



## AGESIC

### Arquitectura 4+1

Desarrollo  
versión 2.4

Histórico de revisiones			
FECHA	VERSION	DESCRIPCION	AUTOR
30/09/2015	1.0	Creación del documento	Valentina Rodríguez
2/10/2015	1.1	Actualización de diagramas	Valentina Rodríguez, Gonzalo Capra, Ignacio Álvarez.
15/12/2015	1.2	Actualización Fase 2	Valentina Rodríguez
18/03/2016	1.3	Actualización Fase 3	Ignacio Alvarez Valentina Rodríguez
9/5/2016	2	Actualización Fase 4 + Comentarios AGESIC	Ignacio Alvarez Valentina Rodríguez
2/6/2016	2.1	Actualización Comentarios AGESIC	Ignacio Alvarez
14/6/2016	2.2	Impacto de diagramas actualizados, con comentarios AGESIC	Ignacio Alvarez
1/7/2016	2.3	Actualización en base a sugerencias de AGESIC	Ignacio Alvarez, Carlos Yarza
7/12/2016	2.4	Actualización en base a sugerencias de AGESIC	Carlos Yarza, Adrián Curbelo

## Índice de contenido

1	Introducción.....	5
2	Escenarios.....	5
2.1	Casos de Uso.....	5
2.1.1	Actores.....	5
2.1.2	Diagramas.....	6
2.1.2.1	BackOffice.....	6
2.1.2.2	Acceso a Servicios de Trazabilidad.....	8
2.1.3	Especificación Casos de Uso.....	9
3	Vista Lógica.....	18
4	Vista de Implementación.....	19
4.1	Dependencias entre aplicaciones.....	19
4.2	Aplicación: itramites-bruto.....	20
4.2.1	Componentes y Responsabilidades.....	20
4.2.2	Diagrama de Componentes.....	20
4.2.3	Servicios Web.....	21
4.2.4	Interfaces de Negocio.....	21
4.2.5	Clases de Negocio.....	21
4.3	Aplicación: itramites-estructurado.....	22
4.3.1	Componentes y Responsabilidades.....	22
4.3.2	Diagrama de Componentes.....	23
4.3.3	Servicios Web.....	23
4.3.4	Interfaces de Negocio.....	23
4.3.5	Clases de Negocio.....	24
4.4	Aplicación: itramites-integración.....	25
4.4.1	Componentes y Responsabilidades.....	25
4.4.2	Diagrama de Componentes.....	26
4.4.3	Servicios Web.....	26
4.4.4	Interfaces de Negocio.....	26
4.4.5	Clases de Negocio.....	27
4.5	Aplicación: itramites-transferencia.....	28
4.5.1	Componentes y Responsabilidades.....	28
4.5.2	Diagrama de componentes.....	29
4.5.3	Servicios Web.....	29
4.5.4	Interfaces de Negocio.....	29
4.5.5	Clases de Negocio.....	29
4.6	Aplicación: itramites-frontend.....	31
4.6.1	Componentes y Responsabilidades.....	31
4.6.2	Diagrama de Componentes.....	32
4.6.3	Interfaces de Negocio.....	32
4.6.4	Clases de Negocio.....	32
4.7	Componentes Adicionales de la Solución.....	33
4.7.1	Portlet del ciudadano.....	33
4.7.1.1	Componentes y Responsabilidades.....	33
4.7.1.2	Diagrama de componentes.....	33
4.7.1.3	Comentarios Relevantes de Arquitectura.....	33
5	Vista Física.....	35
5.1	Arquitectura de deployment tipo para la aplicación (vista de componentes).....	35
5.2	Arquitectura de deployment para la aplicación (vista física).....	36
5.2.1	Producción.....	36
5.2.2	Pre-Producción.....	37

5.2.3 Testing Integración.....	38
5.2.4 Testing Funcional.....	39
5.3 Conclusiones.....	40
6 Vista de Procesos.....	40
6.1 Proceso de transferencia:.....	40
6.1.1 Funcionamiento del proceso de transferencia (modalidad convencional).....	41
6.1.2 Funcionamiento del proceso de transferencia (modalidad persistente).....	42
6.2 Proceso de validación.....	43
6.2.1 Validaciones disponibles.....	43
6.2.2 Funcionamiento del proceso de validación.....	43
6.2.3 Diagrama de Secuencia:.....	44
6.3 Proceso de reporte de calidad:.....	44
6.4 Proceso de generación de estadísticas:.....	46
6.5 Proceso de carga de guía de trámite:.....	46
7 Configuraciones.....	47
8 GLOSARIO.....	48
9 Anexo: Modelo de datos.....	49
9.1 Modelo de datos “Bruto”.....	49
9.2 Modelo de datos “Integración”.....	50
9.3 Modelo de datos “Frontend”.....	50
9.4 Modelo de datos “Estructurado”.....	51

# 1 Introducción

El objetivo de este documento es presentar las decisiones de arquitectura que se tomaron a la hora de implementar el sistema de Trazabilidad de Trámites. Para describir la arquitectura se utilizó el modelo 4+1. Primero se describen los escenarios / casos de uso y finalmente se especifican las cuatro vistas del modelo: vista lógica, vista de implementación, vista de procesos y la vista física.



Para cada vista se presentan los diagramas y se detalla lo necesario para comprender la arquitectura.

## 2 Escenarios

Los escenarios son una abstracción de los requerimientos más importantes. Para describir esta vista, se utilizaron diagramas de casos de uso.

### 2.1 Casos de Uso

#### 2.1.1 Actores

- Usuario backoffice: persona que se autentica en el sistema y se encarga de gestionar los trámites dependiendo del rol.
- Administrador: persona que se autentica en el sistema y tiene permisos de ABM sobre roles, usuarios, validadores, y listas de correos.
- Administrador intermedio: persona que se autentica en el sistema y tiene permisos de ABM sobre roles, usuarios, validadores, y listas de

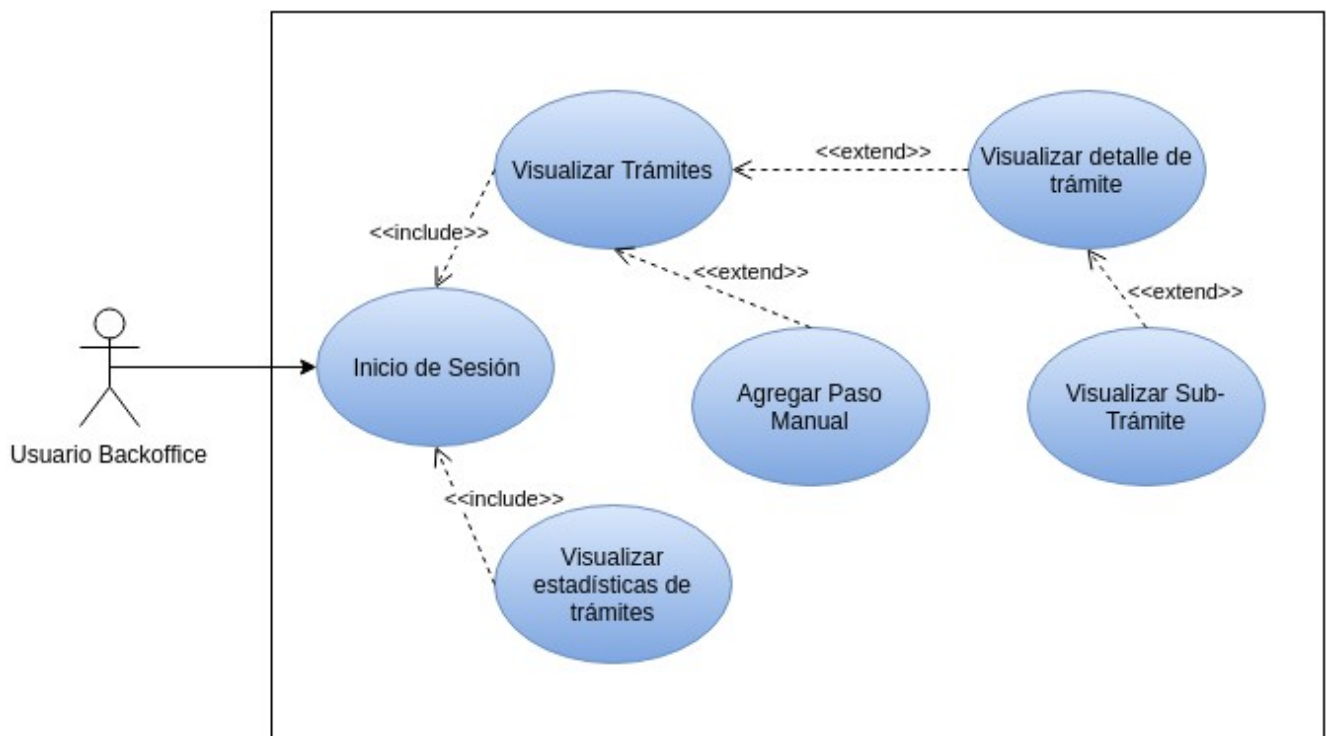
correos dependiendo del inciso/ue que tiene asociado.

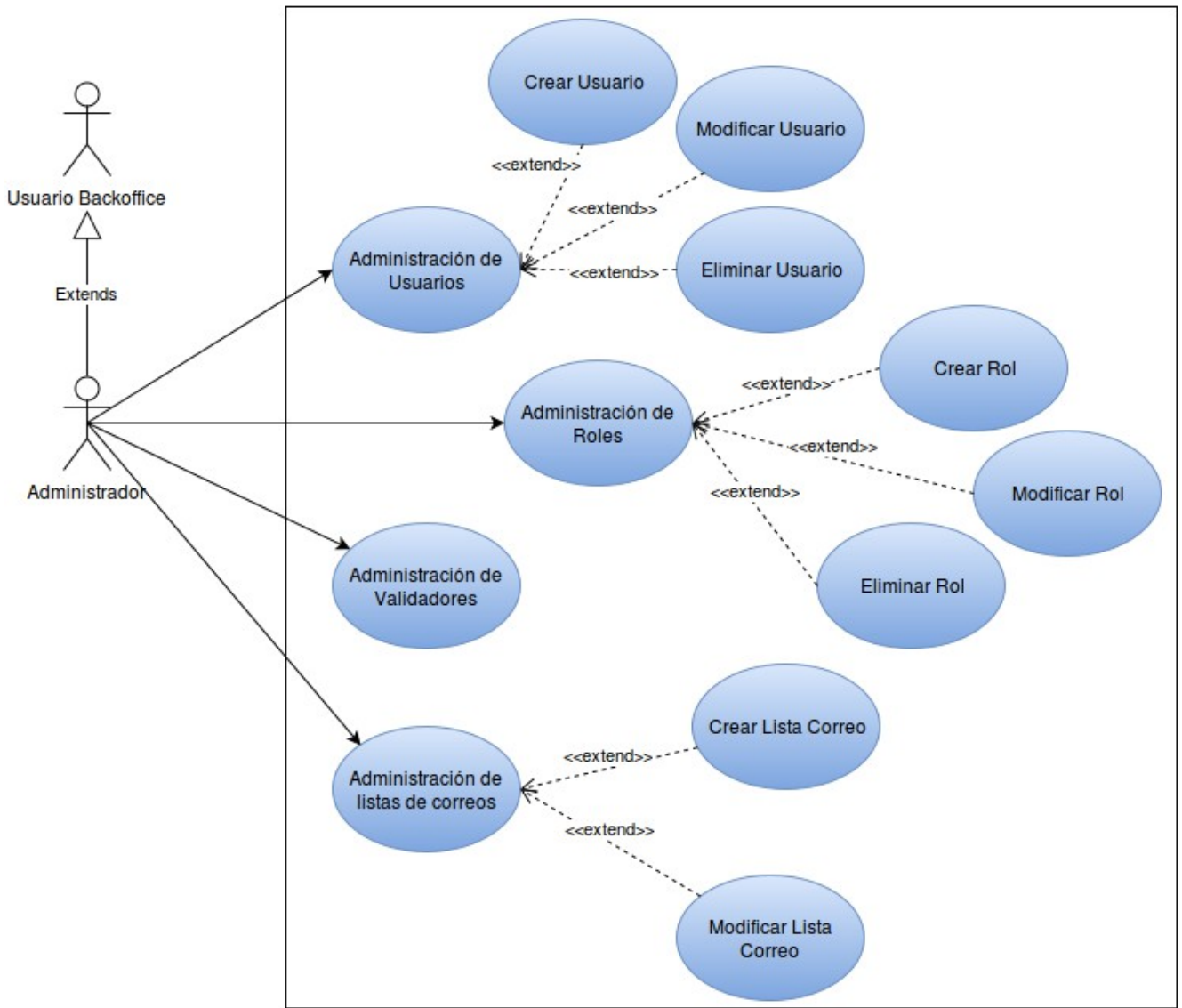
- Sistema externo: sistema que invoca a los servicios expuestos.
- Operador: persona que invoca los servicios de mantenimiento para agregar de forma masiva nuevos usuarios y roles.
- Ciudadano: persona que inicia el trámite.

## 2.1.2 Diagramas

A continuación se presentan los diagramas de casos de usos más relevantes

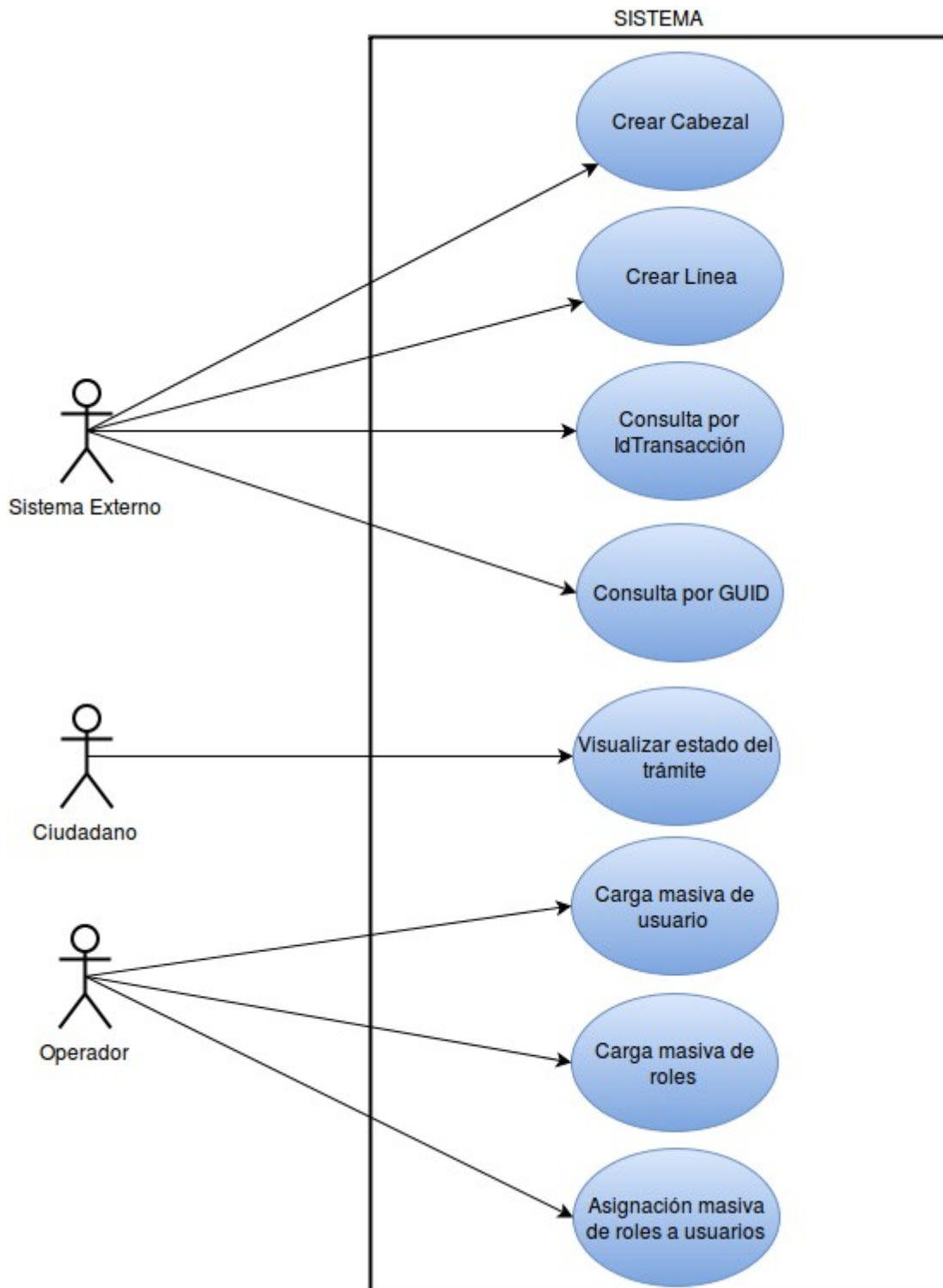
### 2.1.2.1 BackOffice





Nota: A los efectos de los casos de uso, los Administradores Intermedios, o el Super Administrador determinan flujos alternativos dentro de los mismos casos de uso.

## 2.1.2.2 Acceso a Servicios de Trazabilidad



### 2.1.3 Especificación Casos de Uso

A continuación se describen algunos de los casos de usos más relevantes.

#### ● **CU001 - Inicio de Sesión**

Pre-condición: Usuario está registrado en Clave única

Cuando el usuario intenta ingresar al sistema, se verifica que tenga credenciales cargadas. En caso de no tenerlas, el sistema redirige al sistema de autenticación federado (Clave única) para procesar la autenticación. Luego de recibir desde el sistema de autenticación federado (Clave única) la información asociada al usuario logueado (en base al protocolo SAML v2), el sistema carga en la sesión cuáles son los roles asociados al usuario.

#### ● **CU002 - Visualizar Trámites**

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario tiene al menos un rol asignado

Un usuario podrá visualizar trámites dependiendo del rol asignado.

Una vez que el usuario se autentica al sistema, el mismo lo redirige a la página de lista de trámites.

Si el usuario tiene un rol asignado, se le mostrarán todos los cabezales asociados al rol y tendrá la posibilidad de realizar filtros de búsqueda.

#### ● **CU003 - Visualizar detalle de un trámite**

Condición de extensión: Usuario presiona botón “Más Info” (imagen lupa).

Al hacer clic en la lupa que se encuentra a la derecha de cada trámite, el sistema redirige a la página “Detalle de Trámite” en la cual se visualizará la información de los pasos de ese trámite.

Los datos a mostrarse son: idTransacción, organismo, cantidad de pasos, paso, fecha y hora del organismo, oficina, tipo de paso, descripción del paso, estado, más info.

#### ● **CU004 - Agregar Paso Manual**

Condición de extensión: Usuario presiona botón “Agregar traza”

Si el usuario autenticado posee permisos para generar pasos, el sistema habilitará un botón a la derecha de cada trámite (con el símbolo de “+”). Al hacer clic en el mismo, el sistema despliega un modal en la cual se podrán ingresar los datos del nuevo paso.



Los datos a ingresar son: oficina, número de paso, descripción del paso, aclaraciones, paso del proceso, estado.

Una vez completados los datos, el usuario hace clic en “Agregar” y el sistema le muestra un mensaje de que se ingresó correctamente el paso. En caso de haber un error se muestra un mensaje de error en la parte superior del modal.

### ● **CU005 - Visualizar Sub-Trámite**

Condición de extensión: Usuario presiona botón “ver detalle” en una línea de tipo sub-trámite (imagen lupa).

Si el trámite posee sub-trámite, luego de ingresar en el detalle del trámite, el sistema presentará una lupa a la derecha de la línea correspondiente al sub-trámite, y al presionarla, el usuario podrá visualizar el detalle del sub-trámite vinculado.

### ● **CU006 - Visualizar estadísticas de trámites**

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Ejecución del proceso de generación de estadísticas (Ver sección procesos por más detalle).

Un usuario de backoffice autenticado podrá visualizar las estadísticas de los trámites que tienen al menos un registro en el sistema en el último año.

Al hacer clic en el menú sección Dashboard se despliega la información estadística de los trámites del organismo asociado al usuario.

Los datos a mostrarse son: Ranking de los 10 trámites con mayor cantidad de iniciados vs finalizados, ranking de los 10 trámites con mayor duración promedio, las 3 gráficas de los primeros 3 trámites con mayor cantidad de iniciados vs finalizados en el año y las 3 gráficas con los trámites de mayor duración promedio en el año.

Al hacer clic en los botones “Ver todos” se desplegará la información de todos los trámites del organismo.

### ● **CU007 - Crear Cabezal**

Se recibe un cabezal de traza, origen del rastreo realizado por el sistema. El sistema almacena la información recibida, y la deja disponible para ser procesada

Tipo de servicio: SOAP

Información de entrada: Información del cabezal (según estándar UNIT de trazabilidad)

Datos del servicio:

- tipoProceso: tipoProcesoEnum
- idProceso: long
- idTransaccion :string
- edicionModelo :long
- idTransaccionPadre :string
- pasoPadre :long
- cantidadPasosProceso :long
- inicioAsistidoProceso :inicioAsistidoProcesoEnum

- canalDelInicio :canalDelInicioEnum
- fechaHoraOrganismo :dateTime
- datoExtra1 :string
- datoExtra2 :string
- datoExtra3 :string
- datoExtra4 :string

Información de salida: Resultado de la operación, GUID generado para el cabezal

## ● CU008 - Crear Línea

Se recibe un paso en particular, de un cabezal ya registrado. El sistema establece un vínculo entre ambos, y lo almacena para ser procesado.

Tipo de servicio: SOAP

Información de entrada: Información del línea/paso (según estándar UNIT de trazabilidad)

Datos del servicio:

- idTransaccion :string
- edicionModelo :long
- idOficina :string
- oficina :string
- fechaHoraOrganismo :dateTime
- tipoRegistroTrazabilidad :tipoRegistroTrazabilidadEnum
- paso :long
- descripcionDelPaso :string
- aclaraciones :string
- pasoDelProceso :long
- idOficinaDestino :string
- oficinaDestino :string
- estadoProceso :estadoProcesoEnum
- datoExtra1 :string
- datoExtra2 :string
- datoExtra3 :string
- datoExtra4 :string

Información de salida: Resultado de la operación

## ● CU009 - Consulta por IdTransaccion

Consulta de información de trámites por id de transacción

Tipo de servicio: SOAP

Información de entrada: Identificador de transacción

Datos del servicio:

- idTransaccion :string
- rol :string - Rol de PDI asociado

Información de salida:

Estructura que contiene:

1. Cabezal (se detalla abajo)
2. Línea (se detalla abajo)
3. Subtramite - Se compone de cabezal y línea

**Estructura de CABEZAL:**

canalDelInicio :canalDelInicioEnum

datoExtra1 :string

datoExtra2 :string

datoExtra3 :string  
datoExtra4 :string  
estadoConsultable :estadoConsultableEnum  
estadoValidacion :estadoValidacionEnum  
fechaUltimaModificacion :dateTime  
guid :string  
idProceso :long  
idProcesoInfo :idProcesoInfo  
idTransaccion :string  
idTransaccionPadre :string  
inicioAsistidoProceso :inicioAsistidoProcesoEnum  
pasoPadre :long  
rol :string  
rolPDI :string  
tipoProceso :tipoProcesoEnum

#### **Estructura de IDPROCESOINFO:**

idProceso :long  
idUnidadEjecutora :string  
nombreIdProceso :string  
nombreUnidadEjecutora :string

#### **Estructura de LINEA:**

aclaraciones :string  
datoExtra1 :string  
datoExtra2 :string  
datoExtra3 :string  
datoExtra4 :string  
descripcionDelPaso :string  
estadoProceso :estadoProcesoEnum  
estadoValidacion :estadoValidacionEnum  
fechaHoraOrganismo :dateTime  
fechaHoraTrazabilidad :dateTime  
idOficina :string  
idOficinaDestino :string  
idTransaccion :string  
oficina :string  
oficinaDestino :string  
paso :long  
pasoDelProceso :long  
rolPDI :string  
tipoRegistroTrazabilidad :tipoRegistroTrazabilidadEnum

### ● **CU010 - Consulta por GUID**

Un sistema externo podrá consultar información de trámites invocando a un servicio expuesto por el sistema.

Tipo de servicio: SOAP (expuesto por PDI)

Se invoca al servicio con los datos que se describen a continuación y el sistema despliega la información del trámite.

Datos del servicio:

- consultaPorGUID :string

Información de salida:

Estructura que contiene:

1. Cabezal (se detalla en el caso de uso anterior)
2. Linea (se detalla en el caso de uso anterior)
3. Subtramite - Se compone de cabezal y linea

## ● **CU011 - Administración de Usuarios**

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario tiene rol de superAdmin

Al ingresar en la opción “Usuarios” del menú, el sistema muestra el listado completo de usuarios administrados, y las operaciones disponibles (creación, modificación y eliminación).

Cursos Alternativos:

### **Administración de usuarios para Inciso/Ue**

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario es administrador de Inciso/Ue y no superAdmin.

Al ingresar en la opción “Usuarios” del menú, el sistema muestra el listado completo de usuarios administrados, y las operaciones disponibles (creación, modificación y eliminación) para los incisos/Ues que administra el usuario.

## ● **CU012 - Crear Usuario**

Al hacer clic en Crear usuario el sistema redirige a página de creación de usuario, en la cual se ingresará el usuario (tiene que ser un usuario válido de clave única) y se permite asignar roles al usuario.

El usuario confirma la operación presionando el botón “Crear Usuario” y el sistema registra el usuario y la relación con los roles.

Cursos Alternativos:

### **Crear Usuario para Administrador de Inciso/Ue**

Al hacer clic en Crear usuario el sistema redirige a página de creación de usuario, en la cual se ingresará el usuario (tiene que ser un usuario válido de clave única) y se permite asignar roles al usuario. Estarán disponibles solamente los roles de los incisos administrados.

El usuario confirma la operación presionando el botón “Crear Usuario” y el sistema registra el usuario y la relación con los roles.

## ● **CU013 - Modificar Usuario**

Al hacer clic en el icono con forma de lápiz, el sistema redirige a página de creación de usuario, en la cual se desplegarán los datos del usuario seleccionado y se podrá modificar los roles que tiene asociados.

El usuario hace clic en “Modificar Usuario” y el sistema actualiza los roles del usuario.

## ● CU014 - Eliminar Usuario

Al hacer clic en el icono con forma de cruz, el sistema muestra un pop up de confirmación.

El usuario hace clic en Aceptar y el sistema realiza un borrado lógico del usuario.

## ● CU015 - Administración de Roles

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario tiene rol de superAdmin

Al ingresar en la opción “Roles” del menú, el sistema muestra el listado completo de roles administrados, y las operaciones disponibles (creación, modificación y eliminación).

Cursos Alternativos:

### **Administración de roles para Inciso/Ue**

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario es administrador de Inciso/Ue y no superAdmin.

Al ingresar en la opción “Roles” del menú, el sistema muestra el listado completo de roles administrados, y las operaciones disponibles (creación, modificación y eliminación) para los incisos/Ues que administra el usuario.

## ● CU016 - Crear Rol

El usuario ingresa nombre, selecciona el permiso, elige el organismo que pertenece (Inciso/Ue), selecciona si es administrador y hace clic en Crear Rol. El sistema registra el rol y lo muestra en el listado de roles.

Cursos Alternativos:

### **Crear Rol para Administrador de Inciso/Ue**

El usuario ingresa nombre, selecciona el permiso, elige el organismo que pertenece (Inciso/Ue), selecciona si es administrador y hace clic en Crear Rol. Estarán disponibles solamente los organismos de los incisos administrados (y sus ues asociadas al inciso).

## ● CU017 - Modificar Rol

Al hacer clic en el icono con forma de lápiz, el sistema carga los datos a ser modificados en la parte superior y hace clic en “Modificar Rol”.

El sistema actualiza la información del rol y muestra un mensaje de confirmación.

## ● CU018 - Eliminar Rol

Al hacer clic en el icono con forma de cruz, el sistema muestra un pop up de confirmación.

El usuario hace clic en Aceptar y el sistema elimina el rol si el mismo no está asociado a un usuario, de lo contrario despliega un mensaje de error.

## ● CU019 - Administración de validadores

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario tiene rol de superAdmin

Al ingresar en la opción “Validador” del menú, el sistema muestra el listado completo de los validadores administrados. Se podrán realizar filtros por inciso, unidad ejecutora, trámite y/o validador.

Se podrán habilitar o deshabilitar características de los validadores, como por ejemplo hacer que un validador sea bloqueante o no (en caso de ser un validador a nivel de registro no se podrá determinar como bloqueante). También se puede habilitar o deshabilitar el validador para que no sea ejecutado para ese trámite.

Cursos Alternativos:

## Administración de validadores para Inciso/Ue

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario es administrador de Inciso/Ue y no superAdmin.

Al ingresar en la opción “Validador” del menú, el sistema muestra el listado completo de los validadores administrados. Se podrán realizar filtros por inciso, unidad ejecutora, trámite y/o validador de acuerdo al inicio/ue administrado.

Se podrán habilitar o deshabilitar características de los validadores administrados, como por ejemplo hacer que un validador sea bloqueante o no (en caso de ser un validador a nivel de registro no se podrá determinar como bloqueante). También se puede habilitar o deshabilitar el validador para que no sea ejecutado para ese trámite.

## ● CU020 - Administración de listas de correos

Pre-condición: Usuario autenticado mediante clave única

Pre-condición: Usuario tiene rol de superAdmin

Al ingresar en la opción “Reportes” del menú, el sistema muestra el listado completo de las listas de correos. Se podrán realizar filtros por inciso, unidad ejecutora, nombre de lista y/o trámite.

## ● CU021 - Crear lista de correo

Al hacer clic en “Crear Lista” el sistema redirige a página de creación de la lista

de correo para reportes, en la cual se ingresará el nombre de la lista, se selecciona el inciso y la unidad ejecutora, se agregan los correos electrónicos separados por coma y finalmente se eligen los trámites que aplicará la lista. El usuario confirma la operación presionando el botón “Guardar” y el sistema registra la lista.

#### Cursos Alternativos:

### **Crear lista de correo para Administrador de Inciso/Ue**

Al hacer clic en “Crear Lista” el sistema redirige a página de creación de la lista de correo para reportes, en la cual se ingresará el nombre de la lista, se selecciona el inciso y la unidad ejecutora (estarán disponible solamente el inciso y unidades ejecutoras administradas por el usuario), se agregan los correos electrónicos separados por coma y finalmente se eligen los trámites que aplicará la lista.

El usuario confirma la operación presionando el botón “Guardar” y el sistema registra la lista.

### ● **CU022 - Modificar lista de correo**

Al hacer clic en el icono con forma de lápiz, el sistema redirige a página de edición de lista, en la cual se desplegarán los datos de la lista seleccionada y se podrá modificar los trámites y los correos electrónicos que tiene asociados.

El usuario hace clic en “Guardar” y el sistema actualiza los roles del usuario.

### ● **CU023 - Visualizar estado del trámite**

Un ciudadano podrá visualizar el estado del trámite.

El usuario ingresa el GUID y hace clic en el botón Ver estado de trámite.

El sistema despliega la información del trámite.

### ● **CU024 - Carga masiva de usuarios**

Pre-condición: El operador deberá tener acceso a los equipos físicos donde se ejecuta el servicio.

Un operador podrá crear un conjunto de usuarios invocando a un servicio expuesto por el sistema. Primero deberá copiar el archivo csv a un directorio existente (este debe tener asignado permisos de lectura y escritura) del servidor donde se ejecuta el servicio. Luego el operador invoca el servicio.

Tipo de servicio: SOAP

Se invoca al servicio con los datos que se describen a continuación y el sistema crea los usuarios que se indican en el archivo csv.

Datos del servicio:

- `nombreArchivo :string`

Información de salida: Resultado de la operación

## ● CU025 - Carga masiva de roles

Pre-condición: El operador deberá tener acceso a los equipos físicos donde se ejecuta el servicio.

Un operador podrá crear un conjunto de roles invocando a un servicio expuesto por el sistema. Primero deberá copiar el archivo csv a un directorio existente (este debe tener asignado permisos de lectura y escritura) del servidor donde se ejecuta el servicio. Luego el operador invoca el servicio.

Tipo de servicio: SOAP

Se invoca al servicio con los datos que se describen a continuación y el sistema crea los roles que se indican en el archivo csv.

Datos del servicio:

- `nombreArchivo :string`

Información de salida: Resultado de la operación

## ● CU026 - Asignación masiva de roles a usuarios

Pre-condición: El operador deberá tener acceso a los equipos físicos donde se ejecuta el servicio.

Un operador podrá asignar roles a usuarios invocando a un servicio expuesto por el sistema. Primero deberá copiar el archivo csv a un directorio existente (este debe tener asignado permisos de lectura y escritura) del servidor donde se ejecuta el servicio. Luego el operador invoca el servicio.

Tipo de servicio: SOAP

Se invoca al servicio con los datos que se describen a continuación y el sistema asigna los roles a los usuarios que se indican en el archivo csv.

Datos del servicio:

- `nombreArchivo :string`

Información de salida: Resultado de la operación

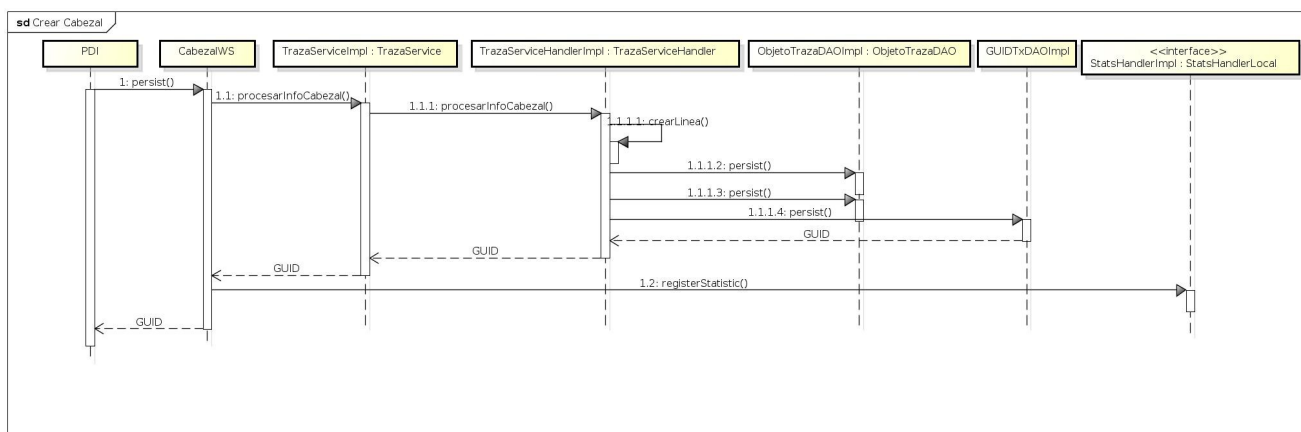


### 3 Vista Lógica

En esta vista se representa la funcionalidad que el sistema proporcionará a los usuarios finales. Es decir, se ha de representar lo que el sistema debe hacer, y las funciones y servicios que ofrece.

Para esta vista, se hizo una selección de casos de uso relevantes, respecto al template de arquitectura seguido dentro del desarrollo de la aplicación

Los WebServices ejecutan lógica interna, pero fundamentalmente invocan EJBs que resuelven la lógica de negocio (en este caso `TrazaServiceImpl`, implementación de la interfaz local `TrazaService`<sup>1</sup>). Se invoca a otro EJB que resuelve la lógica de parseo y procesamiento de los datos del cabezal, para luego invocar a los DAOs pertinentes. Luego se le delega al `statsHandlerLocal`<sup>2</sup> la captura de métricas de ejecución.



El WebService de captura de líneas sigue los mismos criterios generales: Para el caso de la interacción con páginas web (por ejemplo `consultaTrazas`), la representación conceptual de la página la realiza el framework de presentación (Wicket), y dentro de la implementación de la página, nos detenemos en uno de los métodos principales (`initUI`), que representa de qué manera un componente de presentación accede a la lógica del negocio para ir a buscar la información de las trazas. En este caso se delega a `IdProcesoService` la lógica de manejo de los ids de proceso (parte del proyecto de integración), y se delega la persistencia a un DAO.

1 Ver explicación de responsabilidades del componente en sección 4.2.4

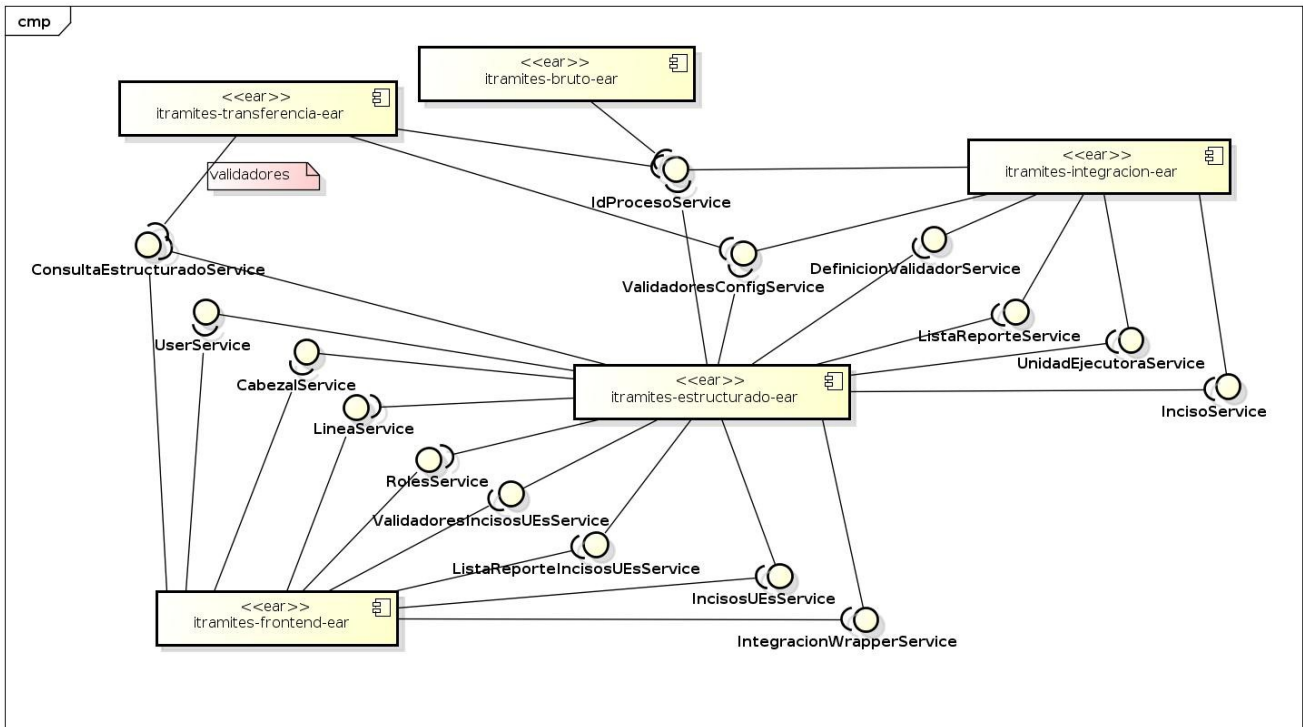
2 Ver explicación de responsabilidades del componente en sección 4.2.4

## 4 Vista de Implementación

Esta vista brinda la información desde la perspectiva del programador y muestra de forma estática la organización de los módulos de software.

Para mayor claridad dividimos los diagramas entre los 5 módulos principales.

### 4.1 Dependencias entre aplicaciones



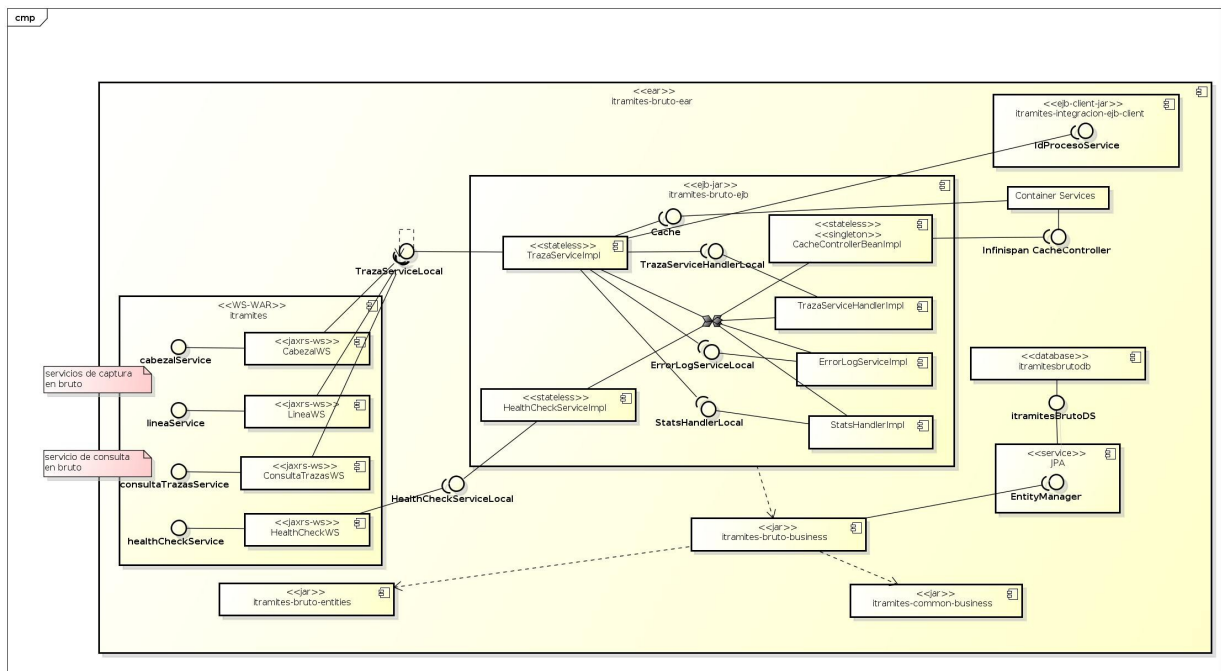
## 4.2 Aplicación: itramites-bruto

Esta aplicación se encarga fundamentalmente de la captura y consulta de trazas.

### 4.2.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
itramites	Implementación de servicios jax-rs	Exponer servicios web (SOAP y REST)
itramites-bruto-ejb	Interfaces locales y remotas a servicios (EJB) del negocio EJBs	Exponer interfaces locales y remotas a servicios (EJB) del negocio Fachada de invocaciones necesarias para realizar operaciones de negocio Interconexión entre módulos
itramites-bruto-business	Implementaciones concretas de DAOs Métodos utilitarios para el negocio de bruto	Acceso a capa de datos Lógica común
Itramites-bruto-entities	Entidades JPA	Mapeo Objeto-Relacional de las entidades
itramites-common-business	Código de negocio común a todas las aplicaciones. Implementaciones genéricas a patrones (como ser DAO). Código utilitario para módulos de negocio	Implementaciones genéricas de capa de persistencia
itramites-common	Lógica compartida en toda la aplicación	Código utilitario del negocio
itramites-common-dtos	Interfaces y DTOs	-
Itramites-integracion-ejb (ejb-client)	Cliente para acceder a los servicios de itramites-integracion-ejb	Acceso a información de trámites

### 4.2.2 Diagrama de Componentes



### 4.2.3 Servicios Web

- CabezaWS: Servicio Web de captura de cabecales - Para ser utilizado por aplicaciones externas a través de PDI.
- LineaWS: Servicio Web de captura de líneas - Para ser utilizado por aplicaciones externas a través de PDI.
- ConsultaTrazasWS: Servicio web de consulta de información “cruda” - En caso de ser utilizado, sería usado por operadores a través de PDI (opción: conector PGE).
- HealthCheckWS (soap y REST): Servicio web de consulta de salud de la aplicación. Retorna estadísticas de carga de los servicios de captura. Utilizado por opsview.

### 4.2.4 Interfaces de Negocio

- TrazaServiceLocal: Interface local que da soporte a los requerimientos de los servicios web de consulta y captura de trazas
- StatsHandlerLocal: Interface local que brinda una API para contabilizar estadísticas de captura de trazas (cantidad de invocaciones, tiempo de respuesta, etc.) y luego consultarlas.
- HealthCheckService / HealthCheckServiceLocal: servicios definidos para dar soporte a los servicios de monitoreo. Corresponden a las interfaces remota y local respectivamente. Ambas exponen las mismas operaciones.
- TrazaServiceHandlerLocal: interface local que brinda la funcionalidad de procesar y persistir la información de un cabezal.
- ErrorLogServiceLocal: interface local que permite guardar y consultar los errores detectados al momento de recibir las trazas, por ejemplo un error posible sería recibir un IdProceso no válido.

### 4.2.5 Clases de Negocio

- `TrazaServiceHandlerImpl`: Stateless SessionBean que implementa la interface `TrazaServiceHandlerLocal`.
- `TrazaServiceImpl`: Stateless SessionBean que implementa la interface `TrazaServiceLocal`.
- `HealthCheckServiceImpl`: Stateless SessionBean que implementa la interface `HealthCheckService`.
- `StatsHandlerImpl`: Stateless SessionBean que implementa la interface `StatsHandlerLocal`.
- `ErrorLogServiceImpl`: Stateless SessionBean que implementa la interface `ErrorLogServiceLocal`.
- `CacheControllerBeanImpl`: Singleton que contiene la lógica necesaria para manejar la cache de la lista de `IdProceso` y `GUID`.

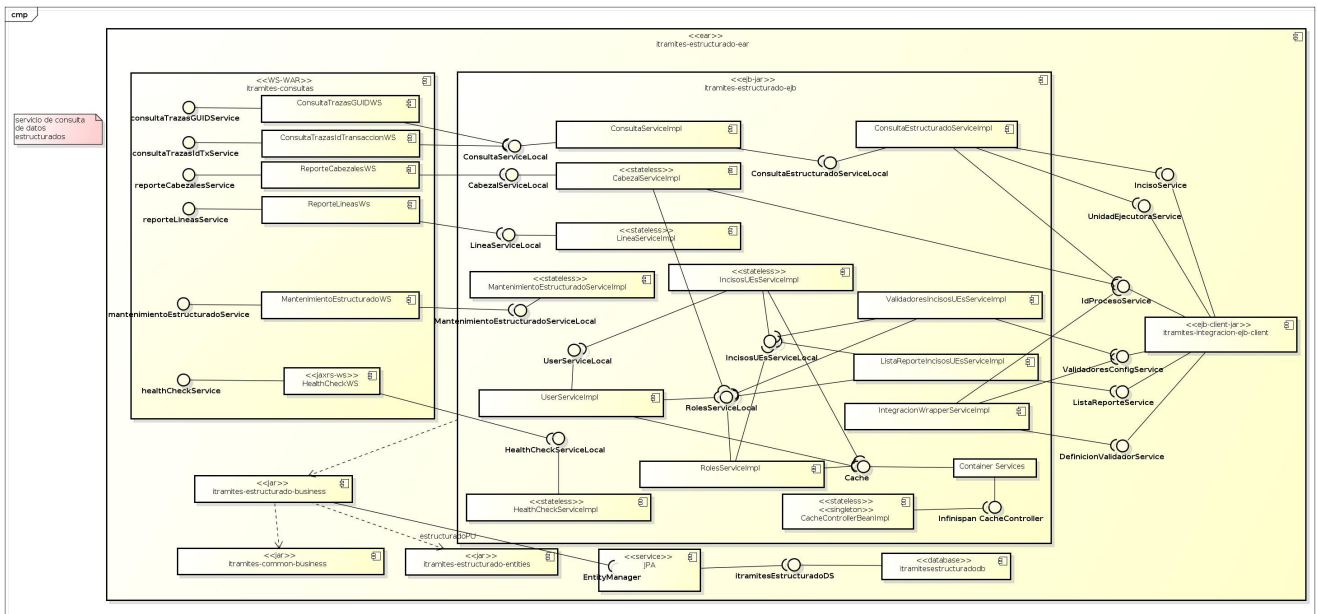
## 4.3 Aplicación: itramites-estructurado

Esta aplicación se encarga fundamentalmente en obtener información de las trazas transferidas, así como de la información de los usuarios, roles, incisos, unidades ejecutoras y validadores.

### 4.3.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
itramites-consultas	Implementación de servicios jax-rs	Exponer servicios web (SOAP y REST)
itramites-estructurado-ejb	Interfaces locales y remotas a servicios (EJB) del negocio EJBs	Exponer interfaces locales y remotas a servicios (EJB) del negocio Fachada de invocaciones necesarias para realizar operaciones de negocio Interconexión entre módulos
itramites-estructurado-business	Implementaciones concretas de DAOs Métodos utilitarios para el negocio de bruto	Acceso a capa de datos Lógica común
itramites-estructurado-entities	Entidades JPA	Mapeo Objeto-Relacional de las entidades
itramites-common-business	Código de negocio común a todas las aplicaciones. Implementaciones genéricas a patrones (como ser DAO). Código utilitario para módulos de negocio	Implementaciones genéricas de capa de persistencia
itramites-common	Lógica compartida en toda la aplicación	Código utilitario del negocio
itramites-common-dtos	Interfaces y DTOs	-
itramites-integracion-ejb (ejb-client)	Cliente para acceder a los servicios de itramites-integracion-ejb	Acceso a información de trámites

## 4.3.2 Diagrama de Componentes



## 4.3.3 Servicios Web

- **ConsultaTrazasGUIDWS:** Servicio web de consulta de información de trazas. Para ser usado por operador para consultar información particular a través de PDI.
- **ConsultaTrazasIdTransaccionWS:** Servicio web de consulta de información de trazas. Para ser usado por operador para consultar información particular a través de PDI.
- **ReporteCabezasWS:** Servicio web de consulta de información de trazas. No expuesto hoy para ser usado externamente vía PDI.
- **ReporteLineasWS:** Servicio web de consulta de información de trazas. No expuesto hoy para ser usado externamente vía PDI.
- **MantenimientoEstructuradoWS:** Servicio web de ejecución de tareas de mantenimiento. Para ser utilizado por operadores con acceso a los equipos físicos (es necesario copiar los archivos al servidor donde se ejecute).
- **HealthCheckWS (soap y REST):** Servicio web de consulta de salud de la aplicación. Utilizado por opsview

## 4.3.4 Interfaces de Negocio

- **CabezalService / CabezalServiceLocal:** Servicios de consulta de cabezales. Interfaz remota utilizada por el FrontEnd, interfaz local (con las mismas operaciones) utilizada por el servicio web ReporteCabezasWS.
- **LineaService / LineaServiceLocal:** Servicios de consulta de líneas. Interfaz remota utilizada por el FrontEnd, interfaz local (con las mismas operaciones) utilizada por el servicio web ReporteLineasWS.
- **ConsultaServiceLocal:** Servicio de apoyo a los servicios web de

- consulta (ConsultaTrazasPorGUIDWS y ConsultaTrazasIdTransaccionWS). Corresponde a una interface local.
- UserService / UserServiceLocal: Interface remota y local que encapsula las funciones de negocio relacionadas con consulta de permisos y roles del usuario logueado.
  - RolesServiceLocal: Interface que expone las operaciones de consulta para la gestión de roles de usuario.
  - HealthCheckService / HealthCheckServiceLocal: interface remota y local (mismo conjunto de operaciones) que da soporte a los servicios de monitoreo.
  - ConsultaEstructuradoServiceLocal: Interface local que brinda funcionalidades de consulta por GUID, IdProceso, IdCabezal y IdTransacción.
  - MantenimientoEstructuradoServiceLocal: interface local que da soporte a las operaciones de alta de usuarios y roles a partir de un archivo csv.
  - IdProcesoWrapperService: interface remota que permite obtener los permisos asociados al par usuario e idProceso.
  - IncisosUEsServiceLocal: interface local que encapsula las funcionalidades de consultas de Incisos y Unidades ejecutoras.
  - ValidadoresIncisosUEsService: interface remota que brinda las operaciones de consulta de validadores filtrando por inciso y unidad ejecutora.
  - ListaReporteIncisosUEsService: interface remota que encapsula las funcionalidades alta, modificación y consulta de lista destinatarios para los reportes.

#### 4.3.5 Clases de Negocio

- ConsultaEstructuradoServiceImpl: Stateless SessionBean que implementa la interface ConsultaEstructuradoServiceLocal.
- LineaServiceImpl: Stateless SessionBean que implementa las interfaces LineaService y LineaServiceLocal.
- UserServiceImpl: Stateless SessionBean que implementa las interfaces UserService y UserServiceLocal.
- RolesServiceImpl: Stateless SessionBean que implementa la interface RolesServiceLocal.
- CabezalServiceImpl: Stateless SessionBean que implementa las interfaces CabezalService y CabezalServiceLocal.
- ConsultaServiceImpl: Stateless SessionBean que implementa la interface ConsultaServiceLocal.
- CacheControllerBeanImpl: Singleton que contiene la lógica necesaria para manejar la cache asociada a las unidades ejecutoras, información del usuario y roles.
- HealthCheckServiceImpl: Stateless SessionBean que implementa las



interfaces de HealthCheckService y HealthCheckServiceLocal.

- MantenimientoEstructuradoServiceImpl: Stateless SessionBean que implementa la interface MantenimientoEstructuradoServiceLocal.
- IdProcesoWrapperServiceImpl: Clase encargada de implementar la interface IdProcesoWrapperService.
- IncisosUEsServiceImpl: Stateless SessionBean que implementa la interface IncisosUEsServiceLocal.
- ValidadoresIncisosUEsServiceImpl: Stateless SessionBean que implementa la interface ValidadoresIncisosUEsService.
- ListaReporteIncisosUEsServiceImpl: Stateless SessionBean que implementa la interface ListaReporteIncisosUEsService.

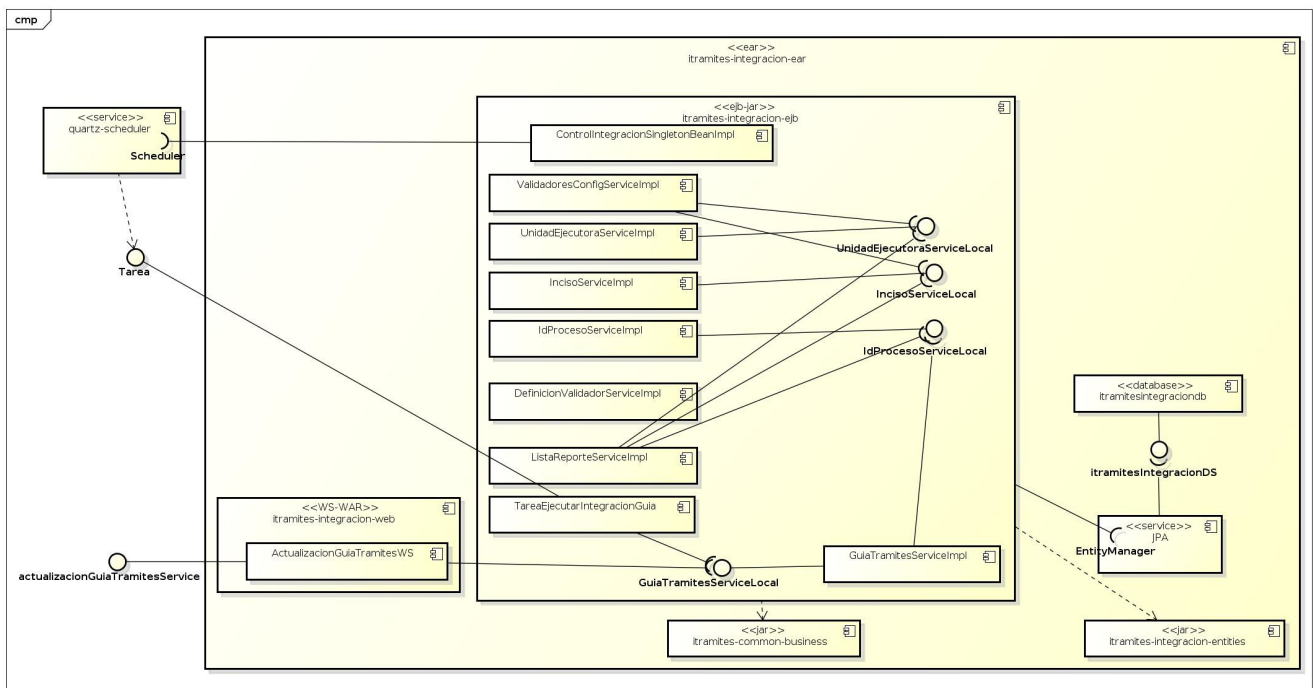
## 4.4 Aplicación: itramites-integración

Esta aplicación se encarga principalmente de la sincronización y mantenimiento de los trámites a través del consumo del servicio expuesto por la Guía de trámites.

### 4.4.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
itramites-integracion-web	Implementación de servicios jax-rs	Exponer servicios web (SOAP y REST)
itramites-integracion-ejb	Interfaces locales y remotas a servicios (EJB) del negocio EJBs	Exponer interfaces locales y remotas a servicios (EJB) del negocio Fachada de invocaciones necesarias para realizar operaciones de negocio Interconexión entre módulos
itramites-integracion-entities	Entidades JPA	Mapeo Objeto-Relacional de las entidades
itramites-common-business	Código de negocio común a todas las aplicaciones. Implementaciones genéricas a patrones (como ser DAO). Código utilitario para módulos de negocio	Implementaciones genéricas de capa de persistencia
itramites-common	Lógica compartida en toda la aplicación	Código utilitario del negocio
itramites-common-dtos	Interfaces y DTOs	-

## 4.4.2 Diagrama de Componentes



## 4.4.3 Servicios Web

- ActualizacionGuiaTramitesWS: Servicio web que permite ejecutar a demanda la sincronización con guía de trámites (no expuesto a PDI)

## 4.4.4 Interfaces de Negocio

- ValidadoresConfigService / ValidadoresConfigServiceLocal: Interfaces remota y local que encapsula la lógica de obtención de la configuración específica (jerárquica) para un validador.
- UnidadEjecutoraService / UnidadEjecutoraServiceLocal: Interfaces remota y local que brinda la funcionalidad de consulta de unidad ejecutora.
- IncisoService / IncisoServiceLocal: Interfaces remota y local que brinda la funcionalidad de consulta de inciso.
- IdProcesoService / IdProcesoServiceLocal: Interfaces remota y local que encapsula las funcionalidades de IdProceso.
- GuiaTramitesServiceLocal: interface local que brinda la funcionalidad de sincronización con el servicio de la guía de trámites.
- DefinicionValidadorService / DefinicionValidadorServiceLocal: interfaces remota y local que contienen las operaciones de consulta sobre la definición de los validadores existentes.
- ListaReporteService / ListaReporteServiceLocal: Interfaces remota y local que encapsula las operaciones de alta, modificación y consulta de lista de destinatarios para los reportes.
- Tarea: Interface que expone el método ejecutar para invocar la funcionalidad agendada.

#### 4.4.5 Clases de Negocio

- ValidadoresConfigServiceImpl: Stateless SessionBean que implementa las interfaces ValidadoresConfigService y ValidadoresConfigServiceLocal.
- UnidadEjecutoraServiceImpl: Stateless SessionBean que implementa las interfaces UnidadEjecutoraService y UnidadEjecutoraServiceLocal.
- IncisoServiceImpl: Stateless SessionBean que implementa las interfaces IncisoService y IncisoServiceLocal.
- IdProcesoServiceImpl: Stateless SessionBean que implementa las interfaces IdProcesoService y IdProcesoServiceLocal.
- GuiaTramitesServiceImpl: Singleton que implementa la interface GuiaTramitesServiceLocal.
- DefinicionValidadorServiceImpl: Stateless SessionBean que implementa las interfaces DefinicionValidadorService y DefinicionValidadorServiceLocal.
- ListaReporteService: Stateless SessionBean que implementa las interfaces ListaReporteService y ListaReporteServiceLocal.
- TareaEjecutarIntegracionGuiaTramites: Singleton que implementa la interface Tarea, cada vez que se invoca el método ejecutar se sincronizan los trámites con los de Guía de Trámites a través de la interface GuiaTramitesServiceLocal.
- ControllIntegracionSingletonBeanImpl: Singleton que se encarga de agendar el proceso quartz de sincronización con Guía de trámites, se agenda la clase TareaEjecutarIntegracionGuiaTramites.

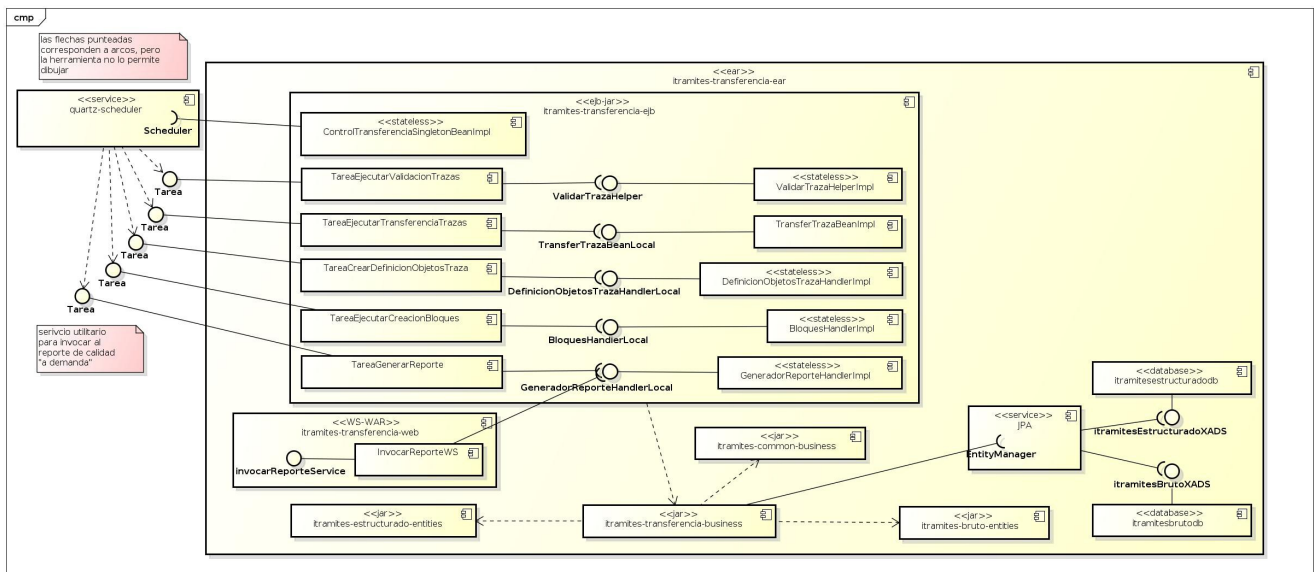
## 4.5 Aplicación: itramites-transferencia

Esta aplicación se encarga fundamentalmente de los procesos de transferencia, y validación de trazas. Para el caso de la transferencia, existen dos modalidades de funcionamiento: convencional y persistente.

### 4.5.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
itramites-transferencia-web	Implementación de servicios jax-rs	Exponer servicios web (SOAP y REST)
itramites-transferencia-ejb	Interfaces locales y remotas a servicios (EJB) del negocio EJBs	Exponer interfaces locales y remotas a servicios (EJB) del negocio Fachada de invocaciones necesarias para realizar operaciones de negocio Interconexión entre módulos
itramites-transferencia-business	Implementaciones concretas de DAOs Métodos utilitarios para el negocio de bruto	Acceso a capa de datos Lógica común
itramites-estructurado-entities itramites-bruto-entities	Entidades JPA	Mapeo Objeto-Relacional de las entidades
itramites-common-business	Código de negocio común a todas las aplicaciones. Implementaciones genéricas a patrones (como ser DAO). Código utilitario para módulos de negocio	Implementaciones genéricas de capa de persistencia
itramites-common	Lógica compartida en toda la aplicación	Código utilitario del negocio
itramites-common-dtos	Interfaces y DTOs	-

## 4.5.2 Diagrama de componentes



## 4.5.3 Servicios Web

- InvocarReporteWS: Permite ejecutar a demanda el reporte de validadores y fallos

## 4.5.4 Interfaces de Negocio

- ValidarTrazaHelperLocal: Interface local que expone las operaciones de consulta y validación de trazas completas.
- TransferTrazaBeanLocal: Interface local que brinda las funcionalidades para reagendar las trazas colgadas y transferir las trazas desde la base cruda a la estructurada.
- DefinicionObjetoTrazaHandlerLocal: Interface local que encapsula las funcionalidades de la definición de los objetos trazas y asociación con los bloques de transferencia.
- BloquesHandlerLocal: Interface local que contiene las operaciones necesarias para crear y manejar los bloques de transferencias de trazas.
- GeneradorReporteHandlerLocal: interface local que brinda las funcionalidades de envío de correo con los reportes de fallos.

## 4.5.5 Clases de Negocio

- ValidarTrazaHelperImpl: Singleton que implementa la interface local ValidarTrazaHelperLocal.
- TransferTrazaBeanImpl: Singleton que implementa la interface TransferTrazaBeanLocal.
- DefinicionObjetosTrazaHandlerImpl: Singleton que implementa la interface DefinicionObjetosTrazaHandlerLocal.
- BloqueHandlerImpl: Stateless SessionBean que implementa la interface BloquesHandlerLocal.

- `GeneradorReporteHandlerImpl`: `Stateless SessionBean` que implementa la interface `GeneradorReporteHandlerLocal`.
- `ControlTransferenciaSingletonBeanImpl`: `Singleton` que se encarga de agendar los procesos quartz correspondientes a las tareas detalladas a continuación.
- Las siguientes clases son todas tareas que se programan utilizando `Quartz Scheduler`:
  - `TareaEjecutarCreacionBloques`: Lanza una ejecución del proceso de creación de bloques, internamente invoca al EJB `BloquesHandlerLocal`.
  - `TareaCrearDefinicionObjetosTraza`: Lanza una ejecución del proceso de creación de definición de objetos traza, internamente invoca el EJB `DefinicionObjetosTrazaHandlerLocal`.
  - `TareaEjecutarTransferenciaTrazas`: Lanza una ejecución del proceso de transferencia de trazas. Invoca internamente al EJB `TransferTrazaBeanLocal`.
  - `TareaEjecutarValidacionTrazas`: Lanza una ejecución del proceso de validación de trazas. Invoca internamente al EJB `ValidarTrazaHelperLocal`.
  - `TareaGenerarReporte`: Lanza una ejecución del reporte de validadores y fallos, interamente se invoca al EJB `GeneradorReporteHandlerLocal`.

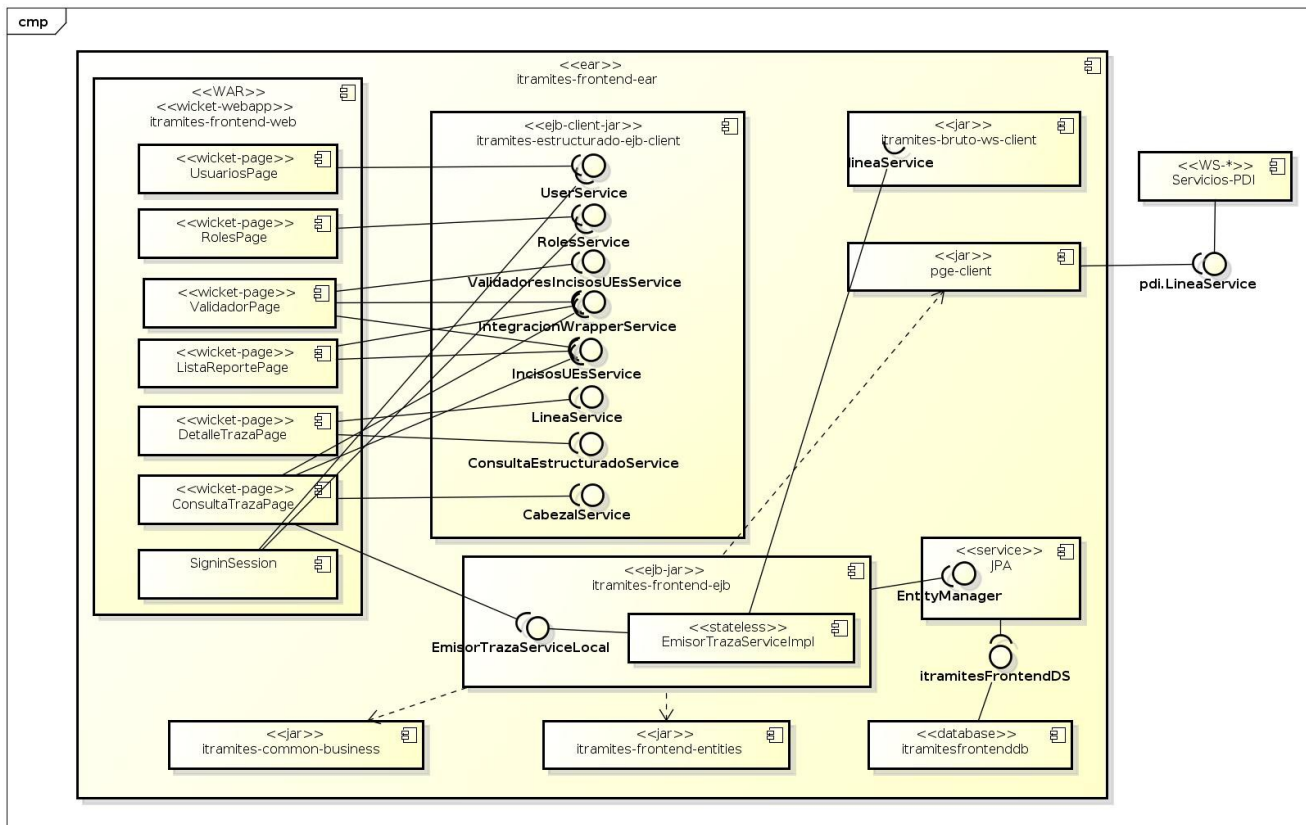
## 4.6 Aplicación: itramites-frontend

Esta aplicación se encarga principalmente de la lógica y visualización de las páginas web asociadas a la administración de trazas, usuarios, roles, validadores y lista de destinatarios para reporte.

### 4.6.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
itramites-frontend-web	Implementación de Interfaz de Usuario en Wicket	Presentar una interfaz de usuario para el BackOffice Validaciones mínimas de datos previo a consultas al negocio
itramites-frontend-ejb	Interfaces locales y remotas a servicios (EJB) del negocio EJBs	Exponer interfaces locales y remotas a servicios (EJB) del negocio Fachada de invocaciones necesarias para realizar operaciones de negocio Interconexión entre módulos
itramites-frontend-entities	Entidades JPA	Mapeo Objeto-Relacional de las entidades
itramites-common-business	Código de negocio común a todas las aplicaciones. Implementaciones genéricas a patrones (como ser DAO). Código utilitario para módulos de negocio	Implementaciones genéricas de capa de persistencia
itramites-common	Lógica compartida en toda la aplicación	Código utilitario del negocio
itramites-common-dtos	Interfaces y DTOs	-
itramites-bruto-ws-client	Interfaz y cliente de servicios JAX-WS	Exponer interfaz para consumo de servicio de captura de líneas (emisor de trazas)
itramites-estructurado-ejb-client	Cliente para acceder a los servicios de itramites-estructurado-ejb	Acceso al negocio de trazabilidad

## 4.6.2 Diagrama de Componentes



## 4.6.3 Interfaces de Negocio

- EmisorTrazaServiceLocal: Interface local que cumple la funcionalidad de proxy de acceso al WS de generación de líneas. Utiliza el patrón Strategy para determinar si el acceso es por PDI o consumo de WS "Local".

## 4.6.4 Clases de Negocio

- EmisorTrazaServiceImpl: Stateless SessionBean que implementa la interface EmisorTrazaServiceLocal.
- UsuarioPage: clase que contiene la lógica asociada a la página de los usuarios.
- RolesPage: clase que contiene la lógica asociada a la página de los roles.
- ValidadorPage: clase que contiene la lógica asociada a la página de los validadores.
- ListaReportePage: clase que contiene la lógica asociada a la página de la lista de destinatarios para los reportes.
- DetalleTrazaPage: clase que contiene la lógica asociada a la página del detalle de una traza.
- ConsultaTrazaPage: clase que contiene la lógica correspondiente a la página de consulta de trazas.



- SignInSession: clase que encapsula la lógica de autenticación de los usuarios.

## 4.7 Componentes Adicionales de la Solución

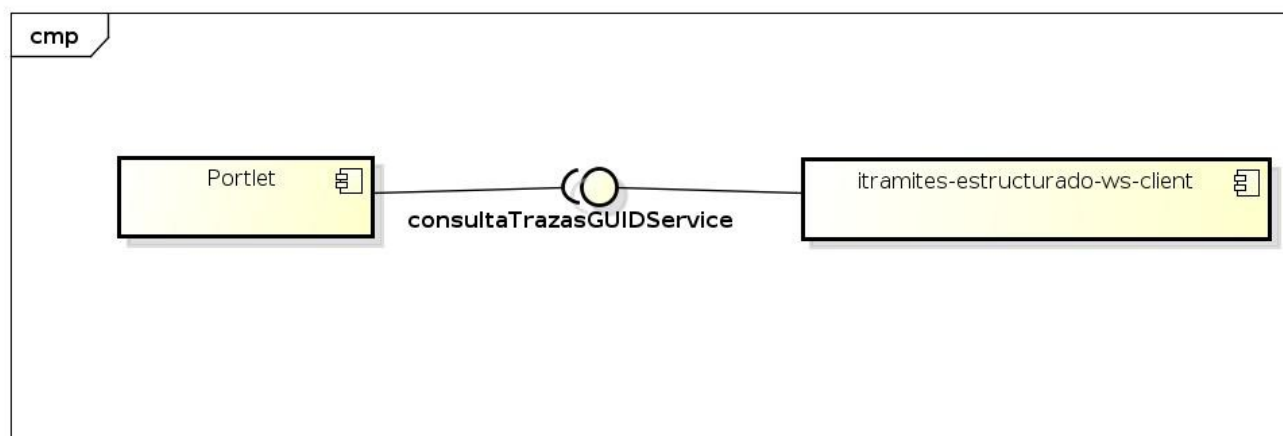
### 4.7.1 Portlet del ciudadano

El portlet del ciudadano es un componente que implementa el estándar portlet 2.0, y genera una vista de consulta de trazas. La presentación utiliza el framework Spring-MVC, presentando dos páginas (solicitud de Guid y despliegue de resultados). La información desplegada se obtiene a partir de una consulta al servicio ConsultaTrazasGUIDWS. El componente está desarrollado en Java 7, pero es compatible con la ejecución en Java 6

#### 4.7.1.1 Componentes y Responsabilidades

Componente	Elementos Principales	Principales Reponsabilidades
Portlet	Implementación del estándar Portlet 2.0. Spring MVC se encarga de las transiciones entre pantallas y los mapeos.	Presentar una interfaz de usuario para capturar GUID a consultar y desplegar la información que devuelve el servicio
itramites-estructurado-ws-client	Interfaz y cliente de servicios JAX-WS	Exponer interfaz para consumo de servicio de consulta de trazas por GUID

#### 4.7.1.2 Diagrama de componentes



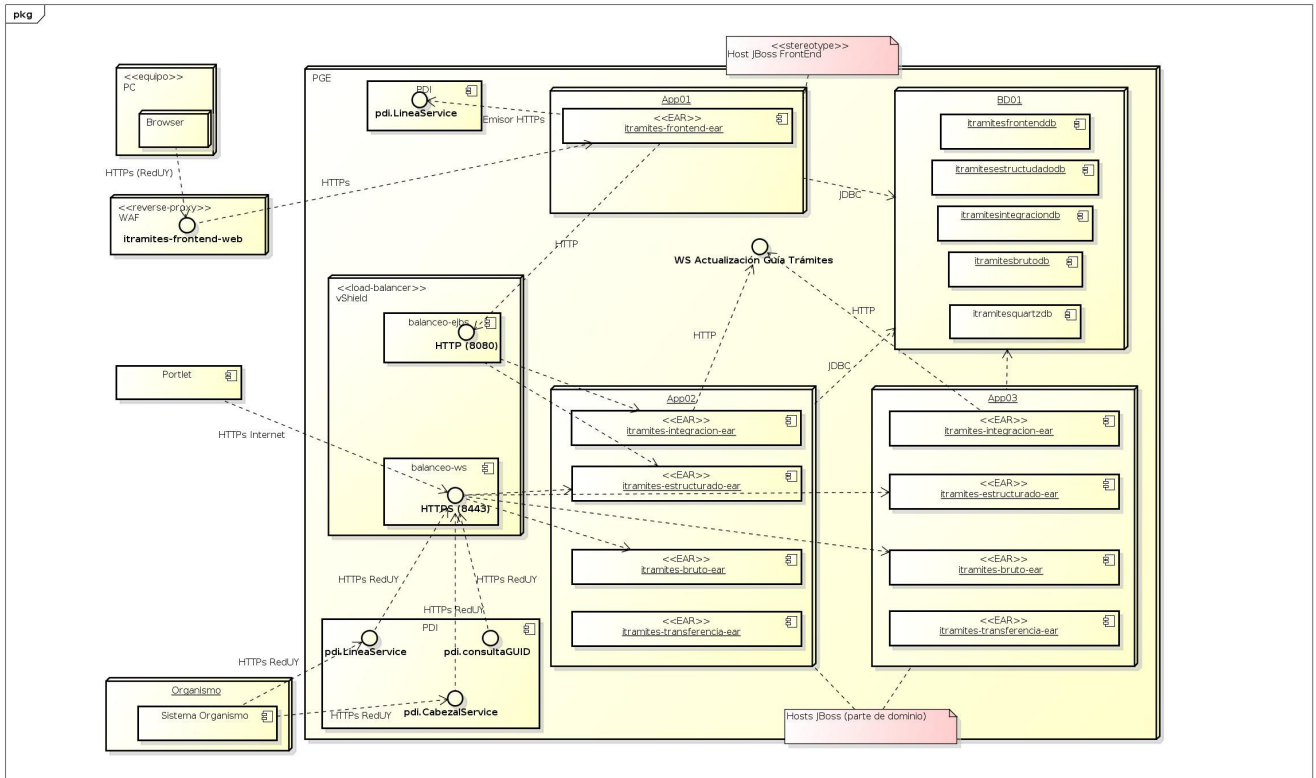
#### 4.7.1.3 Comentarios Relevantes de Arquitectura

Para este componente, y con el objetivo de no limitar la portabilidad en tiempo de ejecución, se tomó la decisión de no ir a través de plataforma de interoperabilidad, sino que se hace una consulta directa al servicio, delegando la seguridad de acceso al servicio a las configuraciones pertinentes de firewalls.



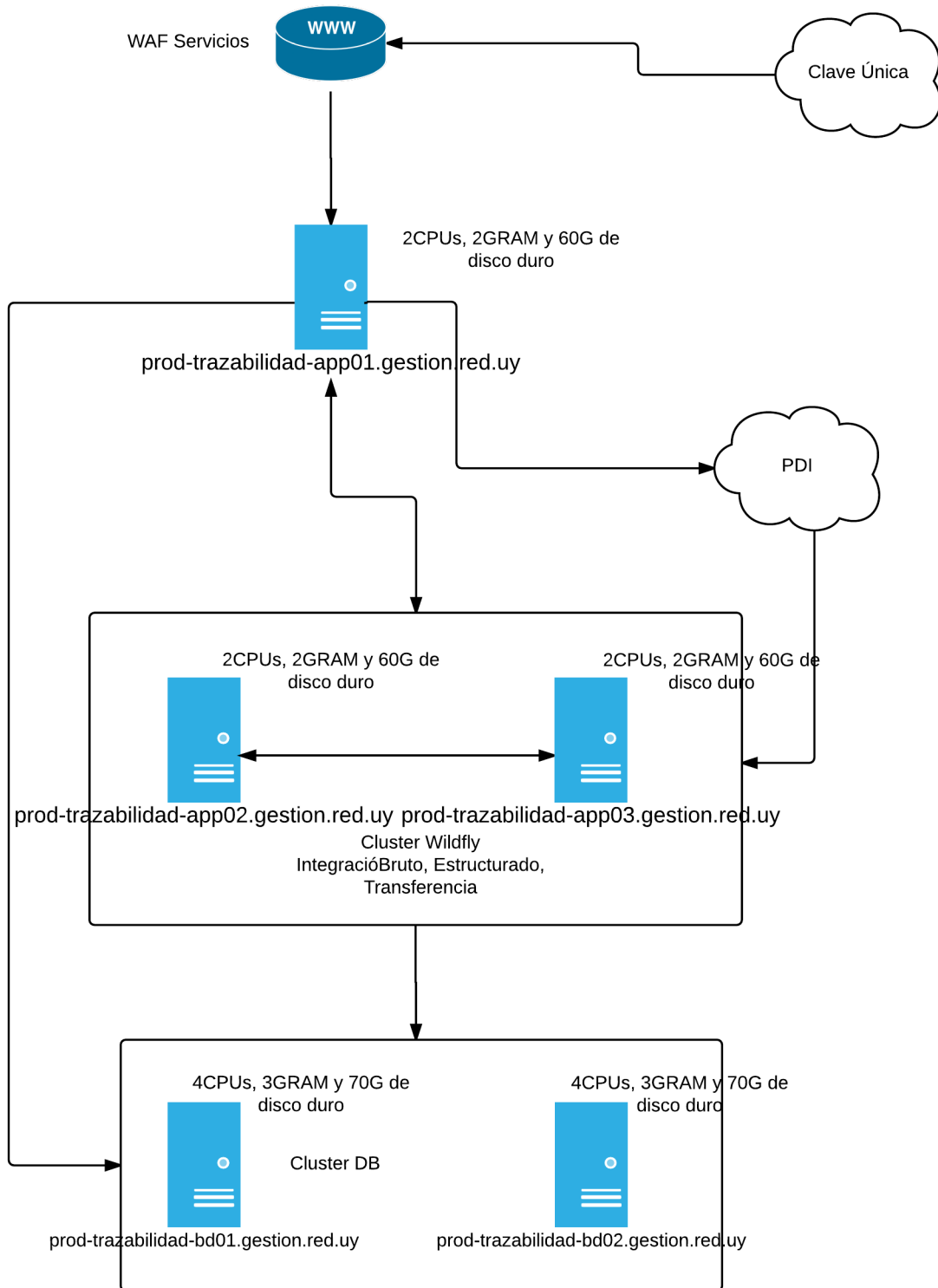
# 5 Vista Física

## 5.1 Arquitectura de deployment tipo para la aplicación (vista de componentes)

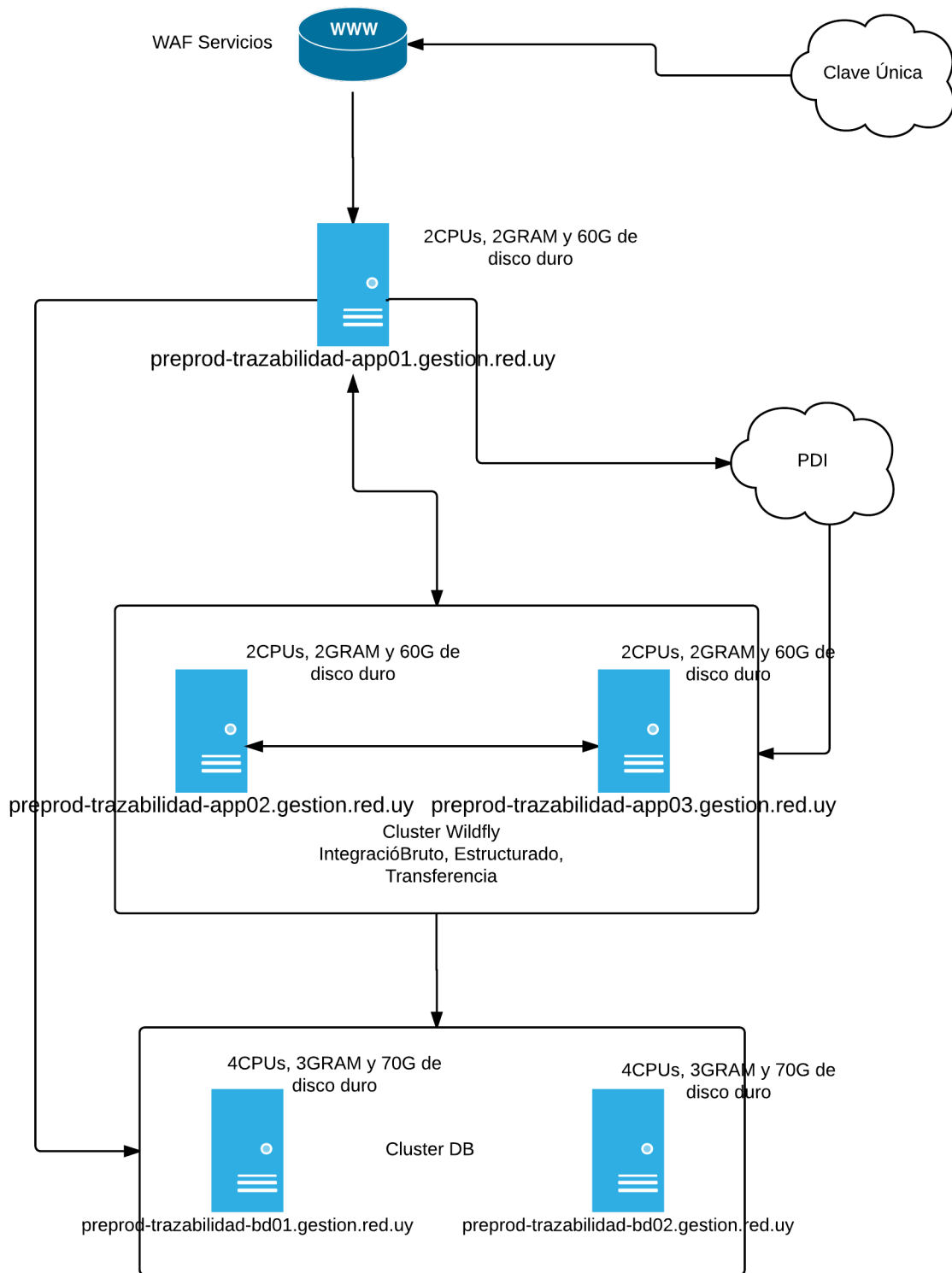


## 5.2 Arquitectura de deployment para la aplicación (vista física)

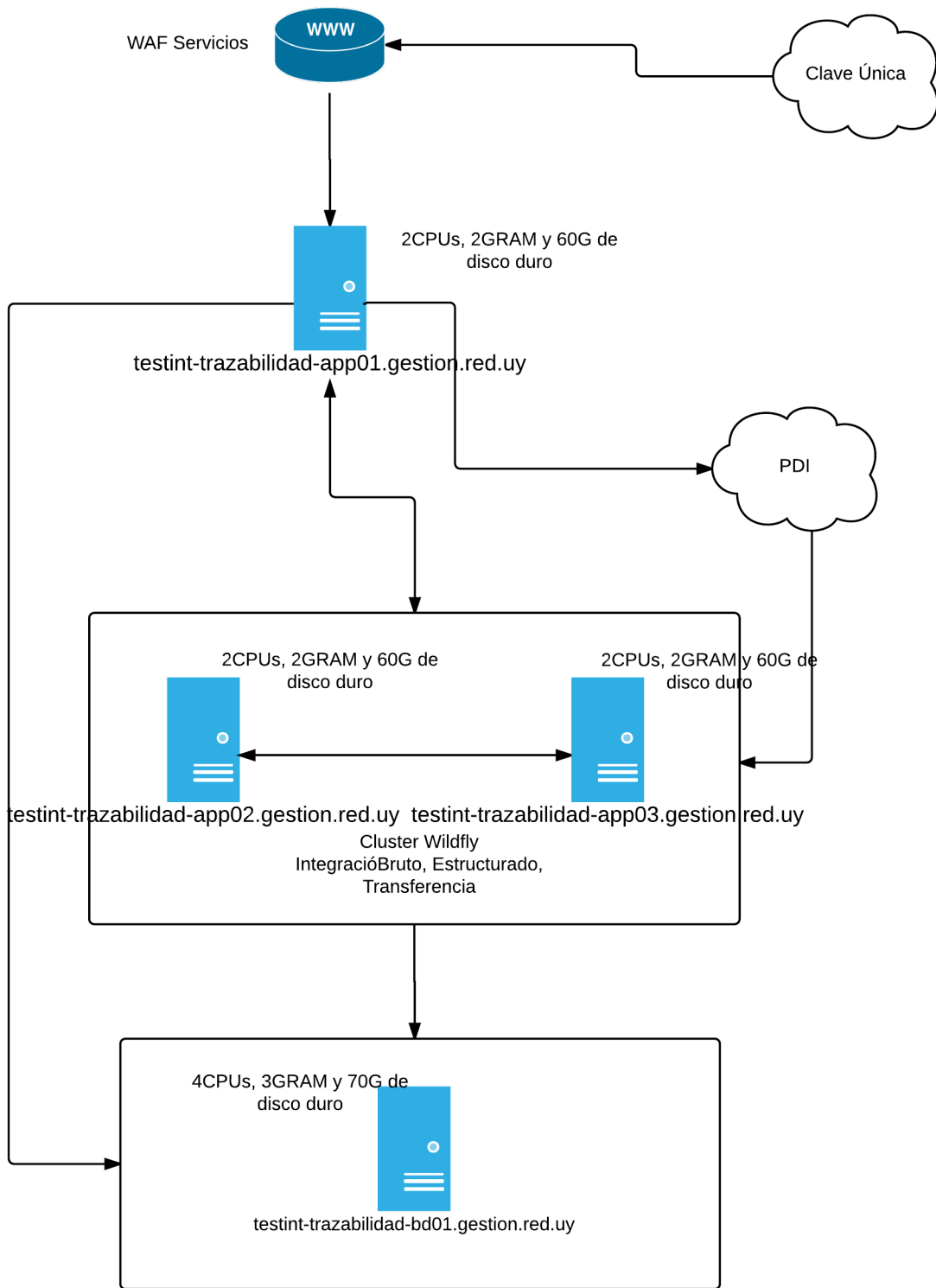
### 5.2.1 Producción



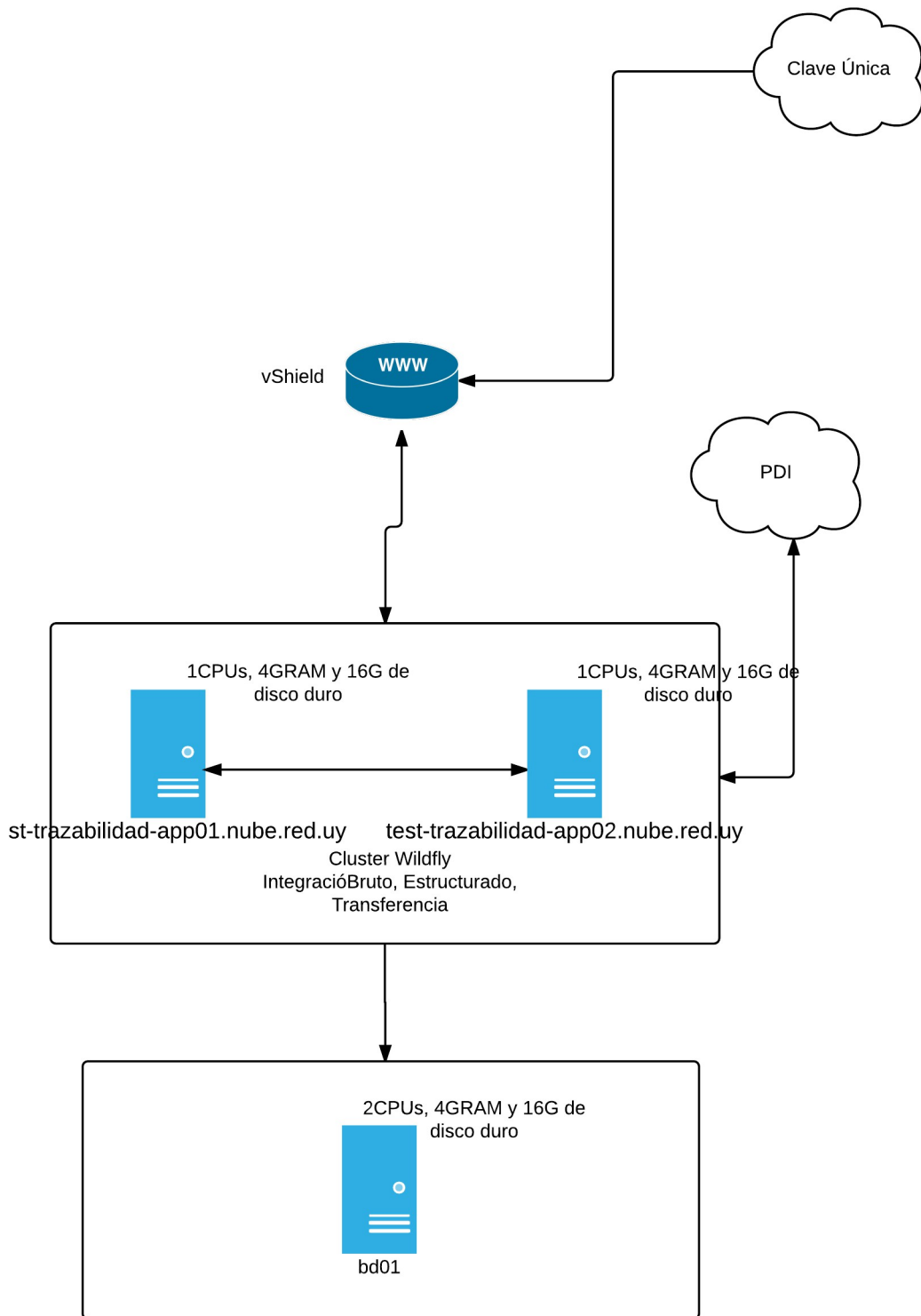
## 5.2.2 Pre-Producción



### 5.2.3 Testing Integración



### 5.2.4 Testing Funcional



## 5.3 Conclusiones

Toda la arquitectura de la aplicación se definió para que cada uno de los componentes principales no tuviera dependencias entre sí. De hecho, se pensó en un modelo incremental, en el cual se puede comenzar con todas las aplicaciones ejecutando en el mismo “stack” de servidores, y en caso de aumentar la carga de las aplicaciones separarlos.

Los elementos de integración entre las aplicaciones itramites-bruto / itramites-transferencia e itramites-estructurado son conectarse a las mismas Bases de Datos (hoy en día relacionales).

Itramites-bruto e itramites-transferencia deben compartir la BD itramitesbrutodb

itramites-estructurado e itramites-transferencia deben compartir la BD itramitesestructuradodb

La única dependencia en tiempo de ejecución existe entre la aplicación de frontEnd, y los servicios de consulta (que están dentro de itramites-estructurado). Dicha dependencia se podría liberar en un futuro, ejecutando via Web Services a través de PDI (penalizando considerablemente los tiempos de respuesta del backOffice). Por eso se decidió que al momento la mejor decisión es conectar ambas aplicaciones a través de EJBs remotos. Pueden estar igualmente desplegadas en servidores independientes (con conectividad entre sí).

## 6 Vista de Procesos

### 6.1 Proceso de transferencia:

Este proceso está configurado para correr en los segundos 12,24,36, y 48 de cada minuto. La librería que maneja la ejecución de tareas programadas (quartz) se encarga de asignar dinámicamente las tareas a una de las instancias de servidores disponibles.

Existen dos modalidades de funcionamiento, que modifican fundamentalmente la capacidad que tiene el proceso de ser reintentado.

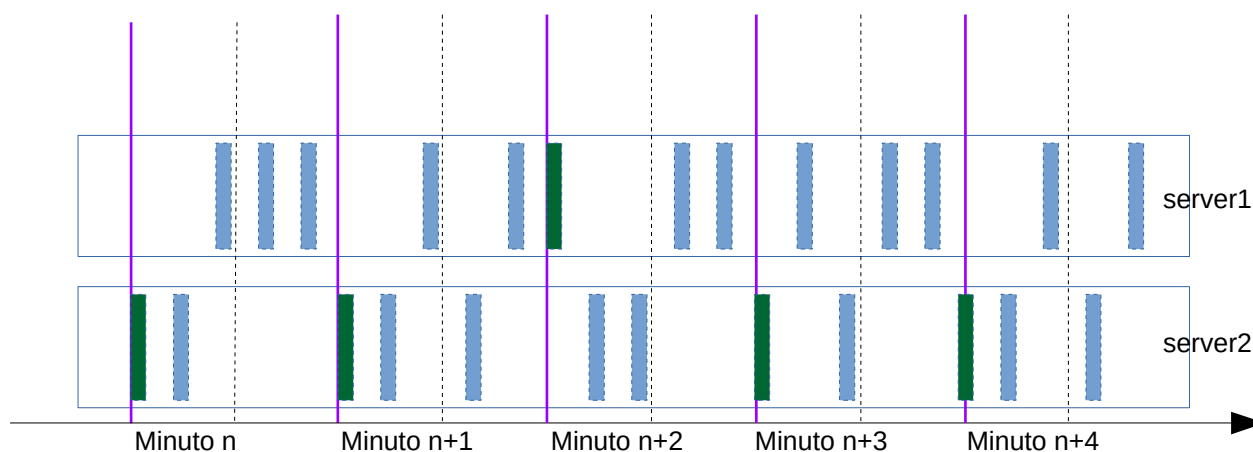
La modalidad convencional se basa en almacenar una máquina de estados en la base bruta (tabla EstadoTraza), que controla cuándo una traza en particular fué transferida. La gestión del proceso la lleva adelante la clase ProcesadorTrazaBeanImpl, y es ejecutado por la tarea TareaEjecutarTransferenciaTrazas en caso de corresponder.

La modalidad persistente, se basa en 2 sub-procesos. Un sub-proceso que notifica en la base estructurada la existencia de una traza candidata a ser transferida (tabla DefinicionObjetoTraza) invocado por la tarea TareaCrearDefinicionObjetosTraza, y un sub-proceso que crea bloques de trazas y los deja prontos para ser transferidos (tablas BloqueTransferencia y TrazaBloqueTransferencia) invocado por la tarea TareaEjecutarCreacionBloques.



Los bloques pueden ser creados a partir de trazas nuevas, o trazas que dieron error en intentos anteriores de transferencia. La gestión del proceso en su conjunto la lleva adelante la clase `ProcesadorPersistenteTrazaBeanImpl`, y es ejecutado por la tarea `TareaEjecutarTransferenciaTrazas` en caso de corresponder.

A continuación se explica gráficamente en el tiempo cómo funcionarían las ventanas de tiempo donde ejecuta cada proceso de transferencia (celeste) y los procesos de validación (verde).



### 6.1.1 Funcionamiento del proceso de transferencia (modalidad convencional)

A continuación se explica el paso a paso el proceso de transferencia, (para la modalidad convencional), junto con un diagrama de interacción.

- Cuando se invoca a la tarea `TareaEjecutarTransferenciaTrazas`, se obtienen los registros que estén en la tabla `estadotraza` con estado "Pendiente" y los actualiza a estado = "En\_Proceso". Esta consulta es ordenada por tipo, id y tiene un tope configurable de la cantidad de elementos que recuperará
- Por cada registro busca el `objetotraza` correspondiente por id.
- Se realizan validaciones por registro

- Se inserta el registro en la tabla línea o cabezal dependiendo del tipo de entidad de la tabla objetotraza
- Actualiza el estado a “Procesada” de la tabla estadotraza. Si existe algún error al insertar el registro en las tablas línea o cabezal, se actualiza la tabla estadotraza con estado “Error”

Posibles estados de estadoTraza:

- PENDIENTE = 0
- EN\_PROCESO = 1
- PROCESADA = 2
- ERROR = 3

## 6.1.2 Funcionamiento del proceso de transferencia (modalidad persistente)

A continuación se explica el paso a paso el proceso de transferencia, (para la modalidad convencional), junto con un diagrama de interacción.

- Proceso 1 (TareaCrearDefinicionObjetosTraza): Se toman todas las trazas que existan en la tabla ObjetoTraza (bruto) y no existan en DefinicionObjetoTraza (estructurado), y las inserta
- Proceso 2 (TareaCrearBloquesTraza): En caso de encontrar trazas que no sean parte de ningún bloque, o bien formaron parte de un bloque y no fueron transferidas exitosamente, y crea un bloque con dichas trazas (BloqueTransferencia) y una tabla de relación entre el bloque y la traza (TrazaBloqueTransferencia), donde se mantiene la máquina de estados.
- Cuando se invoca a la tarea TareaEjecutarTransferenciaTrazas, se verifica si hay un bloque a ser transferido, y se coloca el bloque en estado “En\_Proceso”.
- Por cada registro busca el objetotraza correspondiente por id
- Se realizan validaciones por registro
- Se inserta el registro en la tabla línea o cabezal dependiendo del tipo de entidad de la tabla objetotraza
- Actualiza el estado a “Procesada” de la tabla TrazaBloqueTransferencia, y dependiendo del resultado del bloque en su conjunto se actualiza el estado del bloque. Si existe algún error al insertar el registro en las tablas línea o cabezal, se actualiza la tabla TrazaBloqueTransferencia con estado “Error”

Posibles estados de TrazaBloqueTransferencia:

- PENDIENTE = 0
- EN\_PROCESO = 1
- PROCESADA = 2
- ERROR = 3

Posibles estados de BloqueTransferencia:

- PENDIENTE = 0
- EN\_PROCESO = 1

- PROCESADA = 2
- ERROR = 3
- ELEMENTOS\_CON\_ERROR = 4
- CREADA = 5

## 6.2 Proceso de validación

### 6.2.1 Validaciones disponibles

- Validadores de registro
  - ValidadorTrazaTardia: valida que la fecha del organismo tenga menos de una tiempo -configurable- de diferencia con respecto a la fecha de registrada por PDI al momento de capturar la traza.
  - ValidadorCamposRequeridos: valida que la traza contenga un conjunto definido (y configurable) de atributos. Se utiliza para atributos que no se quiere forzar por contrato que estén allí, pero sí se quiere identificar cuando no están presentes
- Validadores de Traza Completa
  - ValidadorInicioFin: valida que la traza tenga una línea de fin
  - ValidadorFechaFinConResto: valida que la fecha de organismo de la traza de fin sea mayor a las fechas del resto de las líneas
  - ValidadorSecuencialidadPasos: valida que no hayan huecos en los números de pasos, siempre y cuando haya una traza de fin
  - ValidadorPasosRepetidos: valida que no hayan números de pasos repetidos, siempre y cuando haya una traza de fin

### 6.2.2 Funcionamiento del proceso de validación

Este proceso está configurado para correr todos los minutos, en el segundo 0 (reservado para validaciones). En este proceso se aplica la misma ventana de tiempo que se explica en el punto anterior.

A continuación se explica el paso a paso del mismo y un diagrama de interacción.

1. Obtiene los registros que estén en la tabla cabezal con estadoConsultable = "Pendiente".
2. Obtiene la lista de validadores a ejecutar
3. Por cada registro "Cabezal" busca las líneas asociadas (\*)
4. Se realizan validaciones por cada traza completa, se actualiza el estadoConsultable dependiendo del resultado de la validación.
5. Si no existe un registro de esta validación en la tabla Validadores, se

guarda el resultado de la validación y se actualiza el estadoConsultable a “Finalizado” (en caso de no haber errores)

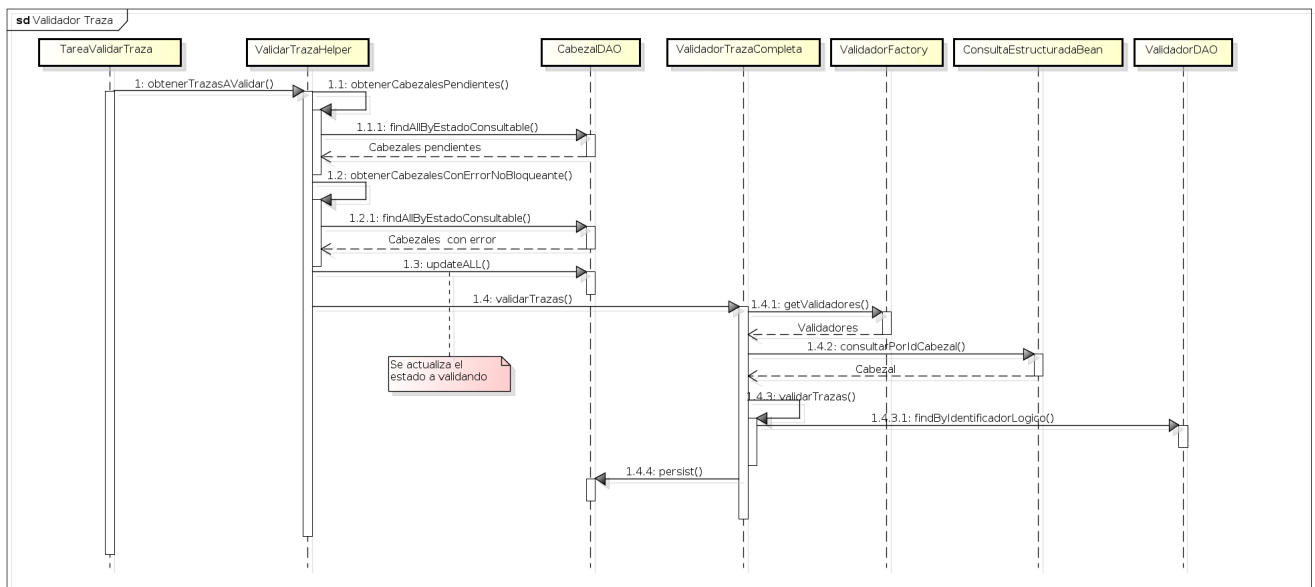
6. Actualiza un timestamp de última validación para cada uno de los cabezales que se validaron

(\*) - En la selección de las trazas a validar, es que radica la lógica de configuración de la periodicidad de validadores.

La lógica es la siguiente:

1. Todas las noches corre un proceso de determinación de exclusiones. Esto es, todos los ids procesos que tengan configurado en forma global (para todos sus validadores) una periodicidad mayor a la diferencia de tiempo con la última ejecución de una validación para ese id proceso, se registran para ser “excluidos” hasta las 0 horas del día siguiente.
  1. El mismo proceso ocurre en forma particular para un par idProceso-idValidador particular, cuando no haya pasado el tiempo marcado por la periodicidad desde su última ejecución (marcado en la tabla registroactualizacionvalidador)
2. Una vez que se “desactivan” los validadores para todo un trámite, o para una combinación idProceso-idValidador, el proceso de validación fluye normalmente, tomando en cuenta para cada cabezal que:
  1. No esté dentro de los trámites excluidos para dicho día
  2. Se considerarán solamente los validadores que no hayan quedado excluidos para el día de la fecha
  3. El tiempo desde la última validación de esa traza en particular (timestampValidacion) no sea menor a la periodicidad marcada para esa combinación idProceso-idValidador.

### 6.2.3 Diagrama de Secuencia:



powered by Astah

### 6.3 Proceso de reporte de calidad:

Este proceso está configurado para correr a la hora 13 de lunes a viernes.

A continuación se explica el paso a paso del mismo.

1. Obtiene los registros de la tabla fallos y de la tabla errorLog.
2. Se genera un excel con la información obtenida.
3. Por cada registro "Cabezal" busca las líneas asociadas.
4. Se genera el correo y se envía a la lista de destinatarios.

## 6.4 Proceso de generación de estadísticas:

Este proceso está configurado para correr a la hora 1am del primer día de cada mes.

A continuación se explica el paso a paso del mismo.

1. Se actualizan los periodos en la base. Para esto se verifica que periodos de los últimos 12 meses no existen en la base y los carga. Toma en cuenta los periodos de acuerdo a los tipos de periodos configurados.
2. Identifica la combinación organismo-tramite que tienen trazas registradas.
3. Por cada tipo de reporte (Ej: duración promedio, in-out, etc) se genera el valor de la estadística y se registra en la tabla valoresMetricas.

**Nota:** Al ejecutarse el primer día del mes, los datos registrados van a ser correspondientes al último día del mes anterior. En caso de necesitarse la información actualizada a un día específico, existe un servicio que ejecuta la lógica del proceso.

## 6.5 Proceso de carga de guía de trámite:

Este proceso está configurado para correr a la hora 1am todos los días.

Pre-condición: debe estar la base cargada la primera vez, este proceso es de actualización.

A continuación se explica el paso a paso del mismo.

1. Se toma la última fecha de actualización de la tabla EjecucionGuia y con la misma se invoca al servicio de Guía de Trámites. El mismo devuelve las altas, bajas y modificaciones desde la fecha con la cual se invoca a la fecha actual.
2. Se actualizan los idProceso en la base con los datos obtenidos en el paso anterior.

**Nota:** El borrado de los idProcesos en la base de datos es lógico para no afectar los datos existentes. En caso de necesitarse la información actualizada en un momento específico del día, existe un servicio que ejecuta la lógica del proceso.

# 7 Configuraciones

## Permisos

**LECTURA = 0**  
**LECTURA\_ESCRITURA = 1**  
**ESCRITURA = 2**  
**NINGUNO = 3**

## EstadoProceso

**INICIO = 1**  
**EN\_EJECUCION = 2**  
**FINALIZADO = 3**  
**CERRADO = 4**

## EstadoValidacion

**PENDIENTE = 0**  
**ERROR = 1**  
**CORRECTO = 2**

## EstadoConsultable

**PENDIENTE = 0**  
**VALIDANDO = 1**  
**BLOQUEANTE = 2**  
**EN\_PROCESO("ERROR NO BLOQUEANTE") = 3**  
**FINALIZADO = 4**

## InicioAsistidoProceso

**NO = 0**  
**SI = 1**

## CanalDelInicio

**WEB\_PC = 1**  
**WEB\_MOVIL = 2**  
**PRESENCIAL = 3**  
**REDES\_COBRANZAS = 4**  
**PAC("PUNTO ATENCION CIUDADANA") = 5**  
**TELEFONICO = 6**  
**CORREO\_ELECTRONICO = 7**

## TipoProceso

**TRAMITE = 1**  
**COMPRA = 2**  
**LICITACION = 3**  
**RECLAMO = 4**  
**SERVICIO = 5**  
**EXPEDIENTE = 6**  
**PROCESO\_GENERICO = 7**

## TipoRegistroTrazabilidad

**TRANSFERENCIA = 1**

**SUBPROCESO = 2**

**COMUN = 3**

## 8 GLOSARIO

Traza: Registro de un trámite.

Cabecal: El trámite se divide en dos secciones. La sección que contiene la información descriptiva del trámite se le llama cabecal.

Línea: Es la sección del trámite que guarda información adicional del trámite.

Paso: Secuencial que describe el orden de las líneas de un trámite.

Id Transacción: identificador del trámite.

Rol: Rol de PDI usado para validar.

GUID: Identificador del trámite para el ciudadano.

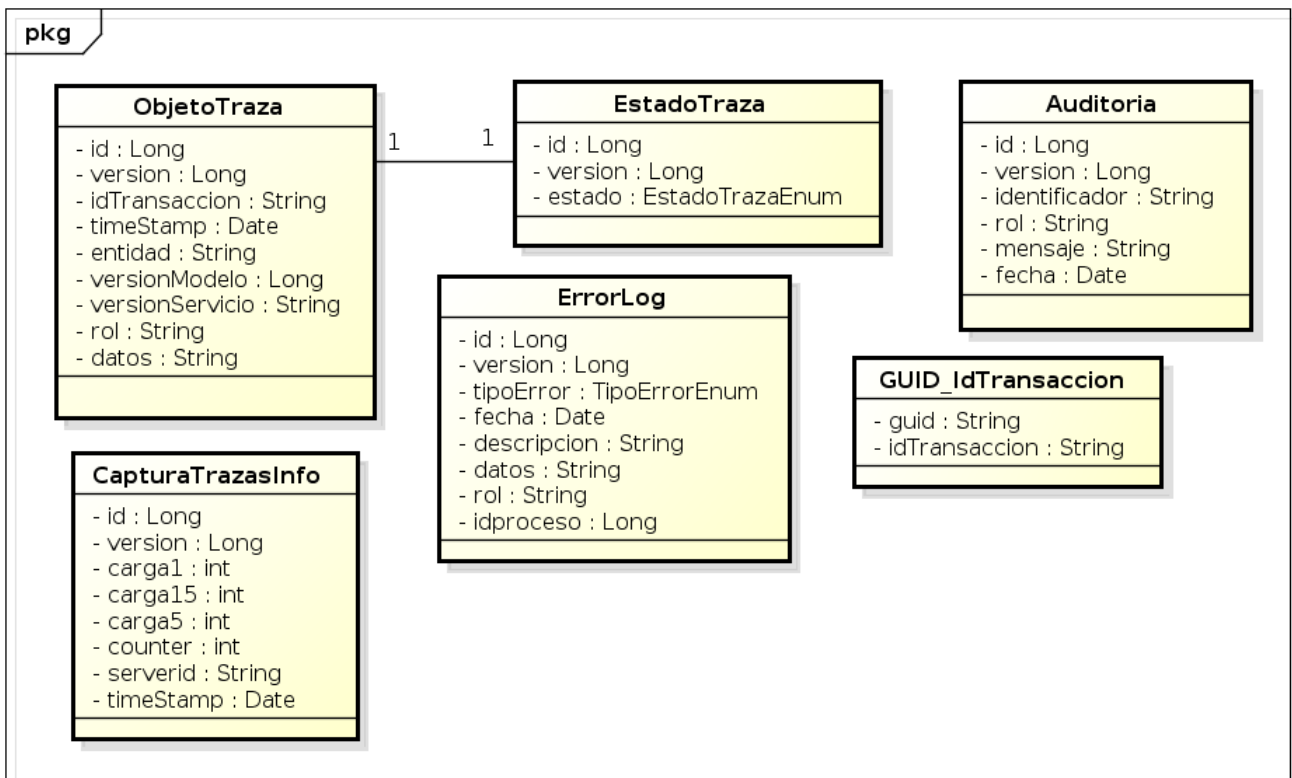
Clave Unica: Servicio federado de autenticación para el Estado (Control de Acceso). Implementación actual: Coesys



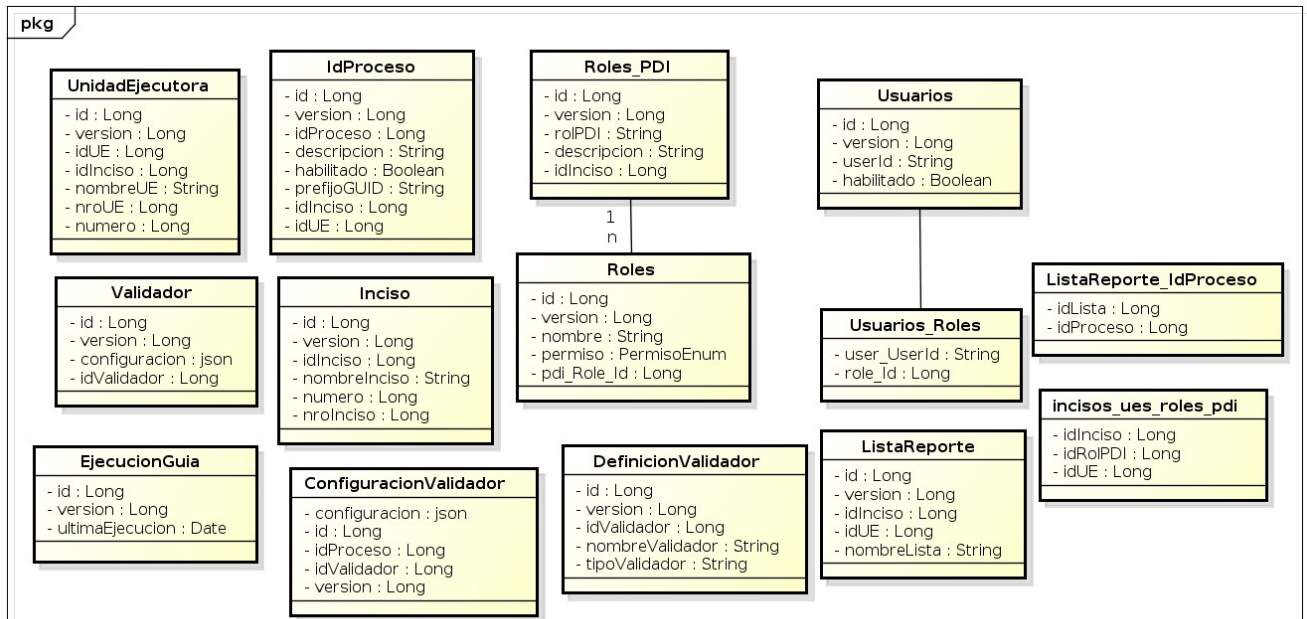
## 9 Anexo: Modelo de datos

A continuación se muestran los modelos de datos de los diferentes módulos.

### 9.1 Modelo de datos “Bruto”

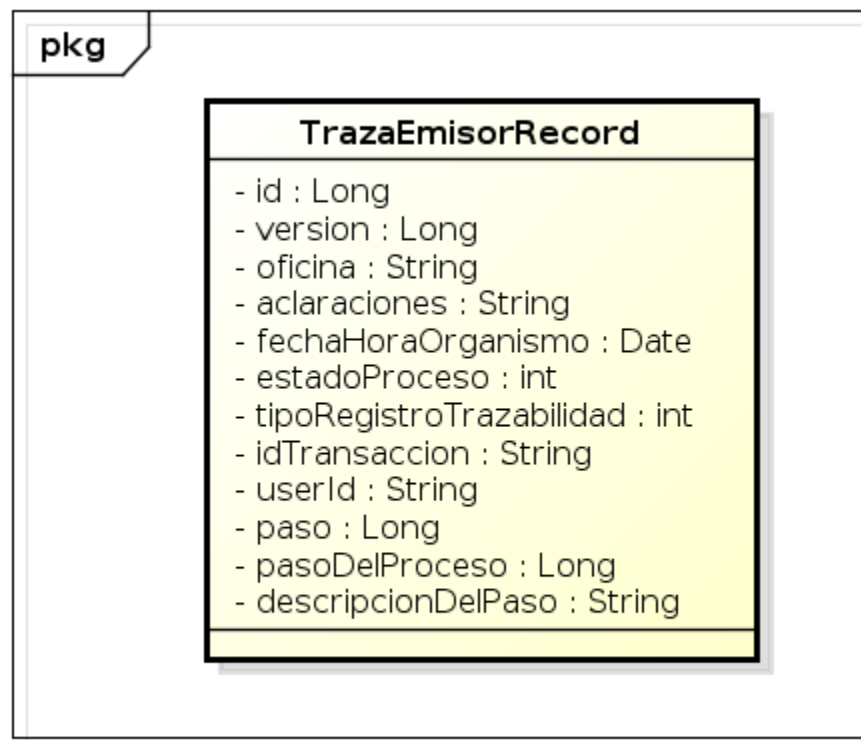


## 9.2 Modelo de datos “Integración”



powered by Astah

## 9.3 Modelo de datos “Frontend”



powered by Astah

## 9.4 Modelo de datos “Estructurado”

