Pliego de condiciones técnicas particulares.

Título: Servicios de mantenimiento, nuevas funcionalidades y desarrollo de nuevas aplicaciones web

Número de expediente: IZNP_S_007_2026

Tipo: ☐ Servicios, ☐ Suministro, ☐ Mixto

Izenpe 2026

Este documento es propiedad de Izenpe. Únicamente puede ser reproducido en su totalidad.

Beato Tomás de Zumárraga
71 - 1ª Planta
01008
Vitoria - Gasteiz

www.izenpe.eus
izenpe@izenpe.eus

Documento firmado digitalmente por:
IÑIGO ROBREDO GUINEA (02/02/2026 18:05 CET), ISABEL LUMBRERAS AXPE (02/02/2026 21:10 CET)
IZASKUN URRESTARAZU AMUNDARAIN (04/02/2026 16:12 CET)

Índice

I. Necesidades de contratación....3
II. Entorno tecnológico 4
III. Requisitos del servicio 5
1. Desarrollo de nuevas funcionalidades y mantenimiento de los componentes. 5
2. Resolución de Incidencias. 6
3. Integración y despliegue (CI/CD). 6

I. Necesidades de contratación

Ziurtapen eta Zerbitzu Enpresa -Empresa de Certificación y Servicios, Izenpe S.A. (en adelante Izenpe) dispone de:

- Una serie aplicaciones Web tanto de acceso restringido como de acceso público construidas haciendo uso de los siguientes elementos de infraestructura, productos y protocolos, sobre plataforma LINUX:
    - Elementos infraestructura:
- HTTP Apache
- Weblogic 12 (sobre java 1.8)
- Oracle 12
- MySQL
    - Productos
        - Safelayer KeyOne VA/TSA
        - Safelayer TrustedX (plataforma ZAIN y plataforma GILTZA)
        - Aplicación cliente de firma por protocolo Id@zki
        - Veridas
        - Boidas
        - NISAE
        - Latinia (plataforma SMS)
        - LLeidaNET (plataforma SMS)
        - Jira
        - Bitbucket
        - Nexus
        - Jenkins
        - Sonarqube
    - Estándares y protocolos
        - PKI
        - PKCS#7 / CMS / CAdES
        - XMLDsig / XAdES
        - PDF / PAdES
        - WS-Security
        - S/MIME
        - ITU-T X.509v3
        - IETF OCSP

- IETF TSP
- OASIS DSS
- Protocolos SAML y OAuth 2.0 para desarrollo de aplicaciones cliente de Giltza (TrustedX eIDAS) (plataforma para autenticación, federación de identidades y firma de Izenpe).
- JWT

II. Entorno tecnológico

En este punto se van a describir los principales elementos que componen el entorno tecnológico y los requisitos necesarios:

- Zain: se requiere conocimientos a nivel de usuario, para la verificación y generación de las diferentes firmas realizadas desde los componentes de los servicios de Izenpe.
- Giltza: se requiere conocimientos a nivel a nivel técnico, con conocimientos en la gestión, mantenimiento de los plugins de terceros.
- Java EE v8: lenguaje de programación en su modalidad Empresarial para el desarrollo y mantenimiento de las diferentes aplicaciones. Se requieren conocimientos de Java FX y AWT para el mantenimiento de las aplicaciones de escritorio que se registran por protocolo.
- Eclipse IDE: plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para el desarrollo de aplicaciones.
- .NET: plataforma para desarrollo de aplicaciones creado por Microsoft. Permite realizar aplicaciones de todo tipo, desde programas de escritorio a aplicaciones web o programas de consola. Se requiere conocimientos en esta plataforma para el desarrollo y mantenimiento de de los componentes (librerías dll, aplicaciones de kits de bienvenida de Zain y giltza).
- Microsoft Visual Studio: entorno de desarrollo integrado (IDE) con numerosas características que dan respaldo a muchos aspectos del desarrollo de software. Se requiere conocimientos para el desarrollo y mantenimiento de los componentes de los servicios de Izenpe implementados con .NET.
- Weblogic v12: servidor de aplicaciones Java EE utilizado en los entornos de Izenpe. Se requiere conocimientos de administración y despliegue de aplicaciones web para poder gestionar la publicación de las aplicaciones de los servicios de Izenpe.
- Base de datos Oracle y MySQL. Se requieren conocimientos de administración y gestión de modelado de base de datos relacionales.
- Servidor HTTP Apache. Se requieren conocimientos para la gestión y configuración de instancias seguras y redirección al servidor de aplicaciones Weblogic.
- Jira: se requieren conocimientos para la administración de tareas de un proyecto y el seguimiento de errores e incidencias.
- Bitbucket (GIT-FLOW): se requieren conocimientos para el control de versiones de un proyecto.

- Jenkins: se requieren conocimientos para la administración y gestión en este servidor open source para la integración continua de software.
- Gradle: sistema de automatización de construcción de código de software que construye sobre los conceptos de Apache Ant y Apache Maven e introduce un lenguaje específico del dominio basado en Groovy. Se requieren conocimientos para el desarrollo y mantenimiento de las aplicaciones.
- Maven: herramienta de software para la gestión y construcción de proyectos Java
- OWASP: se requieren conocimientos de herramientas de análisis de seguridad de código.
- SonarQube: plataforma open source de gestión de la calidad estática del software, que permite analizar y medir de forma continua la calidad del código de los proyectos. Se requiere conocimientos en esta herramienta para la generación de informes de calidad de código.
- Postman: herramienta para hacer peticiones a APIs y generar colecciones de peticiones que permite probarlas de una manera rápida y sencilla. Se requieren conocimiento en esta herramienta para la generación de colecciones de pruebas unitarias.
- SoapUI: herramienta para la realización de pruebas a aplicaciones con arquitectura orientada a servicio y transferencia de estado representacional. Soporta múltiples protocolos como SOAP, REST, HTTP, etc. Se requieren de conocimiento en esta herramienta para la generación de colecciones de pruebas unitarias.
- Advanced Installer: herramienta para crear instaladores de Windows. Se requieren conocimientos para la generación de los instaladores .msi y futuras mejoras en los instaladores.

III. Requisitos del servicio

Se desarrollará siguiendo el siguiente plan de trabajo.

1. Desarrollo de nuevas funcionalidades y mantenimiento de los componentes.

El desarrollo de nuevas funcionalidades y mantenimiento de componentes se realizará de la siguiente manera:

1. Izenpe planteará una descripción detallada de la nueva funcionalidad y sus requerimientos específicos, y lo enviará a través de la herramienta JIRA que permite el seguimiento de errores e incidencias. Izenpe proporcionará la herramienta JIRA a la que accederá el licitador.
2. Dependiendo del tipo de funcionalidad Izenpe puede requerir una oferta que incluirá detalle específico tanto en número de horas como en fechas de entrega.
3. Si la funcionalidad requiere de oferta, no se comenzará el trabajo hasta estar aprobada por Izenpe. En caso de no requerir oferta, el desarrollo se podrá realizar de forma inmediata.
4. Como resultado del desarrollo se entregará a Izenpe:
    - Código fuente y archivos necesarios para la instalación y pruebas en los servidores de Izenpe. Será necesario el control de versión del código fuente por cada entrega realizada. De esta forma, todo el código se entregará en los diferentes repositorios de Izenpe

(Bitbucket) mediante GIT (Git-Flow), en una nueva rama con el nombre del código de la tarea del JIRA.

- Informe de las pruebas realizadas incluyendo si fuera necesario pruebas de stress, rendimiento, calidad de código, etc.
- Documentación de implantación, uso, etc.

2. Resolución de Incidencias.

El plan de trabajo para la resolución de incidencias:

1. En caso de detectar una incidencia, Izenpe lo notificará a través de la herramienta JIRA que permite el seguimiento de errores e incidencias. Izenpe proporcionará la herramienta JIRA a la que accederá el licitador.
2. Dependiendo del tipo y gravedad de la incidencia, se puede requerir un informe con tiempo estimado, recursos necesarios y estudio de diferentes alternativas posibles. Este informe será entregado en un plazo inferior a 2 días hábiles.
3. Si no es necesario informe, se procederá a solucionar la incidencia de forma inmediata.
4. Una vez solucionada la incidencia se elaborará un informe con la descripción del problema, su solución, pruebas realizadas, informe de calidad de código y el total de horas dedicadas.
5. Como resultado de la resolución se entregará a Izenpe:
    - Código fuente y archivos necesarios para la instalación y pruebas en los servidores de Izenpe. Será necesario el control de versión del código fuente por cada entrega realizada. De esta forma, todo el código se entregará en los diferentes repositorios de Izenpe (Bitbucket) mediante GIT (Git-Flow), en una nueva rama con el nombre del código de la tarea del JIRA.
    - Informe de las pruebas realizadas incluyendo si fuera necesario pruebas de stress, rendimiento, calidad de código, etc.
    - Documentación de implantación, uso, etc.

Tanto en el mantenimiento evolutivo, como en la resolución de incidencias, si se supera el plazo estimado en la oferta, se contactará con Izenpe para evaluar la situación y decidir las acciones a seguir.

3. Integración y despliegue (CI/CD).

Una vez realizada la entrega en el repositorio el código fuente, Izenpe se encargará de realizar los procesos de integración y despliegue en el entorno de desarrollo (para la realización de las pruebas de aceptación) y producción.

El proceso de integración Izenpe lo realizará de la siguiente manera:

1. Crear una nueva versión en Jira que contenga la(s) tarea(s) asociadas(s) a los requisitos solicitados. La nomenclatura de la versión contendrá la fecha de creación y el código de la(s) tarea(s) que se va a liberar/integrar.

2. Indicar en la(s) tarea(s), asociada(s) a los requisitos, la nueva versión de Jira creada en el punto anterior y cambiar el estado a 'En pruebas'
3. Crear nueva rama release (versión), en el repositorio Bitbucket. La nomenclatura de la versión será la misma que la del nombre de la versión que se ha creado en el primer paso.
4. Integrar la rama master y la(s) rama(s) features asociadas a los requisitos solicitados, en la rama release creada en el paso anterior. En el caso de conflictos que Izenpe no puede resolver, se solicitará al adjudicatario su resolución.

El proceso de despliegue Izenpe lo realizará de la siguiente manera:

1. Cada repositorio de código tiene un directorio <codigo_proyecto>CiConfig que contiene el fichero Jenkins-pipeline para la revisión de la calidad (SonarQube) y seguridad del código (dependencyCheck) y el empaquetado (EAR) que se desplegará en los entornos de desarrollo y producción de Izenpe.
2. Desde la plataforma Jenkins y del proyecto Jenkins asociado al repositorio donde se ha creado la rama release en el proceso de integración, ejecutar el paso de empaquetado y despliegue (seleccionando la rama release) en el Weblogic del entorno de desarrollo.
3. Realizar las pruebas de aceptación de los requisitos solicitados. Si las pruebas son correctas, cambiar el estado de la(s) tarea(s) del JIRA, asociada(s) a los requisitos, a 'Revisión aceptada'. En caso contrario se creará una tarea del tipo incidencia no facturable para solucionar el no cumplimiento del requisito.
4. Desplegar los cambios en producción, para ello realizar el paso 2 seleccionando el entorno de producción para el despliegue.
5. Una vez desplegada la versión en el entorno de producción, integrar la rama release y la(s) rama(s) feature(s).
6. Informar del traspaso a producción, para ello se cambia el estado de la(s) tarea(s) del JIRA, asociada(s) a los requisitos, a 'Cerrada', que desencadena una automatización de envío de email con el detalle del traspaso realizado a producción y el cierre de la versión de Jira asociada a la(s) tarea(s).