestión de profesionales	ECC EHP7 6.0	ES-PS-HR
SAP ECC Normalización Lingüística	Gestión del plan de normalización del uso del Euskera	Euskera	ECC EHP7 6.0	BC-BMT-OM
SAP ECC Vigilancia Salud Laboral	Gestión de la salud laboral de los profesionales	Salud laboral	ECC 6.0 EHP7	EHS
SAP ECC Prevención de Riesgos Laborales	Gestión de la prevención de riesgos laborales	Salud laboral	ECC 6.0 EHP7	EHS-IHS
SAP ECC Formación continuada	Gestión de la formación continuada	Formación	ECC 6.0 EHP7	PE
SAP ECC Selección y Provisión	Gestión de Selección y Provisión de candidatos	Selección y provisión	ECC 6.0 EHP7	ES-PS-HR
SAP ECC Contabilidad RRHH	Contabilidad y gestión de costes de personal	Contabilidad RR.HH.	ECC 6.0 EHP7	PY-XX-DT
SAP BI Presupuestación	Presupuestación general de Osakidetza para el Departamento de Economía y Hacienda del Gobierno Vasco	Finanzas	NW BW 7.4	BI-IP
SAP BI Tesorería	Estimación general de tesorería de Osakidetza para el Departamento de Economía y Hacienda del Gobierno Vasco	Finanzas	NW BW 7.4	BI-IP

Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

SAP BI Reporting Financiero	Herramienta analítica de ayuda a la toma de decisiones (cuadro de mando)	Finanzas	NW BW 7.4	BW
SAP BI Reporting RR.HH.	Herramienta analítica de ayuda a la toma de decisiones (cuadro de mando)	RR.HH.	NW BW 7.4	BW
Osakidetzakoa (Portal del empleado)	Automatización de flujos de información entre el profesional y las Direcciones de Personas	Portal del empleado	NW Portal 7.4	ESS / MSS
Process Integrator	Bus de integración	Transversal	NW PI 7.31	PI

NO SAP

Aplicación	Descripción	Ámbito funcional	Desarrollo propio / Producto	Tecnología
Eginbide	Tramitación de expedientes de contratación	Contratación administrativa	Desarrollo propio	J2EE+Oracle
Bilbide	Gestión de almacenes periféricos de Atención Primaria	Logística	Desarrollo propio	J2EE+Oracle
Eskabide	Peticiones de material	Logística	Desarrollo propio	J2EE
Portal de factura electrónica	Techedge B+ electronicInvoice	Factura electrónica proveedores	Producto	J2EE+Oracle
Proposabide	Propuestas de actuación con impacto económico	Gestión Presupuestaria	Desarrollo propio	J2EE+Oracle

Curriculum Vitae	Registro corporativo de los méritos de los candidatos o empleados	Desarrollo y Formación RR.HH.	Desarrollo propio	J2EE+Oracle
Desarrollo Profesional	Gestión de las convocatorias de Carrera Profesional	Desarrollo y Formación RR.HH.	Desarrollo propio	J2EE+Oracle
Euskera	Solicitud de cursos de euskera y perfiles lingüísticos	Desarrollo y Formación RR.HH.	Desarrollo propio	J2EE+Oracle
Listas de contratación 2004	Proceso selectivo de contratación temporal de personal (convocatoria 2004)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Concurso de Traslados 2012	Proceso selectivo de Concurso de Traslados (convocatoria 2012)	Selección y provisión	Desarrollo propio	J2EE+Oracle
OPE 2006	Proceso selectivo de Oferta Pública de Empleo (convocatoria 2006)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Concurso de Traslados 2009	Proceso selectivo de Concurso de Traslados (convocatoria 2009)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Listas de contratación 2008	Proceso selectivo de contratación temporal (convocatoria 2014)	Selección y provisión	Desarrollo propio	J2EE+Oracle
OPE 2008	Proceso selectivo de Oferta Pública de Empleo (convocatoria 2008)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Listas de contratación 2011	Proceso selectivo de contratación temporal (convocatoria 2011)	Selección y provisión	Desarrollo propio	J2EE+Oracle
OPE 2011	Proceso selectivo de Oferta Pública de Empleo (convocatoria 2011)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Listas de contratación 2014	Proceso selectivo de contratación temporal (convocatoria 2014)	Selección y provisión	Desarrollo propio	J2EE+Oracle
Concurso de Traslados 2016	Proceso selectivo de Concurso de Traslados (convocatoria 2016)	Selección y provisión	Desarrollo propio	J2EE+Oracle

Aplicaciones modernas

ID.	Sistema de Información	Área	Tecnología	Arquitectura	Criticidad	Descripción
M001	ATEA	Asistencial-Especializada	Contenedores/Microservicios	Centralizada	24x7	HIS hospitalario. Reingeniería
M002	Farmacia Hospitalaria	Asistencial-Farmacia	Contenedores/Microservicios	Centralizada	24X7	Aplicación web que permite la prescripción de medicamentos por parte del personal médico, la preparación de los tratamientos por parte de farmacia y la administración a pie de cama por el colectivo de enfermería. Sustituye al modulo B27-Farmacia alojado bajo A53-eOsabide.
M003	STO	Asistencial-Especializada	Contenedores/Microservicios	Centralizada	24x7	Servidor terminologico
M004	GESTPATH	Asistencial-Anatomia Patologica	Contenedores/Microservicios	centralizada	24x7	Sistema de información para Anatomia patologica. Sustituye a B46-VITROPATH
M005	Tuberculosis	Asistencial	Contenedores/Microservicios	Centralizada	8x5	Nuevo sistema de información para la gestión de la Tuberculosis.
M006	ESKUTIK	Asistencial	Contenedores/Microservicios	Centralizada	24x7	Aplicación para mantener a los familiares de los pacientes informados en tiempo real.
M007	EMBARAZO 360	Asistencial	Contenedores/Microservicios	Centralizada	8x5	Aplicacion para la gestión del embarazo 360
M008	Cambio de contraseña Norbide	Gestión	Contenedores/Microservicios	Centralizada	8x5	Cambio de contraseña Norbide

Contratación de servicios de Asistencia Técnica Calidad Osakidetza

Pliego de condiciones técnicas

M009	GDO Manager	Asistencial - Gestión documental	Contenedores/Microservicios	Centralizada	8x5	Aplicación para la gestión del gestor documental Opentext Documentum
M010	Datubide	Horizontal- LOPD	Contenedores/Microservicios	Centralizada	8x5	Evolución J06- auditoría LOPD modulo onpremise
M011	ERANTZUNAK	Gestión	Contenedores/Microservicios	Centralizada	24x7	Servicio PaaS para implementación de microservicio para integración entre EV y solución SaaS de Salesforce.