



Conector de Trazabilidad

Proyecto: Conector de Trazabilidad

Documento de Arquitectura

Creación:	14/04/2016	Autor:	SP
Revisado:	FD	Aprobado:	
Versión:	1.6	Plantilla:	SS-SOFTFAC-TE-56-ES
Código:	AG-CONTRAZA-DA-ES-01	Página:	1 de 23

Índice de contenido

1	Introducción	3
2	Propósito	3
3	Audiencia	3
4	Antecedentes	3
5	Vistas	5
5.1	Vista de Casos de Uso	5
5.2	Vista Lógica	6
5.3	Vista Física	8
5.3.1	Alta disponibilidad	10
5.4	Interacciones	11
5.4.1	RQF 2-01 – Autenticación	11
5.4.2	RQF 2-02 - Envío de cabezal	11
5.4.3	RQF 2-05 - Envío de trazas pendientes	14
5.4.4	RQF 1-02 - Actualización de una configuración	16
5.4.5	RQF 3-01 - Registro de nuevo servicio web	16
5.4.6	RQF 2-04 - Consulta del estado del sistema	17
5.5	Vista de procesos	18
5.6	Vista de base de datos	20
6	Consideraciones para alta disponibilidad	20
7	Requerimientos de Hardware	21
8	Consideraciones Tecnológicas	22
9	Historial de cambios	23

1 Introducción

Este documento presenta una descripción de la Arquitectura de una aplicación, en adelante llamada “ConTraza”, que permita a otras aplicaciones, en adelante llamadas “los clientes”, invocar los servicios web del Sistema de Trazabilidad de AGESIC sin tener que preocuparse por los detalles inherentes a la invocación de servicios web en la Plataforma de Interoperabilidad (PDI) de la Plataforma de Gobierno Electrónico (PGE).

La simplificación respecto de la invocación de los servicios web del Sistema de Trazabilidad consiste en lo siguiente:

- El conector se encarga de resolver los problemas relativos a la invocación de servicios web en la PDI. En particular se encarga de aplicar los mecanismos de seguridad y la obtención del token STS necesario para la invocación de cada servicio web.
- Gestión de errores durante la invocación de los servicios web, tales como indisponibilidad de los servicios o de la plataforma, y también errores propios de la operativa del Sistema de Trazabilidad. En estos casos el conector se encarga de almacenar y reintentar posteriormente el envío de las trazas pendientes, en el orden adecuado.

2 Propósito

Este documento describe la arquitectura propuesta para la construcción de la aplicación mencionada. A lo largo del desarrollo del proyecto también se actualizará para incluir decisiones de diseño e incluso de implementación si fuese necesario.

3 Audiencia

Este documento está dirigido a los stakeholders del proyecto, arquitectos de software, personal de desarrollo y encargados de la gestión del proyecto.

4 Antecedentes

El Sistema de Trazabilidad de AGESIC es una solución que permite que todos los trámites electrónicos que ofrecen los organismos públicos del Uruguay registren trazas de lo que sucede cuando un ciudadano hace uso de ellos. Este sistema, en la actualidad, consta de dos operaciones: registrar un cabezal y registrar una

línea. Cada instancia de traza se compone de un único cabezal y de una o más líneas; si bien lo deseable es que el cabezal siempre se registre antes que cualquier línea, y que las líneas mantengan el orden relativo entre ellas, no es un requerimiento de la aplicación hacerlo ya que el propio Sistema de Trazabilidad se encarga de desplegarlas en orden cuando sea necesario.

Los servicios web que componen el Sistema de Trazabilidad se encuentran disponibles para los clientes a través de la PDI. Esto implica que para poder invocarlos los clientes deben estar conectados a la RedUy y además deben contar con autorización de AGESIC para cada uno de los servicios web. Además, el hecho de que estos servicios web se encuentren publicados en la PDI implica que debe seguirse el protocolo normal para la invocación de servicios seguros de la plataforma, lo que exige que los clientes sean modificados para interoperar con ella. AGESIC ha detectado que, para muchos organismos que cuentan con aplicaciones, implementar los mecanismos para cumplir con dicho protocolo es demasiado complejo, tanto por desconocimiento como por falta de personal técnico adecuado; al mismo tiempo, AGESIC considera que la trazabilidad de todos los trámites en línea es un objetivo importante, por lo que considera de alta relevancia permitir que dichos organismos no dejen de utilizar el Sistema de Trazabilidad por la imposibilidad de implementar los mecanismos necesarios.

Considerando lo anterior, AGESIC propone la implementación de una solución intermedia que se encargue de implementar todos los mecanismos de seguridad necesarios para invocar los servicios web correspondientes al Sistema de Trazabilidad expuesto en la PDI, ofreciendo hacia los clientes una interfaz simplificada de la misma. Así, los clientes invocarían servicios web equivalentes que no requieran de procedimientos complejos de seguridad, y es esta nueva solución la que se encarga de añadir lo que resulte necesario para luego sí invocar los servicios finales en la PDI, retransmitiendo la respuesta a los respectivos clientes. Esta solución es similar, hasta aquí, al Conector PDI ya provisto por AGESIC, que permite invocar servicios web en general, salvo que esta nueva solución está limitada a los servicios web del Sistema de Trazabilidad únicamente. La aplicación ConTraza (Conector de Trazabilidad) se instalaría en el entorno del organismo cliente (de forma que la comunicación entre las aplicaciones clientes y ConTraza sea realizada dentro de la red local, y por tanto con menores requerimientos de seguridad), debiendo estar conectado a RedUy para poder alcanzar los servicios publicados en la PDI (de esta forma no es necesario que las aplicaciones clientes estén también conectados a la RedUy).

Adicionalmente a la necesidad de encargarse de realizar las tareas necesarias para invocar servicios web seguros de la PDI, la aplicación ConTraza debe encargarse también de lidiar con las indisponibilidades del Sistema de Trazabilidad e incluso de la PDI o RedUy. Esto significa que si bien los clientes requieren realizar la traza en tiempo real (cada vez que ocurre un evento relevante en la aplicación cliente ésta necesita registrar la traza en el momento), en el caso de que el Sistema de Trazabilidad no esté disponible o la PDI no funcione correctamente la aplicación ConTraza debe almacenar las trazas que no pudieron

efectuarse para reintentar su envío posteriormente, hasta lograr cumplir con su objetivo, o alcanzar un número de intentos máximos.

Considerando las dos grandes funcionalidades del sistema, los clientes se ven así abstraídos de los pormenores del Sistema de Trazabilidad, siendo para ellos una invocación simple de un servicio web normal. No tienen que preocuparse de las dificultades de la invocación de servicios web publicados en la PDI ni tienen que encargarse de reintentar el envío de trazas cuando el Sistema de Trazabilidad no está disponible.

Además, la aplicación también debe ofrecer dos funcionalidades administrativas fundamentales: permitir registrar nuevas versiones de los servicios web (pudiendo coexistir varias versiones del mismo servicio web en forma simultánea) y permitir visualizar las trazas pendientes de envío para poder tomar las acciones correspondientes: ante el reiterado fracaso en el envío de trazas la aplicación deberá ir incrementando el intervalo que debe transcurrir entre sucesivos intentos hasta llegar a un tiempo máximo, tras lo cual el sistema de reenvío de trazas será detenido completamente (no así la recepción de nuevas trazas), debiendo ser reiniciando en forma manual a través de una interfaz gráfica apropiada.

5 Vistas

En esta sección se presentan las diferentes vistas de la arquitectura de la aplicación. Para su diseño se tomaron en cuenta los casos de uso relevantes.

5.1 Vista de Casos de Uso

En esta sección se describen los casos de uso relevantes para la arquitectura del sistema. A continuación se muestra una tabla con todos los casos de usos del sistema marcando en color amarillo los relevantes para la definición de la arquitectura final de la aplicación. Para cada caso de uso se especifica el nombre y el/los actores que lo realizan.

Los actores son los siguientes:

- **Administrador (A):** es un usuario que puede realizar todas las acciones posibles relacionadas con la administración de la instalación. Es, por ejemplo, el encargado de registrar nuevas versiones de los servicios web, activar o desactivar el mecanismo de reintento de invocación de servicios, etc. Para poder realizar las tareas correspondientes debe iniciar sesión en el sistema proporcionando ciertas credenciales (por ejemplo nombre de usuario y contraseña) las cuales deben ser validadas por la aplicación. Se debe tener en cuenta que si bien puede haber muchos “usuarios

administradores”, no hay diferencia entre ellos en cuanto a las cosas que pueden realizar (no hay otros “roles”).

- **Cliente (C):** es un usuario que puede invocar los servicios web registrados en la instalación, para lo cual no requiere autenticación.
- **Sistema (S):** es un componente que si bien forma parte del sistema (no es un actor externo) tiene autonomía para funcionar por sí mismo (por ejemplo mediante un temporizador) y es capaz de iniciar acciones que no requieren la intervención de una persona.

A continuación se muestra una tabla con los casos de uso identificados, marcando en amarillo aquellos que se consideran más relevantes para definir la arquitectura. La numeración corresponde con la utilizada en el documento AG-CONTRAZA-RG-04-ES - Documento de Requerimientos.

Id	Caso de uso	A	C	S
RQF 1-01	Consulta de configuraciones	*		
RQF 1-02	Actualización de una configuración	*		
RQF 2-01	Autenticación	*		
RQF 2-02	Envío de cabezal		*	
RQF 2-03	Envío de línea		*	
RQF 2-04	Consulta del estado del sistema	*	*	
RQF 2-05	Envío de trazas pendientes			*
RQF 2-06	Consulta de invocaciones	*		
RQF 2-07	Visualización del estado	*		
RQF 2-08	Activar proceso de envío de trazas pendientes	*		
RQF 3-01	Registro de nuevo servicio web	*		
RQF 3-02	Actualización de servicio web	*		
RQF 3-03	Eliminación de servicio web	*		

5.2 Vista Lógica

La siguiente imagen ilustra gráficamente los elementos lógicos que componen el despliegue de la aplicación. Los que están coloreados corresponden a los elementos que serán desarrollados como parte de la solución, mientras que los que están en blanco se asumen existentes.

Nota: los componentes que se desarrollarán como parte de la solución sólo utilizarán librerías OpenSource.

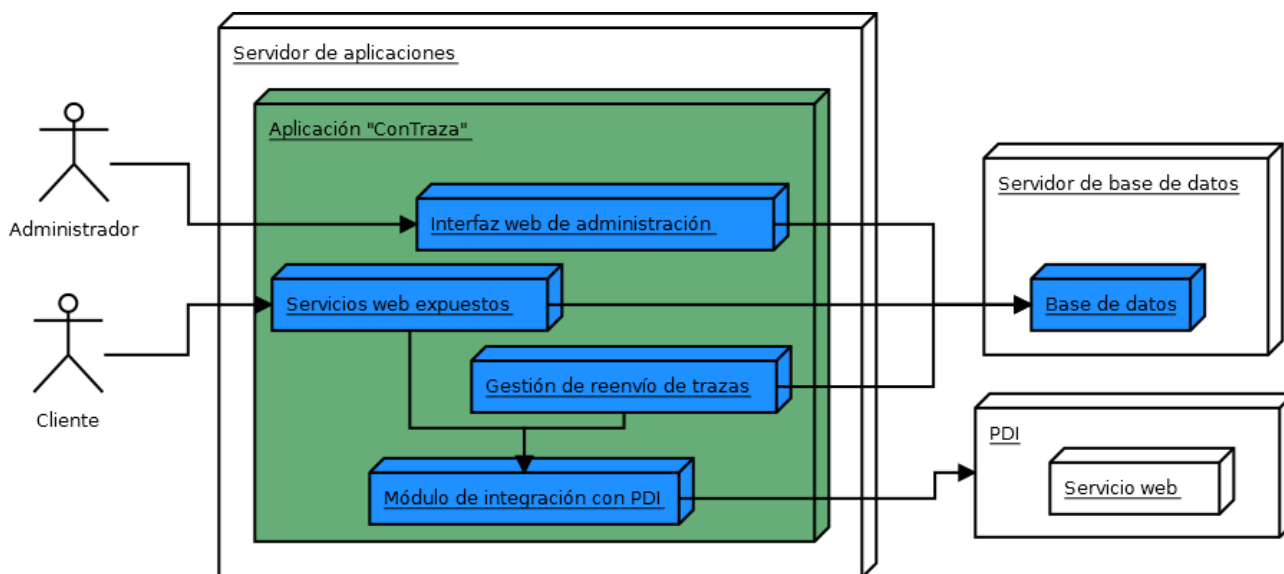


Figura 5.1: Vista lógica con base de datos externa

A continuación se describen los principales elementos que se visualizan en la imagen.

- **Actores.** Básicamente se distinguen dos tipos de actores: los que administran el sistema (deben autenticarse) y los que invocan los servicios web (no se autentican).
- **Servidor de aplicaciones** (o similar). El servidor de aplicaciones es el que da cabida a la aplicación en sí misma. En este componente se instalan tres elementos constituyentes de la aplicación y se configuran los parámetros básicos para el funcionamiento de la misma:
 - **Interfaz de administración:** es el mecanismo por el cual la aplicación permite que un usuario administrador gestione los servicios web disponibles (añadiendo, modificando y eliminando servicios web), así como otros parámetros necesarios, como certificados digitales y datos necesarios para realizar la invocación de los servicios web expuestos en la PDI.
 - **Servicios web expuestos:** es un componente que se encarga de exponer los servicios web que podrán ser invocados por los clientes. Se encarga de recibir las invocaciones, cargar la configuración necesaria y trasladar ambas cosas en conjunto al módulo de integración con PDI. También se encarga de almacenar en la base de datos los datos correspondientes a la invocación, incluyendo los datos enviados y la respuesta recibida, y si la invocación pudo ser realizada correctamente o no.

- **Gestión de reenvío de trazas:** se encarga de intentar reenviar nuevamente las trazas que no pudieron ser enviadas aún.
- **Módulo de integración con PDI:** es un módulo que solo se encarga de invocar servicios web en la PDI, no siendo necesariamente los correspondientes al Sistema de Trazabilidad. Los clientes de este módulo (que son tanto el componente de servicios web expuestos como el de gestión de reenvío de trazas) deben utilizarlo pasando como argumento tanto el mensaje a enviar como toda la información necesaria para hacer la invocación (identificador del servicio web, certificados digitales, y datos requeridos para identificar el servicio web y la operación en la PDI) .
- **PDI (Plataforma de interoperabilidad).** Es donde se encuentran expuestos todos los servicios web de la Plataforma de Gobierno Electrónico, y particularmente los servicios web que forman parte del Servicio de Trazabilidad.
- **Servidor de bases de datos.** En el servidor de bases de datos es donde se almacena toda la información gestionada por la aplicación. Por requerimiento, será MySQL o PostgreSQL.

5.3 Vista Física

La siguiente diagrama muestra el despliegue de componentes a escala física:

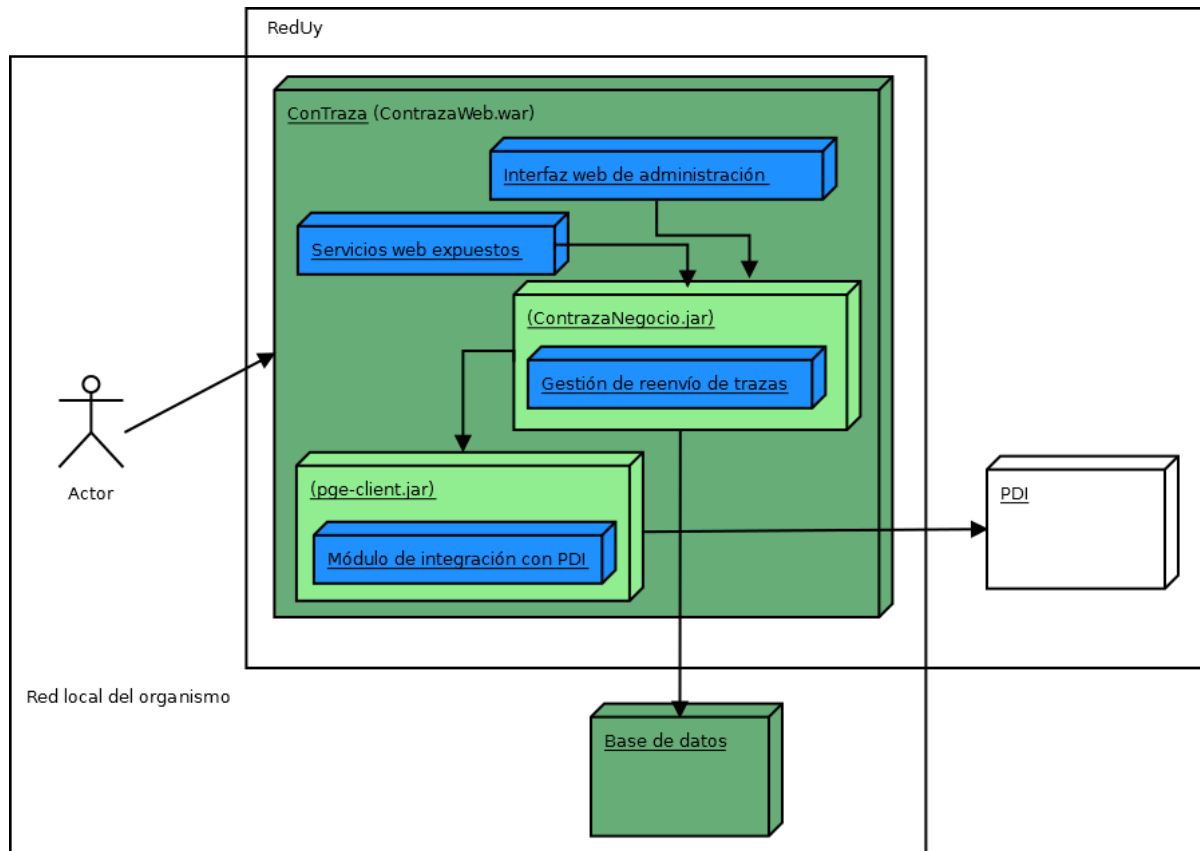


Figura 5.2: Diagrama de despliegue físico

A continuación se describen los principales elementos que se visualizan en la imagen.

- **Actor.** Corresponde a un componente externo a la aplicación que hace uso de ésta a través de alguna de las interfaces expuestas.
- **ConTraza.** Es la aplicación en sí misma, consistente en todas las interfaces que ofrece más los módulos que implementan la lógica operativa y otros componentes externos requeridos para permitir el funcionamiento.
- **PDI.** Representa a la Plataforma de Interoperabilidad, donde estarán todos los servicios web del Sistema de Trazabilidad que deberán ser invocados.
- **Red local del organismo.** Corresponde, como lo dice su nombre, a una red interna al organismo, en donde estarán todos los actores que harán uso de la aplicación (las personas que la administran y los otros sistemas informáticos que la utilizan para invocar servicios web en la PDI). Debe garantizarse que los puntos de acceso a la aplicación solo estén disponibles desde esta red,

particularmente debido a que la aplicación expondrá los servicios web que invocarán los actores solo por HTTP (por requerimiento).

- **RedUy.** Corresponde a la red que interconecta a equipos y redes de múltiples organismos para que interoperen entre sí. En particular, en RedUy se encuentra la PDI, en la cual están expuestos los servicios web del Sistema de Trazabilidad.
- **Base de datos.** Corresponde a la base de datos externa donde se almacenan los datos necesarios para la operativa de la aplicación.

5.3.1 Alta disponibilidad

En el caso de que se instale la solución en un ambiente de alta disponibilidad, el diagrama es el siguiente:

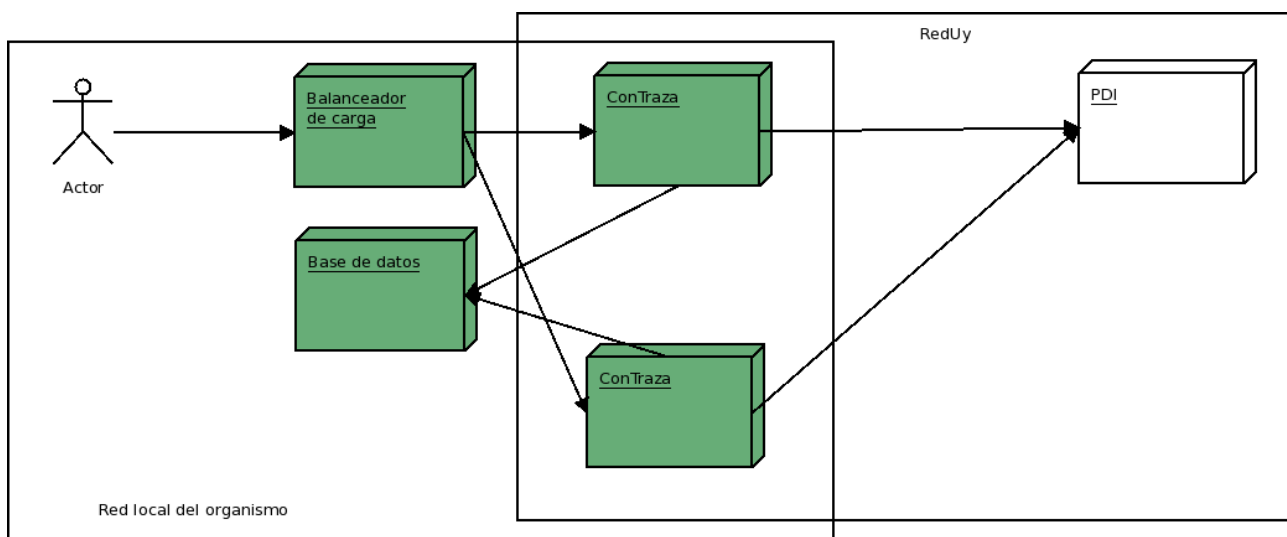


Figura 5.3: Despliegue físico con alta disponibilidad

Los elementos que se muestran son los mismos que se describieron anteriormente, teniendo en cuenta que se tienen varias instancias (en el diagrama solo se muestran dos pero podrían ser más) de la aplicación. Además, se añade lo siguiente:

- **Balanceador de carga.** Corresponde a una aplicación que hará de “proxy” entre el actor y cada una de las instancias de la aplicación, distribuyendo la carga operativa entre todas éstas.

5.4 Interacciones

En esta vista se representa la funcionalidad que el sistema proporciona a los usuarios. Es decir, se representa lo que el sistema hace y las funciones y servicios que ofrece.

5.4.1 RQF 2-01 – Autenticación

Este diagrama describe el proceso general de control de acceso realizado por la aplicación.

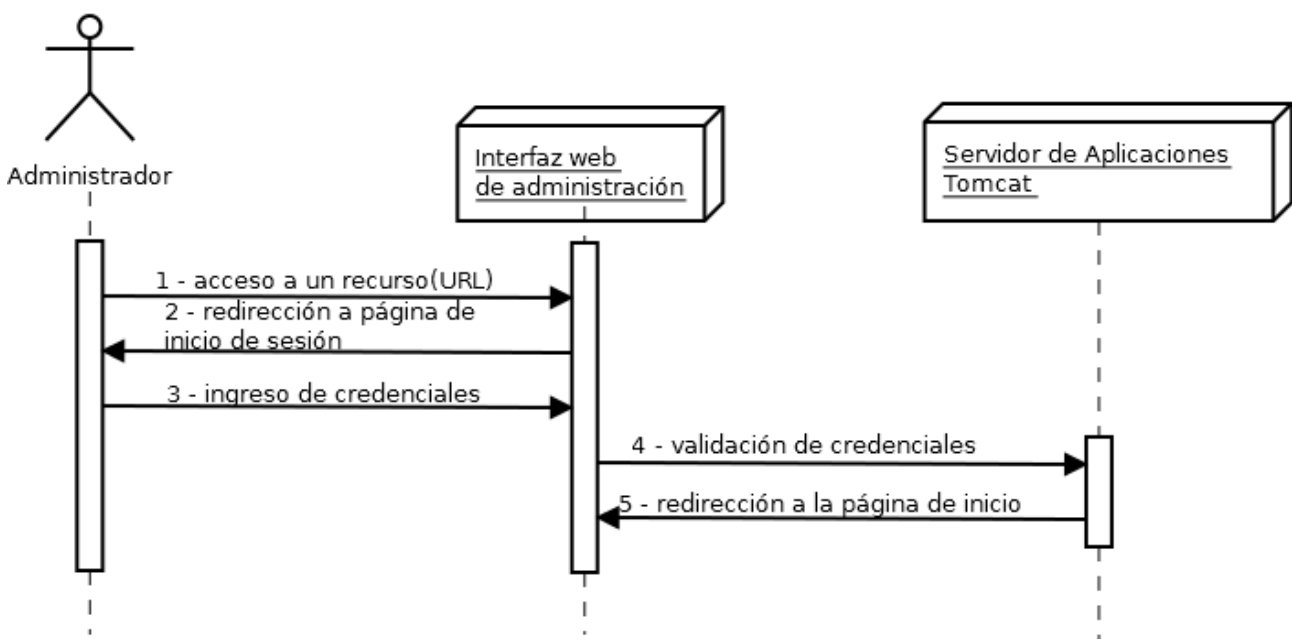


Figura 5.4: Interacciones en el proceso de autenticación

Para acceder a un recurso de la interfaz de administración el usuario debe proporcionar sus credenciales (normalmente código de usuario y contraseña). La verificación de dichas credenciales se hará contra el servidor de base de datos.

El acceso a base de datos se realizará utilizando el framework de persistencia propio de la tecnología Java EE JPA. Por requerimiento no funcional se utilizará Hibernate JPA.

5.4.2 RQF 2-02 - Envío de cabezal

Este diagrama muestra las interacciones que ocurren cuando un cliente invoca el servicio web para registrar un cabezal de la traza.

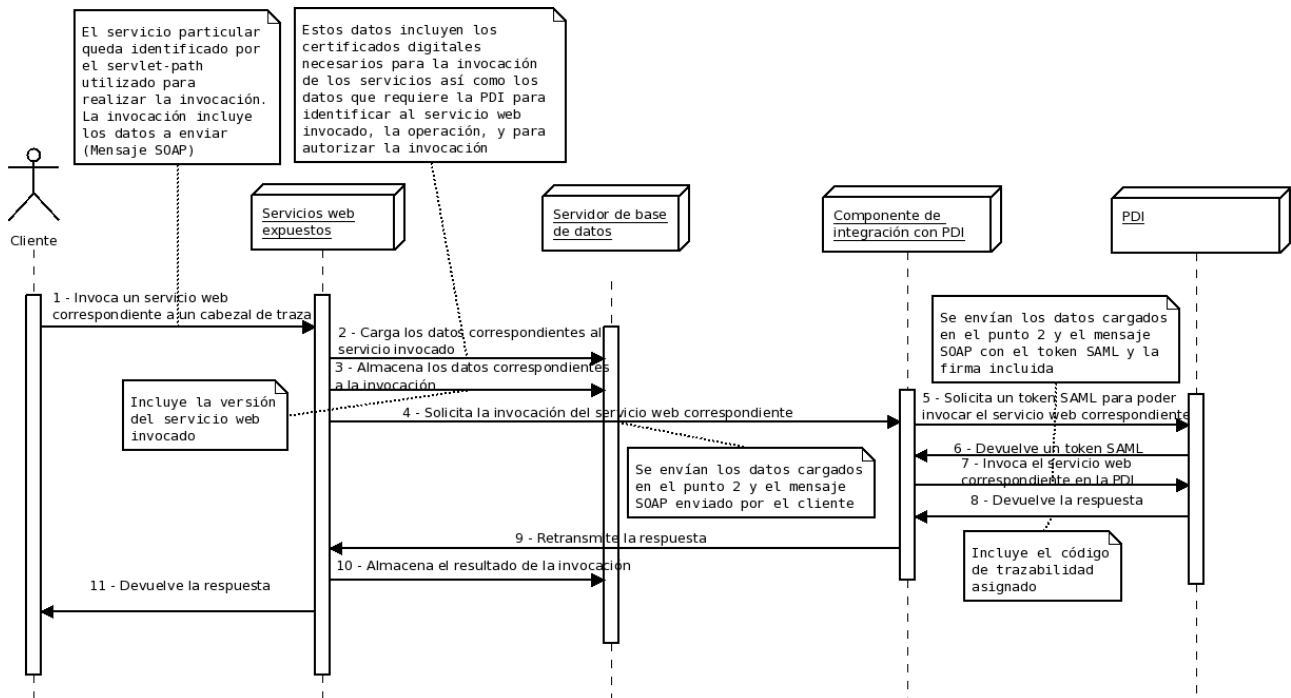


Figura 5.5: Interacciones en el proceso de envío de cabezal

Cuando el cliente solicita la invocación del servicio web para crear un cabezal de trazabilidad se debe determinar cuál servicio es exactamente (podrían haber varias versiones de cabezales), y también se debe determinar todos los datos necesarios para hacer la invocación en la PDI (certificados digitales, URLs lógica y física, rol de autorización, etc). Luego, toda esa información es transferida al módulo de Integración con PDI, el cual se encarga de realizar el proceso normal de solicitud de autorización para invocar un servicio web en la PDI. Si la invocación al servicio web en la PDI concluye satisfactoriamente, la respuesta se devuelve al componente de servicios web expuestos para que a su vez la entregue al cliente inicial. Antes y después de transferir el control al Módulo de Integración con PDI se registra en la base de datos a los efectos de auditoría; además, en el caso de que la invocación realizada por dicho módulo falle, por cualquier razón, el hecho de haber almacenado en la base de datos la información correspondiente permite volver a intentarlo posteriormente.

Flujos alternativos

Una vez transferido el control al Módulo de Integración con PDI, éste se encarga de llevar a cabo todos los pasos necesarios para realizar la invocación del servicio web correspondiente en la PDI. Cualquiera sea el motivo por el cual esta invocación falle (no está disponible la PDI, no hay conexión de red, la PDI no puede validar los datos proporcionados, el Sistema de Trazabilidad no está disponible, etc) el registro correspondiente será marcado como "incompleto" y otro caso de uso se continuará intentando completar la invocación; sin embargo, dado que no hay una respuesta posible para enviar al cliente, se le enviará un

mensaje de error en forma de SOAP Fault. En el caso de que la invocación al servicio web en la PDI devuelva una respuesta en forma de xml y no de tipo SOAPFault, pero que mediante el campo XPathOK del servicio se determine que corresponde a un error, la invocación pasará al estado “error”. En caso de que la invocación al servicio web en la PDI devuelva una respuesta de tipo SOAPFault, se determinará, mediante el campo Xpath SOAPFAULT, el estado final de la invocación. Si el faultfactor coincide con lo incluido en este xpath la invocación quedará como “pendiente”, y sino pasará al estado “error”.

Nota – registro de línea

El envío de una línea es muy similar al descrito en este caso de uso excepto que la respuesta no incluye un código de trazabilidad. La vinculación entre una traza de línea y su correspondiente cabezal se hará mediante el identificador de la transacción.

5.4.3 RQF 2-05 - Envío de trazas pendientes

Este diagrama de secuencia muestra las interacciones que ocurren para el envío de trazas pendientes.

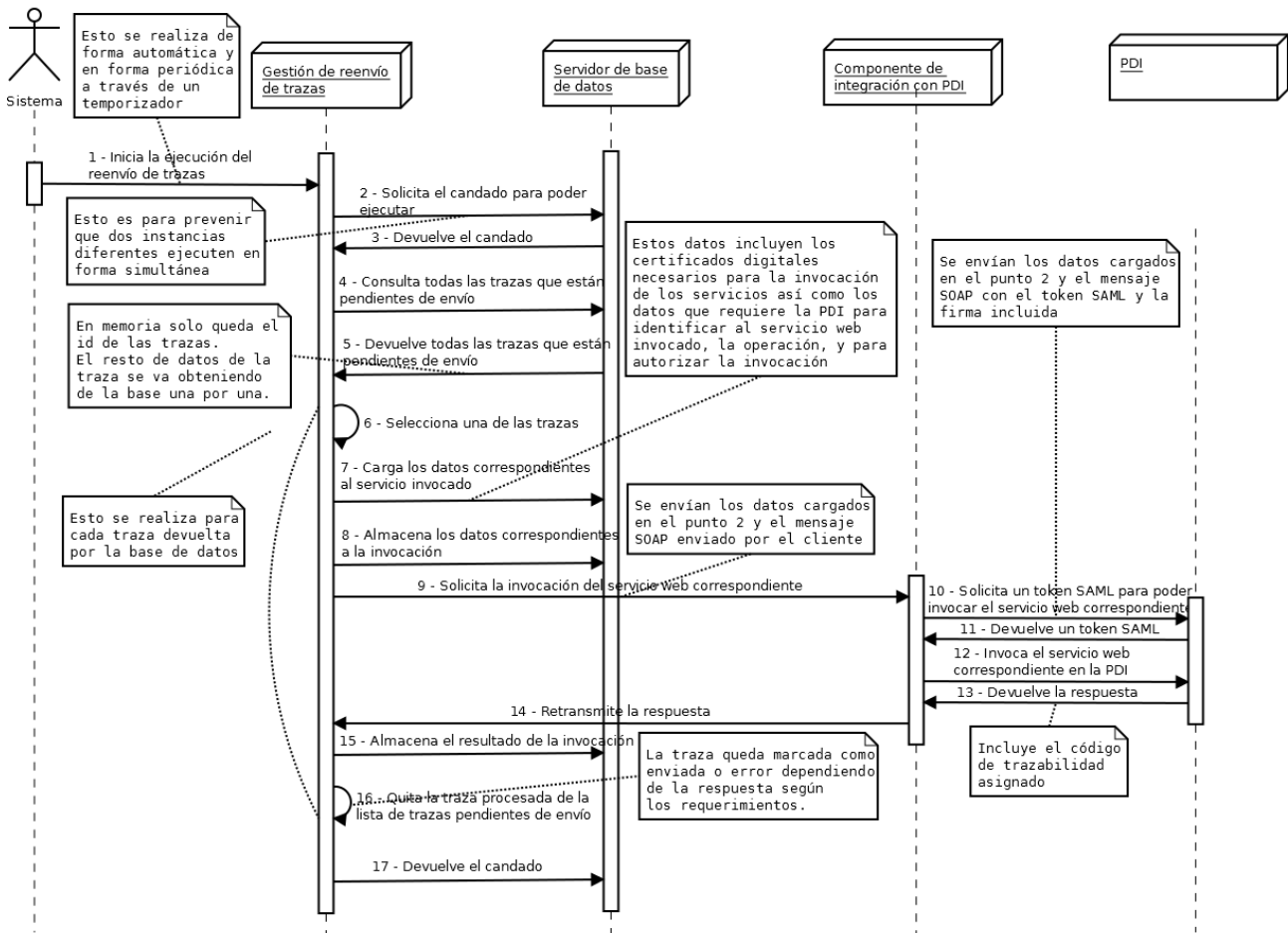


Figura 5.6: Interacciones en el proceso de envío de trazas pendientes

Este caso de uso es muy similar al registro de un cabezal por parte de un cliente, salvo que en lugar de ser iniciado por un cliente es iniciado automáticamente por un temporizador (se ejecuta periódicamente) y abarca a todas las trazas que están pendientes de envío.

Nota: por requerimiento, el temporizador utilizado debe ser Quartz Job Scheduler.

Flujos alternativos

En el punto 2, si al solicitar el candado la base de datos lo rechaza (porque hay otra instancia del proceso

en ejecución) finaliza la ejecución el caso de uso.

Al igual que en el caso del registro del cabezal, si la invocación del servicio web en la PDI falla por cualquier motivo bloqueante (en los puntos 6 a 16) se salta directamente al punto 17. A continuación flujo alternativo para este caso.

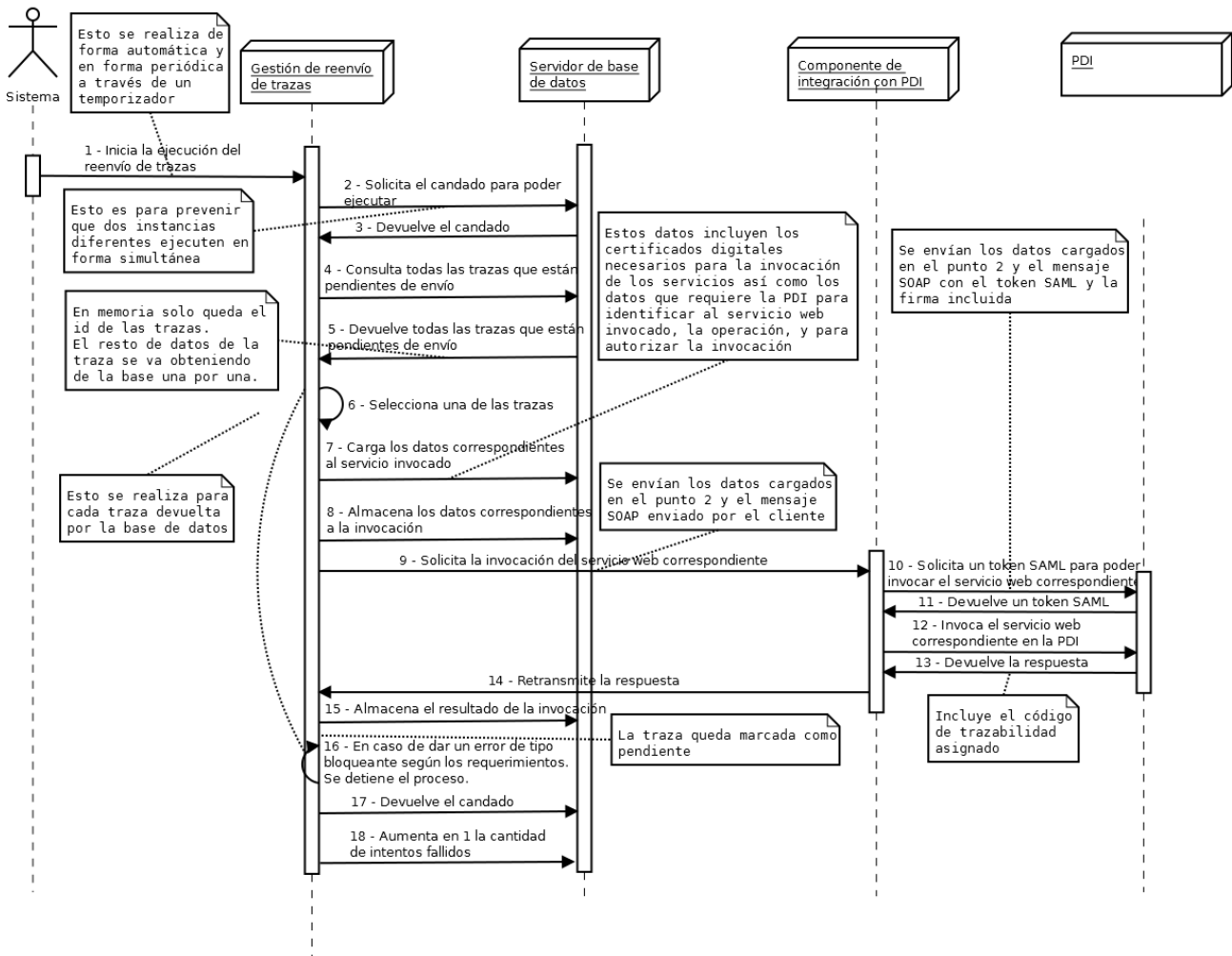


Figura 5.7: Interacciones en el proceso de envío de trazas pendientes - alternativo

Nota: el tiempo transcurrido entre sucesivas ejecuciones del reenvío de trazas es configurable y además variable: mientras el funcionamiento sea normal, la ejecución es periódica; cada vez que ocurra un error en el envío de trazas el tiempo que debe transcurrir para volver a intentar se incrementa, hasta alcanzar un límite, en cuyo caso el proceso de reenvío de trazas se detiene. Luego debe habilitarse manualmente. Al lograr el envío exitoso de una traza el mecanismo vuelve a su funcionamiento normal, con el tiempo mínimo de espera entre ejecuciones.

5.4.4 RQF 1-02 - Actualización de una configuración

Este diagrama de secuencia muestra la interacción entre un administrador y la aplicación para modificar los parámetros de configuración que rigen el funcionamiento de la misma.

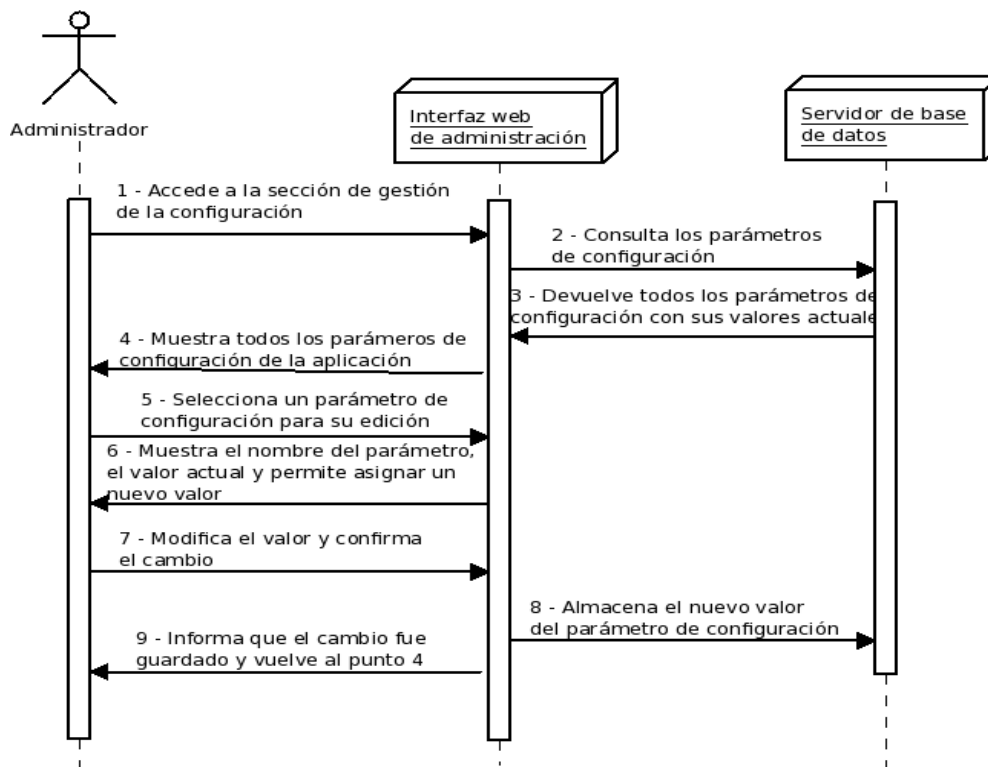


Figura 5.8: Interacciones en el proceso de actualización de una configuración

El caso de uso es autodescriptivo. Se debe considerar que el flujo normal continúa incluso si en el punto 7 el administrador no hace ningún cambio en el valor pero igual confirma la acción.

Flujos alternativos

En el punto 7, si el administrador no confirma el cambio, se vuelve al punto 4.

5.4.5 RQF 3-01 - Registro de nuevo servicio web

Este diagrama de secuencia muestra la interacción entre el administrador y el sistema para registrar un nuevo servicio web.

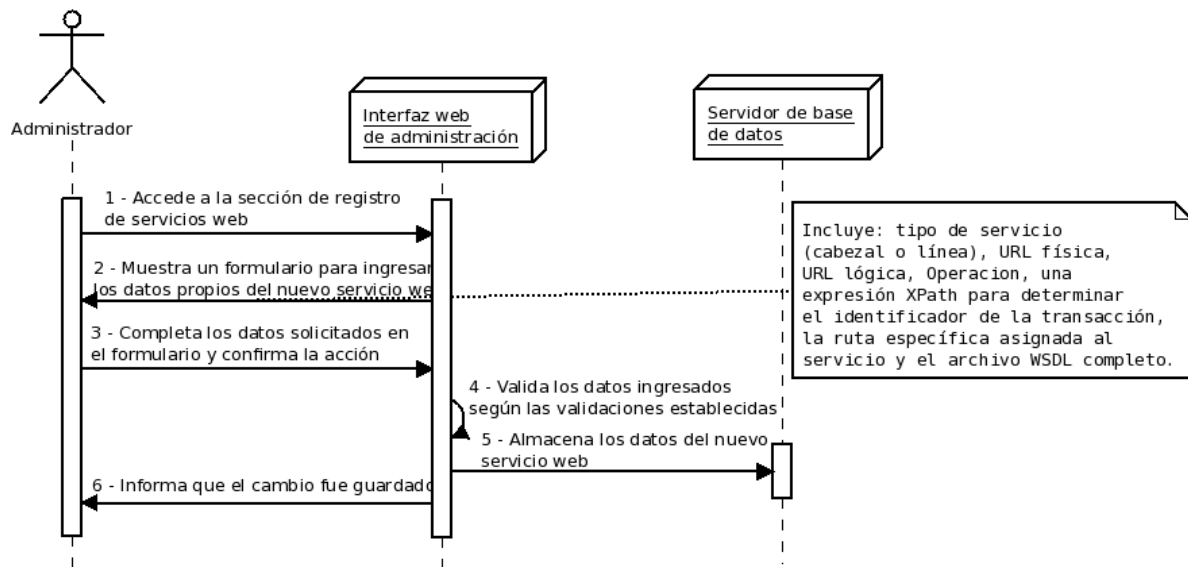


Figura 5.9: Interacciones en el proceso de registro de un nuevo servicio web

Este caso de uso permite registrar los dos tipos de servicios web: cabecal y línea. Los datos que componen el formulario son los mínimos necesarios para poder realizar la invocación de los servicios web publicados en la PDI, y deben ser proporcionados por AGESIC.

Flujos alternativos

En el punto 4, si alguna de las validaciones falla no se permite almacenar el servicio web en la base de datos y se vuelve al punto 2. Entre estas validaciones se incluye la presencia de valores para todos los parámetros, y que no exista otro nombre que coincida en alguno de dichos valores, entre otras.

5.4.6 RQF 2-04 - Consulta del estado del sistema

Este diagrama de secuencia muestra la interacción entre un cliente y el sistema para determinar si el estado del sistema: si el funcionamiento es correcto y si está activado el proceso de reenvío de trazas pendientes, entre otras cosas.

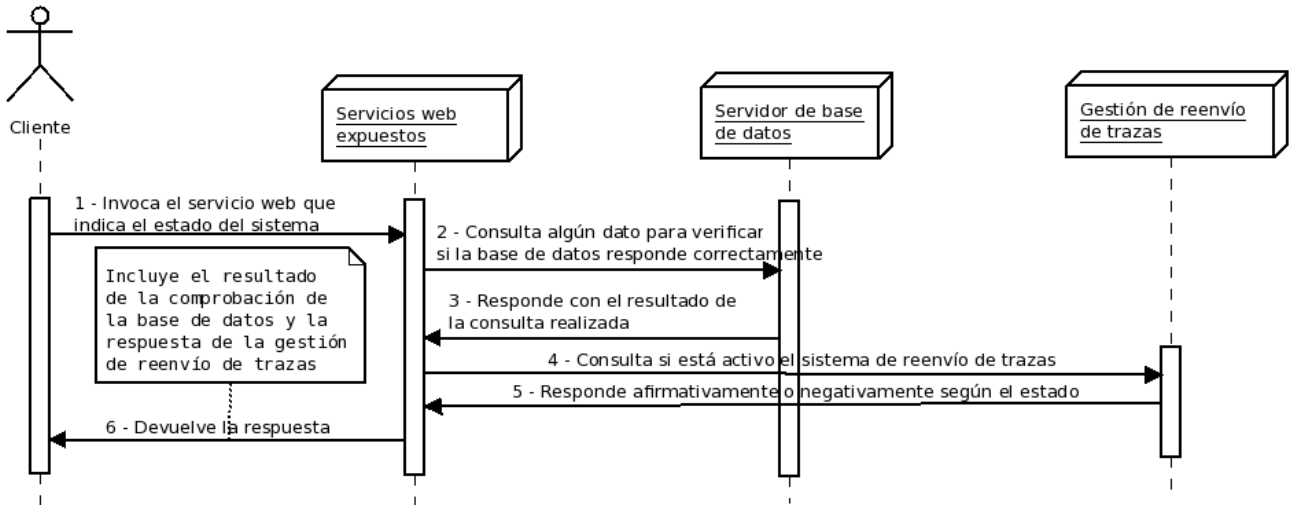


Figura 5.10: Interacciones en el proceso de consulta del estado del

5.5 Vista de procesos

Esta vista muestra los componentes principales previstos en la solución, y como interactúan entre sí para resolver el problema de la invocación de servicios web en la PDI.

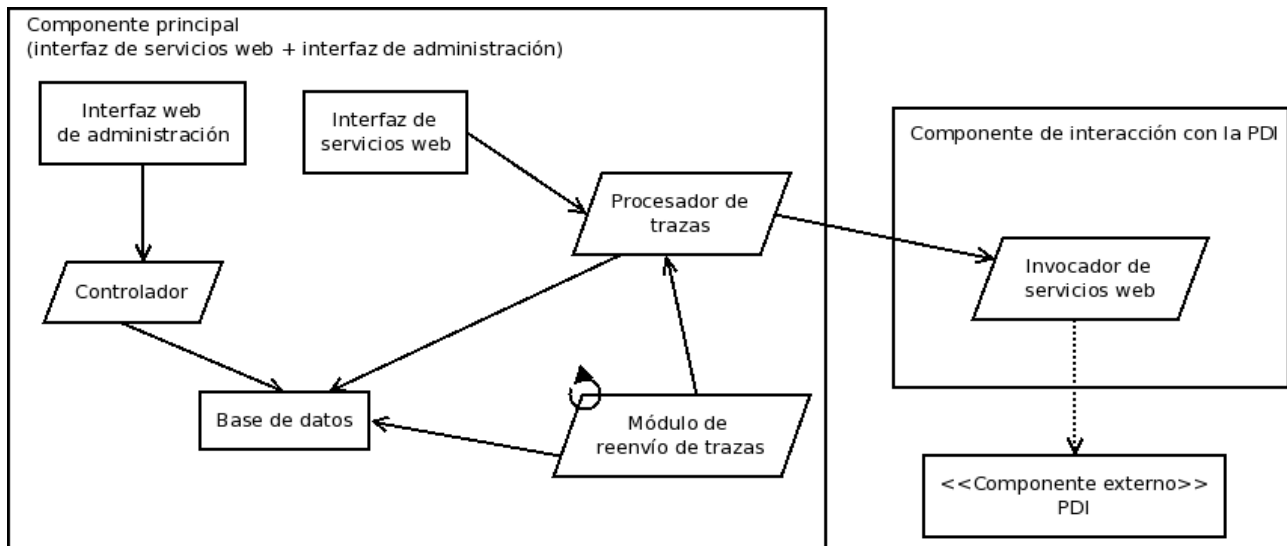


Figura 5.11: Diagrama de procesos

Los componentes que se muestran en la figura son los siguientes:

- **Componente principal:** constituye el cuerpo mayor de la aplicación. Es el que se encarga de proveer las funcionalidades correspondientes a la recepción de invocación de servicios web, el

reenvío de trazas pendientes, el acceso a la base de datos, y también ofrece la interfaz gráfica (web) de administración, tanto para registrar nuevos servicios, editar los existentes y consultar el estado del sistema. Este componente está conformado por lo siguiente:

- **Interfaces y dispositivos:**

- **Interfaz web de administración:** es la interfaz que utilizan los usuarios para interactuar directamente con el sistema.
- **Interfaz de servicios web:** es la interfaz que utilizan otras aplicaciones para invocar los servicios web configurados.
- **Base de datos:** es donde se almacenan los datos sobre los servicios web, las trazas, el estado del sistema y las invocaciones de los servicios web en la PDI.

- **Procesos:**

- **Controlador:** es un proceso que se encarga de tomar los datos de la interfaz web de administración y actuar según corresponda para cumplir con la tarea encomendada. Básicamente, obtiene y almacena información en la base de datos.
- **Procesador de trazas:** es un proceso que se encarga de tomar los datos de la interfaz de servicios web, y transferirlos al componente invocador de servicios web en la PDI; si corresponde, espera respuesta para retornarla.
- **Módulo de reenvío de trazas:** es un proceso que se ejecuta periódicamente y se encarga de obtener desde la base de datos las trazas pendientes de envío y transferirlas al componente invocador de servicios web en la PDI. También se encarga de manejar los errores, aumentando el tiempo entre ejecuciones o deteniendo completamente el proceso.
- **Invocador de servicios web:** es un proceso que se inicia a solicitud del procesador de trazas o del módulo de reenvío de trazas y se encarga de realizar el procedimiento necesario para invocar servicios web de la PDI devolviendo el resultado tal cual lo recibe.

5.6 Vista de base de datos

En esta sección se muestra el esquema planificado inicialmente para la base de datos. Se hace notar que podría cambiar posteriormente según existan modificaciones en los requerimientos o en las necesidades detectadas durante el desarrollo.

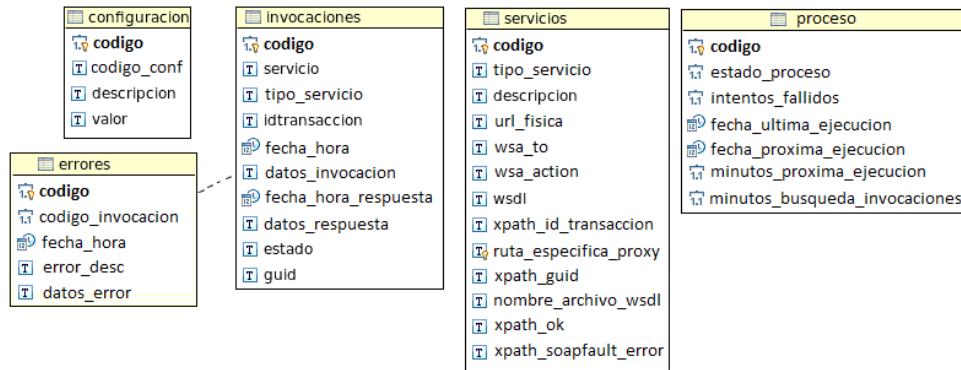


Figura 6.1: Esquema de base de datos inicial

6 Consideraciones para alta disponibilidad

La solución ConTraza podrá ser configurada para lograr alta disponibilidad. En una instalación de alta disponibilidad básicamente se replicará la configuración básica tantas veces como sea necesario o apropiado (con por lo menos dos nodos) excepto la base de datos, la cual continuará siendo una sola, accedida por todas las instancias de la aplicación, más otro servidor que funcionará como réplica del primero en forma pasiva (solo recibirá actualizaciones de datos a partir del servidor maestro pero no atenderá solicitudes de las instancias de la aplicación). Además, se configurará un balanceador de carga, que hará de punto de entrada entre los actores y la aplicación, distribuyendo las invocaciones que reciba entre todos los nodos que conformen la configuración.

El diagrama que muestra esto es el siguiente:

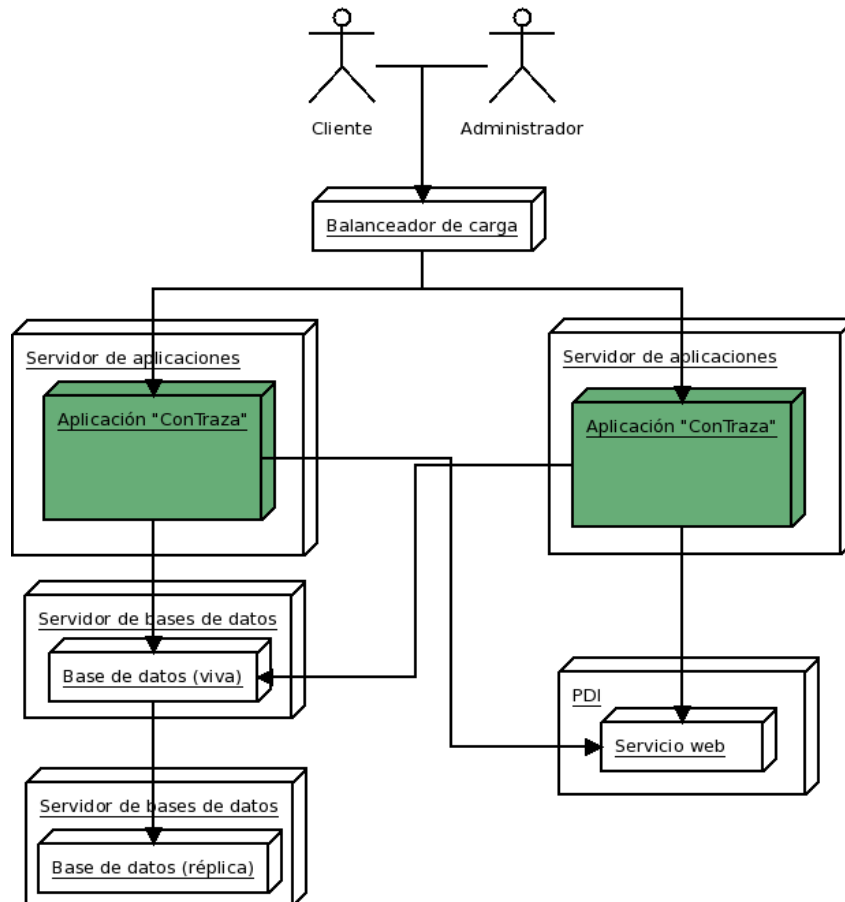


Figura 6.2: Solución en Alta Disponibilidad

Nota: dado que todos los nodos, al estar configurados de igual manera, intentarán el reenvío de trazas pendientes (las cuales toman de la misma base de datos), es necesario implementar una forma de mutua exclusión para que solo uno de ellos pueda hacerlo a la vez. Esto debe hacerse mediante un bloqueo a nivel de base de datos (por ejemplo, a través de una tabla específica), de forma de que cualquiera de los nodos, cuando el temporizador le ordene realizar el reenvío de trazas pendientes, deba primero intentar obtener el candado (lock); en caso de poder obtenerlo, debe intentar enviar las trazas pendientes y al finalizar devolver el candado (sea que pudo enviar las trazas o no), mientras que en caso de no poder obtenerlo debe dar por concluida la operación hasta la próxima instrucción del temporizador.

7 Requerimientos de Hardware

Esta sección tiene como objetivo establecer requerimientos de hardware para el despliegue de la aplicación en un ambiente de producción.

La instalación básica debe estar compuesta por dos equipos iguales, cada uno de los cuales contendrá una instalación de la aplicación (Tomcat 8 + aplicación), más un servidor de bases de datos.. No se considera que sea necesario configurar un ambiente con replicación de sesión pero sí de balanceo de carga como se detalló en el punto 6.

Se recomienda la siguiente configuración mínima (en el caso de contar con redundancia para proporcionar alta disponibilidad, estas configuraciones mínimas aplican a cada uno de los nodos):

- Memoria RAM: 8 GB.
- 4 GB Apache Tomcat 8.
- 2 GB Servidor de bases de datos.
- 2 GB Sistema Operativo.
- Espacio en disco: 80 Gigabytes.
- Conexión red: 1Gbit/s.

8 Consideraciones Tecnológicas

A continuación se detalla el software a ejecutar en dicho sistema.

- a) Sistema Operativo: Linux o MS Windows.
- b) Java: OpenJDK versión 1.8.
- c) Servidor de aplicaciones: Apache Tomcat 8.
- d) Servidor de Base de Datos: MySQL 5.6 o PostgreSQL 9.4.

9 Historial de cambios

Versión	Fecha	Autor	Detalle
1.0	14/04/2016	SP	Versión inicial
1.1	18/04/2016	SP	Ampliación de información.
1.2	22/04/2016	SP	Modificaciones por devoluciones del cliente.
1.3	10/05/2016	SP	Remoción de lo relativo a la base de datos empaquetada y el mantenimiento de orden de trazas; ampliación de información sobre el manejo de errores y bloqueo mutuo y otras modificaciones en base a la devolución.
1.4	01/06/2016	RI	Modificación para incorporar detalles respecto al módulo de integración. Y detalles de war y jars
1.5	12/06/2016	FD	Corrección según devolución de Ignacio Friedmann
1.6	17/06/2016	FD	Se eliminaron referencias a bases de datos empaquetadas.