[Ficha Simplificada]

* Análisis y Diseño de la aplicación y estructura del sistema.
* Diagrama de Arquitectura de la

Herramienta

* Diagrama físico de la base de datos.

|  |
| --- |
|  |
| DIMUTZA S.A | HURTADO 519 Y ESMERALDAS 3ER. PISO OFC 301  1-0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revisión** | **Fecha** | **Modificación Realizada** | **Realizado Por:** |
| 0 | 15 de Noviembre del  2015 | Versión Inicial. | Daniel Chiang |
| 1 | 17 de Diciembre del 2015 | Versión de Revisión | Daniel Chiang |
| 2 | 20 de Diciembre de 2015 | Versión Final | Daniel Chiang |

**TABLA DE CONTENIDO**

**GLOSARIO DE TÉRMINOS .............................................................................................................. 1-4**

1. **INTRODUCCIÓN ......................................................................................................................... 1-5**
   1. OBJETO DEL DOCUMENTO. ......................................................................................................... 1-5
   2. EQUIPO DE TRABAJO. ................................................................................................................. 1-5
   3. MATERIAL REVISADO. ................................................................................................................. 1-6
2. **ARQUITECTURA ........................................................................................................................ 2-6**
   1. ARQUITECTURA LÓGICA ......................................................................................................... 2-6

# 2.2 DOMAIN-DRIVEN DESIGN - JEE 7 ........................................................................................... 2-7

*2.2.1* *Capa de Interfaz de Usuario: .......................................................................................... 2-7*

*2.2.2* *Capa de Aplicación: ........................................................................................................ 2-7*

*2.2.3* *Capa de Dominio ............................................................................................................. 2-7*

*2.2.4* *Capa de Infraestructura: ................................................................................................. 2-8*

2.3 ARQUITECTURA FÍSICA ........................................................................................................... 2-9

1. **MEJORES PRÁCTICAS ........................................................................................................... 3-10**
   1. SONARQUBE .......................................................................................................................... 3-10
   2. GIT ......................................................................................................................................... 3-11
   3. JENKINS ................................................................................................................................. 3-11
   4. ANT ........................................................................................................................................ 3-12

# 3.5 PRUEBAS DE RENDIMIENTO ................................................................................................... 3-12

1. **CALIDAD .................................................................................................................................... 4-12**
2. **DOCUMENTACIÓN .................................................................................................................. 5-12**
3. **ANEXOS ...................................................................................................................................... 6-13**

6.1 MODELO FÍSICO ..................................................................................................................... 6-13

# 6.2 DIAGRAMA DE CLASES .......................................................................................................... 6-14

*6.2.1* *Capa de Dominio ........................................................................................................... 6-14*

*6.2.2* *Capa de Infraestructura ................................................................................................ 6-17*

*6.2.3* *Capa de Aplicación ....................................................................................................... 6-20* *6.2.4* *Capa de UI .................................................................................................................... 6-21*

# **GLOSARIO DE TÉRMINOS**

**Definiciones:**

Dentro del contexto del presente documento se definen los siguientes términos:

**Dinardap:** Dirección Nacional de Registro de Datos Públicos.

**Sinardap:** Sistema Nacional de Registro de Datos Públicos.

**Interoperabilidad:** Servicio Web creado por el Dinardap para proveer datos ciudadanos a las demás instituciones a través de la creación de paquetes de datos por institución.

**Paquetes de datos:** Agrupa todos los datos de un ciudadano en una institución que pueden ser consultados a través del servicio web de Interoperabilidad.

**Interoperador:** Base de datos que contiene información acerca de los paquetes de datos existentes en el servicio web de Interoperabilidad.

**JEE 7:** Es la versión más reciente y estable de un conjunto de especificaciones, funcionalidades y estándares orientadas al desarrollo de aplicaciones empresariales, escalables, portables, y móviles utilizando el lenguaje de programación Java.

**WildFly**: Es un servidor de aplicaciones JEE y también un servidor web HTTP creado por RedHat, para Unix, Linux, Microsoft Windows, y otras plataformas.

**PrimeFaces:** es un framework que utiliza la tecnología JavaServer Faces (JSF) para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

**EJB:** Enterprise JavaBeans proporcionan un modelo de componentes distribuidos estándar del lado del servidor que le permite al programador abstraerse de los problemas generales de una aplicación empresarial para centrarse en el desarrollo de la lógica de negocio en sí.

**CDI:** Contexts and Dependency Injection for Java EE, es un conjunto de servicios contextuales que facilitan a los desarrolladores utilizar Enterprise JavaBeans junto a los JavaServer Faces en aplicaciones web.

**JPA:** Es una herramienta del estándar de Java que realiza el Mapeo objeto-relacional (ORM) entre una base de datos relacional tradicional y el modelo de objetos de una aplicación.

**AJAX:** Es una tecnología asíncrona, en que los datos se solicitan mediante Java al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página maquetando las peticiones y respuestas en formato XML.

**XML**: Es una lenguaje de maquetado de datos. Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

**DDD:** Domain-Driven Design es un enfoque para el desarrollo de software, no es una tecnología ni una metodología. Provee una estructura de prácticas y terminologías para tomar decisiones de diseño que enfoquen y aceleren la creación y mantenimiento de dominios complejos en los proyectos de software.

## 1. INTRODUCCIÓN

**1.1 Objeto del Documento.**

Presenta la descripción de cada uno de los componentes de la arquitectura para la creación del sistema de Fichas Simplificadas.

Una Arquitectura de [Software,](http://es.wikipedia.org/wiki/Software) consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del [software](http://es.wikipedia.org/wiki/Software) de un [sistema de información.](http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n)

La arquitectura de la aplicación permite definir la organización interna de la misma estableciendo una funcionalidad por capas y bien definida. Permiten tener mejor control de la ubicación de las reglas de negocio y por lo tanto facilita las tareas de mantenimiento de las mismas.

**1.2 Equipo de Trabajo.**

Las personas que directa o indirectamente intervinieron fueron las siguientes:

|  |  |
| --- | --- |
| **Nombre** | **Cargo** |
| Andres Revelo | Analista de Infraestructura |
| Alejandra Villavicencio | Directora de Gestión y Registro (Administradora del Contrato) |
| Fernando Moya Leimberg | Coordinador de Infraestructura y Seguridad Informática |
| Tatiana Moncayo | Asesora de Coordinación de Infraestructura y Seguridad Informática |
| Carina Escobar | Directora de Tecnología y Desarrollo |

**Por parte de Dimutza participaron:**

* Eddy Ponguillo
* Daniel Chiang
* Edward Holguín
* Marisol Martínez
* Efrén Villamagua

**1.3 Material Revisado.**

* Estándares y herramientas para el desarrollo de aplicaciones Java en la Dinardap.
* Estándares y herramientas para el desarrollo de aplicaciones Java de Dimutza.

## 2. ARQUITECTURA

### 2.1 Arquitectura Lógica

El patrón de arquitectura elegido para la creación del sistema Ficha Simplificada es el de niveles de abstracción o capas que recomienda el enfoque Domain-Drive Design; este modelo de arquitectura se encuentra basado en la especificación JEE 7 y en patrones de diseño ampliamente usados, así como estándares abiertos.

Interfaz

de Usuario

Applicación

Dominio

(

Lógica de

Negocio

)

Infraestructura

Diseño Guiado por el Dominio

Para implementar la arquitectura JEE se usará el servidor de aplicaciones WildFly 9.0.2 de RedHat que está certificado para soportar JEE 7 que este a la disposición del Dinardap.

### 2.2 Domain-Driven Design - JEE 7

**2.2.1 Capa de Interfaz de Usuario:**

La capa de interfaz de usuario o capa de presentación, se utilizará la especificación de JSF2.2 con Primefaces 5.3, permitiendo crear un ambiente Ajax de manera fácil, rápida y limpia.

Esta capa se encuentra compuesta por:

***Control****.-* Almacena los backing bean de las paginas JSF.

***Web UIs****.-* Contiene las paginas JSF de cada módulo.

**2.2.2 Capa de Aplicación:**

La capa de aplicación es la responsable de manejar el flujo de la aplicación. Contiene las interfaces de servicios de negocio que utilizarán EJB 3.1 para la gestión (conexión) de los JavaBeans de negocio, y el manejo transaccional de la aplicación. Se encarga de la coordinación de los objetos de dominio. Esta capa se puede extender para implementar servicios web mediante JAX-RS o JAX-WS que permita el acoplamiento desde otros sistemas que requieran interactuar con Ficha Simplificada.

Esta capa se encuentra compuesta por:

***Services***.- Usando EJB 3.1 implementa JavaBeans que contienen los métodos de negocio expuestos para su utilización desde la capa de presentación. Maneja las transacciones de la aplicación.

**2.2.3 Capa de Dominio**

La capa de dominio o lógica de negocio contiene la implementación del ORM usando JPA. También contiene las validaciones y reglas propias de negocio

Esta capa se encuentra compuesta por:

***Data***.- JPA Entities que contienen además la lógica de negocio.

***Repository***.- Interfaces dispuestas por el dominio para que la infraestructura pueda consumir sus objetos.

**2.2.4 Capa de Infraestructura:**

La capa de infraestructura contiene los mecanismos necesarios para interactuar con las bases de datos y servicios que provee la Dinardap. Se utiliza JPA como herramienta de mapeo de atributos entre una [base de datos](http://es.wikipedia.org/wiki/Base_de_datos) relacional tradicional y el modelo de [objetos](http://es.wikipedia.org/wiki/Objetos_(programaci%C3%B3n_orientada_a_objetos)) de la aplicación y JAX-WS para el consumo de los servicios de la institución.

Esta capa se encuentra compuesta por:

***Repository***.- JPA Entity Managers encargados de la persistencia de datos Sinardap e Interoperador.

***Security***.- Autenticación LDAP a través de consumos de servicios mediante JAX-WS client.

***Service***.- JAX-WS clients que permite interactuar con el servicio web de Interoperabilidad para la obtención de datos ciudadanos.



### 2.3 Arquitectura Física

A nivel físico el sistema cuenta con un esquema distribuido entre múltiples plataformas:

***Sinardap***.- Base de datos que contiene información acerca de las instituciones, sus usuarios y su rol en las mismas.

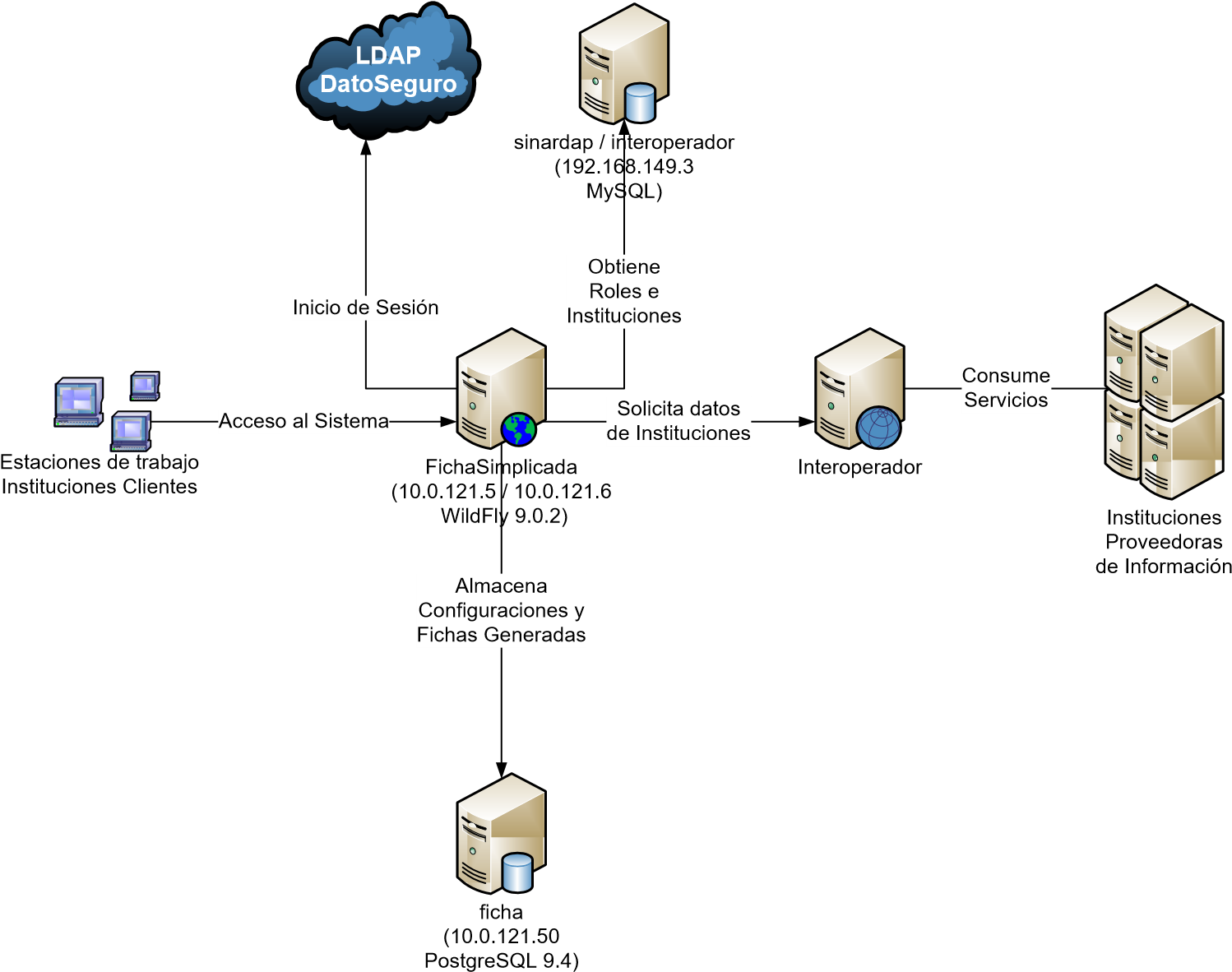
***Interoperabilidad***.- Servicio Web creado por el Dinardap para proveer datos ciudadanos a las demás instituciones a través de la creación de paquetes de datos por institución.

***Interoperador***.-Base de datos que contiene información acerca de los paquetes de datos existentes en el servicio web de Interoperabilidad.

***WildFly***.- Servidor de aplicaciones JEE instalado en dos servidores diferentes para distribución de la carga de trabajo y alta disponibilidad.

***Ficha***.- Servidor que cuenta con el motor de base de datos PostgreSQL 9.4 y sirve para almacenar la data y configuraciones propias del sistema.

***LDAP****.-* Directorio Activo consumido a través de un servicio web dispuesto por la Dinardap que permite autenticar a los usuarios del sistema para su inicio de sesión.



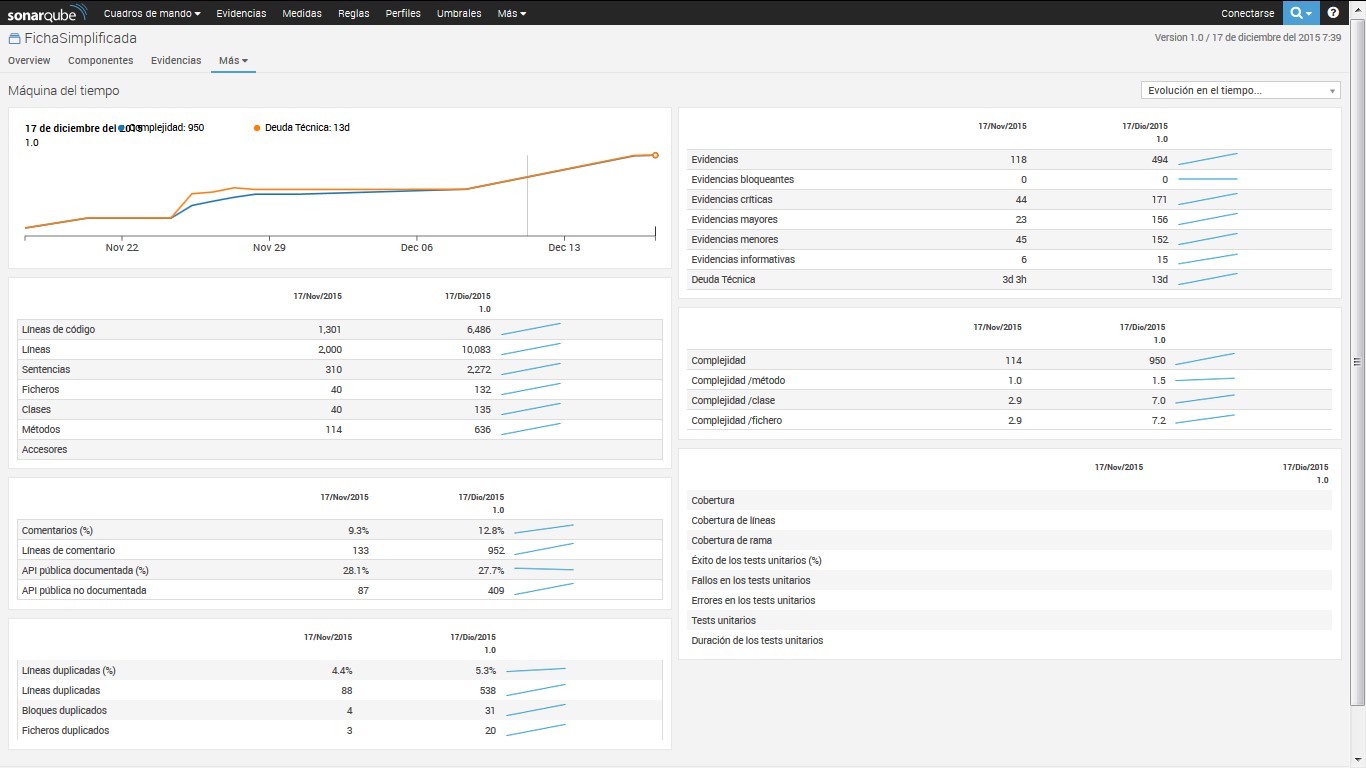
## 3. MEJORES PRÁCTICAS

### 3.1 SonarQube

SonarQube es una herramienta que evalúa el código fuente de nuestro sistema para garantizar que cumpla una serie de reglas que nos ayuden a obtener un código más limpio, elegante, y sencillo de mantener.

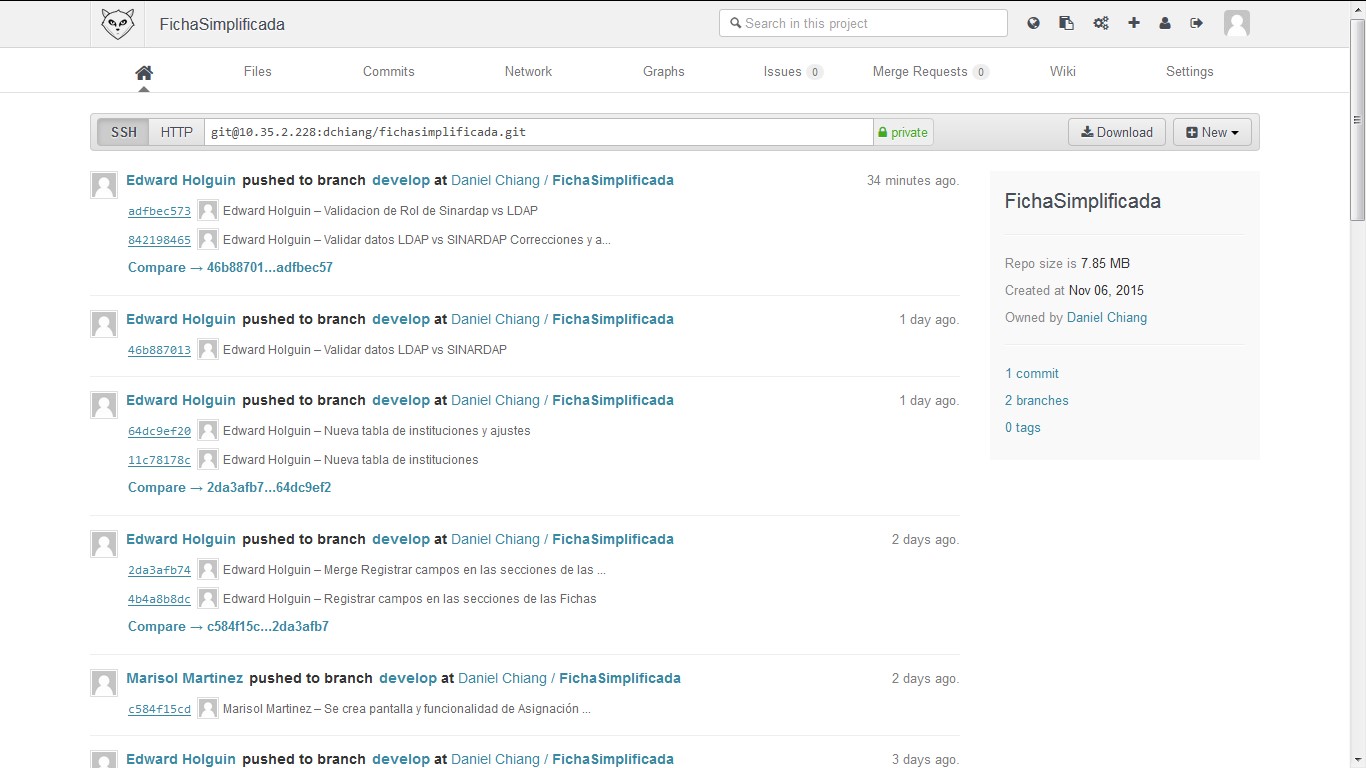
SonarQube busca en el código Java posibles problemas en potencia como pueden ser:

* **Posibles bugs.** Sentencias *try*/*catch/finally/switch* vacías.
* **Código muerto.** Variables locales, funciones sin implementación, parámetros y métodos privados que no se usan.
* **Código no óptimo.** Mal uso de String/StringBuffer/StringBuilder.
* **Expresiones excesivamente complicadas**. Funciones muy extensas, sentencias *if* innecesarias, implementaciones con bucles *while*.
* **Código duplicado.** Código copiado y pegado sin valor agregado.
* **Ruido.** Funciones comentadas que tienden a complicar la interpretación humana del código.
* **Cumplimiento de estándares de programación y comentarios.** Evalúa que se sigan los estándares y convenciones de programación del lenguaje Java establecidos por Sun Microsystems y Oracle.
* **Deuda Técnica.** Presenta las evidencias de los problemas encontrados y el tiempo estimado que tomará resolverlos.
* **Máquina de Tiempo.** Muestra gráficas con el histórico de los defectos en el transcurso del tiempo y la evolución del proyecto.



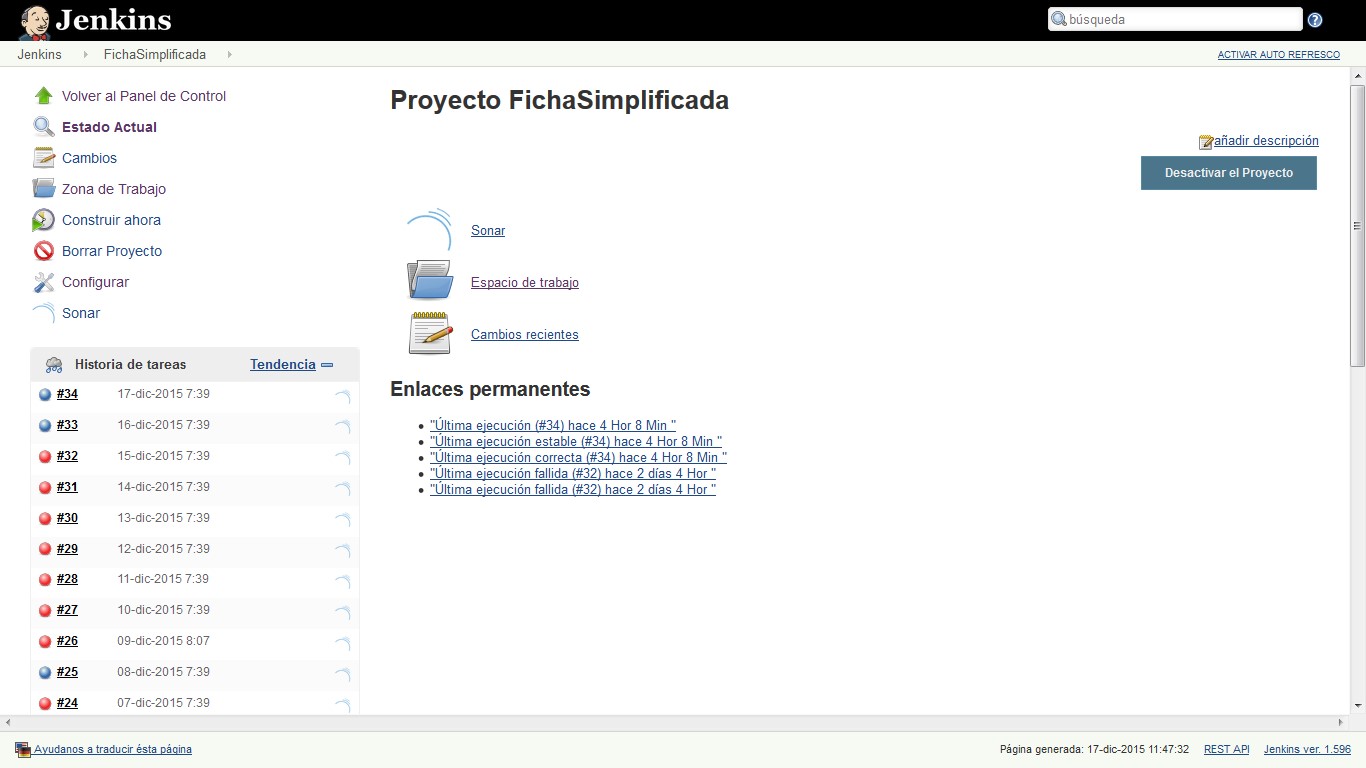
### 3.2 Git

Herramienta de versionamiento de código que permite llevar el control y auditoría de los cambios que realiza cada desarrollador, y cambiar fácilmente de versión para contar siempre con código estable listo para su uso en producción.



### 3.3 Jenkins

Servidor líder de código abierto para integración continua. Toma el código fuente del repositorio de versionamiento para realizar la compilación y despliegue del mismo. Detecta de forma temprana el código que rompe la compilación de un proyecto. Es de ejecución manual o automática mediante la creación de una agenda de trabajo.



### 3.4 Ant

Invocado desde Jenkins sirve para automatizar las tareas de despliegue (deployment) de una aplicación en diferentes ambientes. Suele usarse para tareas tales como compilar proyectos, empaquetar, versionar y publicar.

### 3.5 Pruebas de Rendimiento

Mediante el uso de Jmeter se realizan pruebas para mejorar el rendimiento mediante la realización de pruebas que simulan cargas de estrés. Devuelve estadísticas de tiempo de respuesta y cantidad de recursos necesitados.

## 4. CALIDAD

Las consideraciones arquitectónicas y de diseño mostradas en este documento contribuyen a elevar la calidad del servicio de la siguiente manera:

***Escalabilidad y Confiabilidad:*** apoyada por el diseño modular y la separación en componentes, esta permite distribuir los componentes entre varios servidores en caso de ser requerido.

***Mantenimiento:*** Los patrones arquitectónicos que se usan permiten que la solución sea lo suficientemente flexible separando claramente cada parte funcional de la solución para soportar cambios en una manera consistente y sencilla.

## 5. DOCUMENTACIÓN

Se listan los documentos a entregar junto con el proyecto:

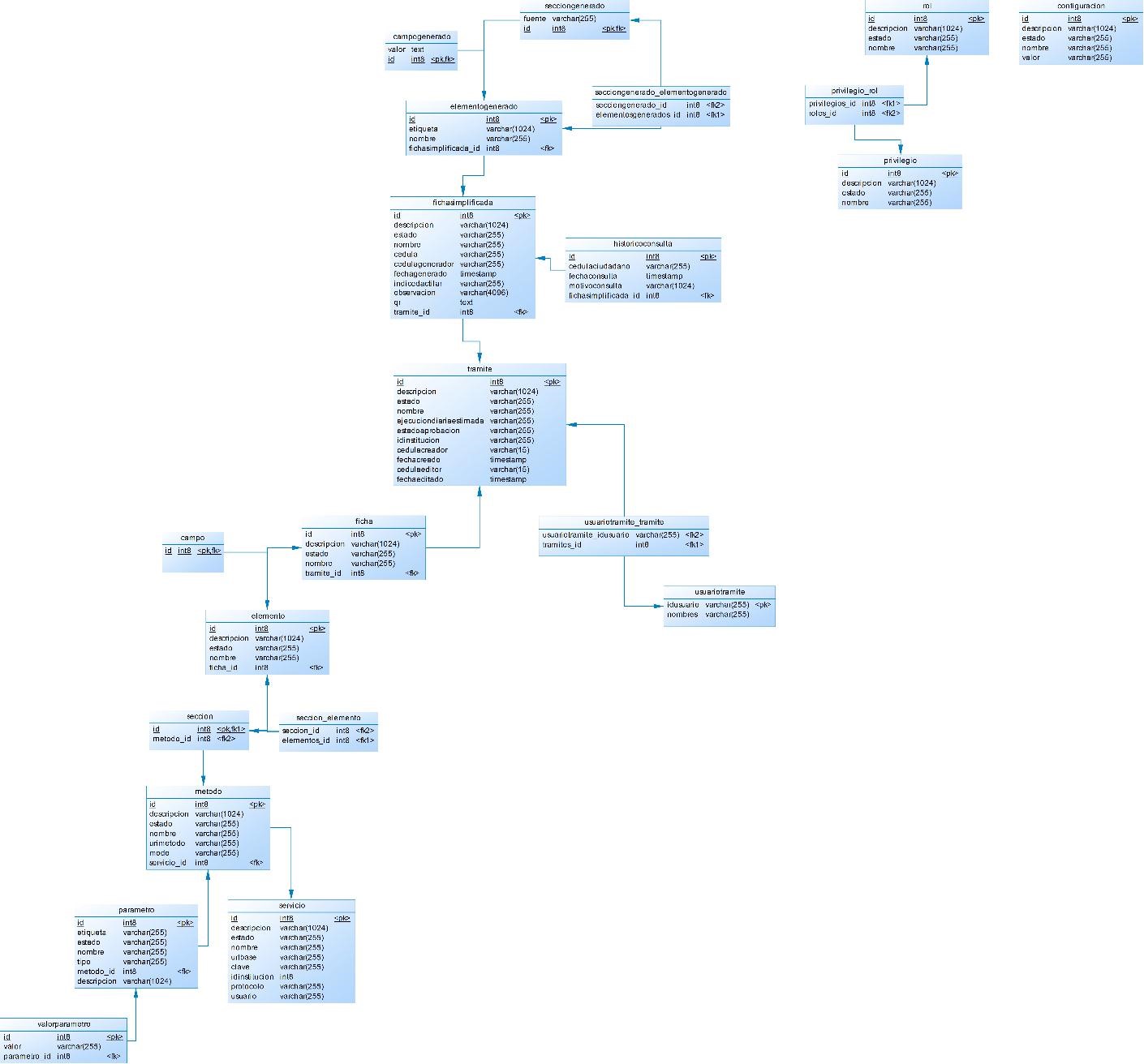
* Diagrama de Arquitectura.
* Diagrama Físico de Base de Datos.
* Manual Técnico de la Herramienta.
* Manual de Usuario.  Plan de Pruebas.
* Catálogo de Errores

## 6. Anexos

### 6.1 Modelo Físico



Modelo Fisico.png

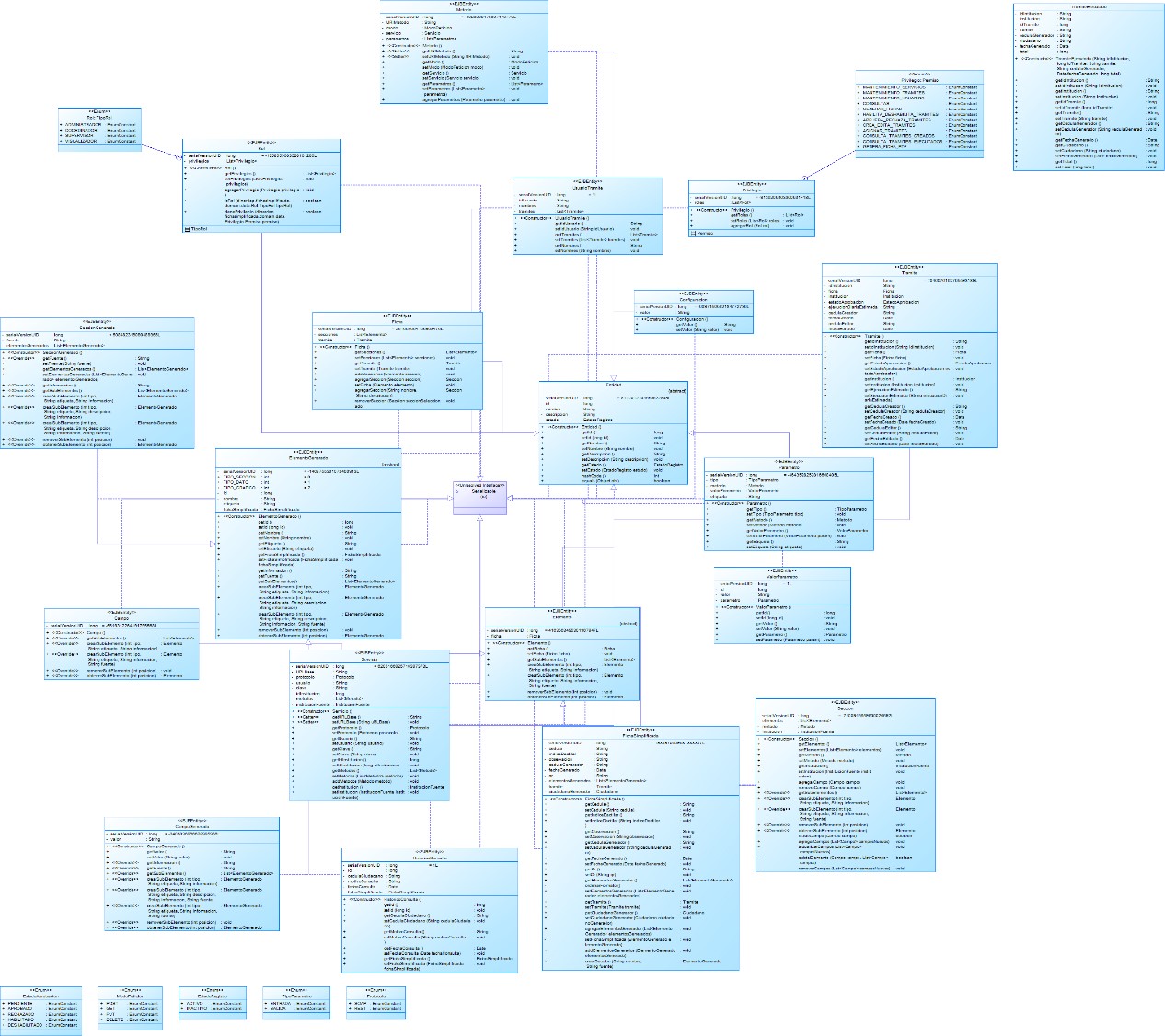


### 6.2 Diagrama de Clases

**6.2.1 Capa de Dominio**



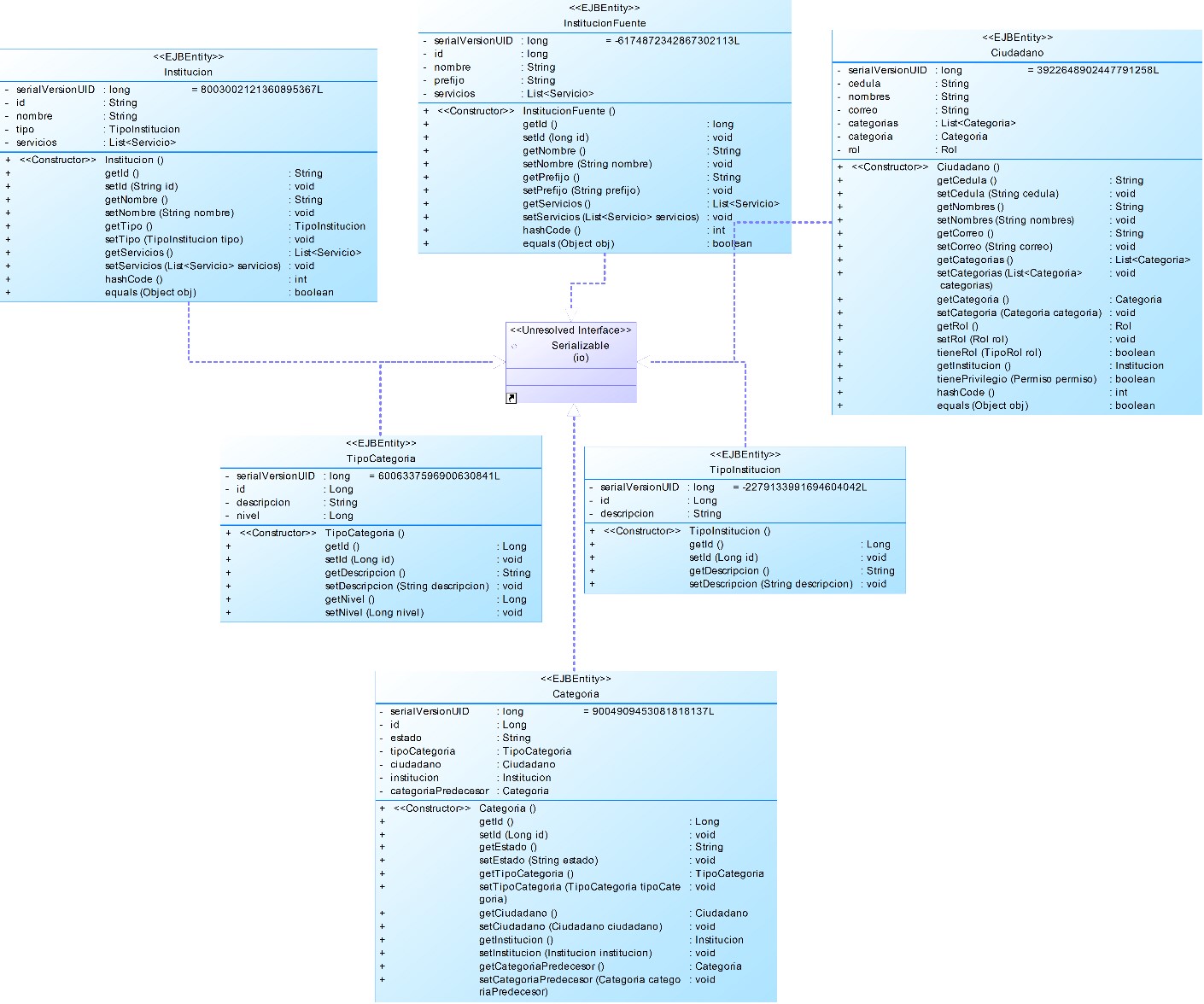
dominio\_fichasimplificada.png



Modelo de Dominio Ficha Simplificada



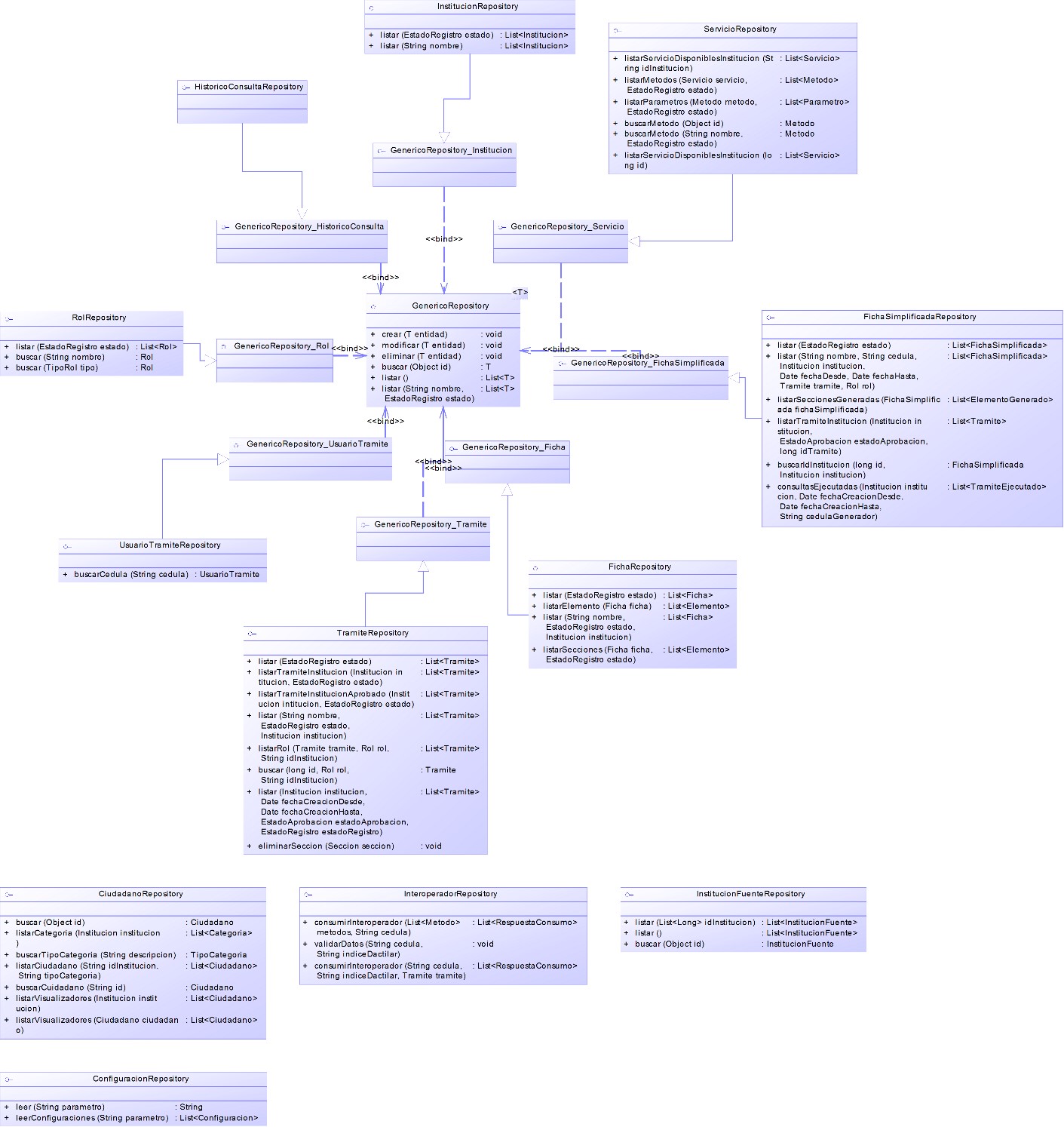
dominio\_sinardap.png



Modelo de Dominio de SINARDAP



InterfacesRepositorios.png

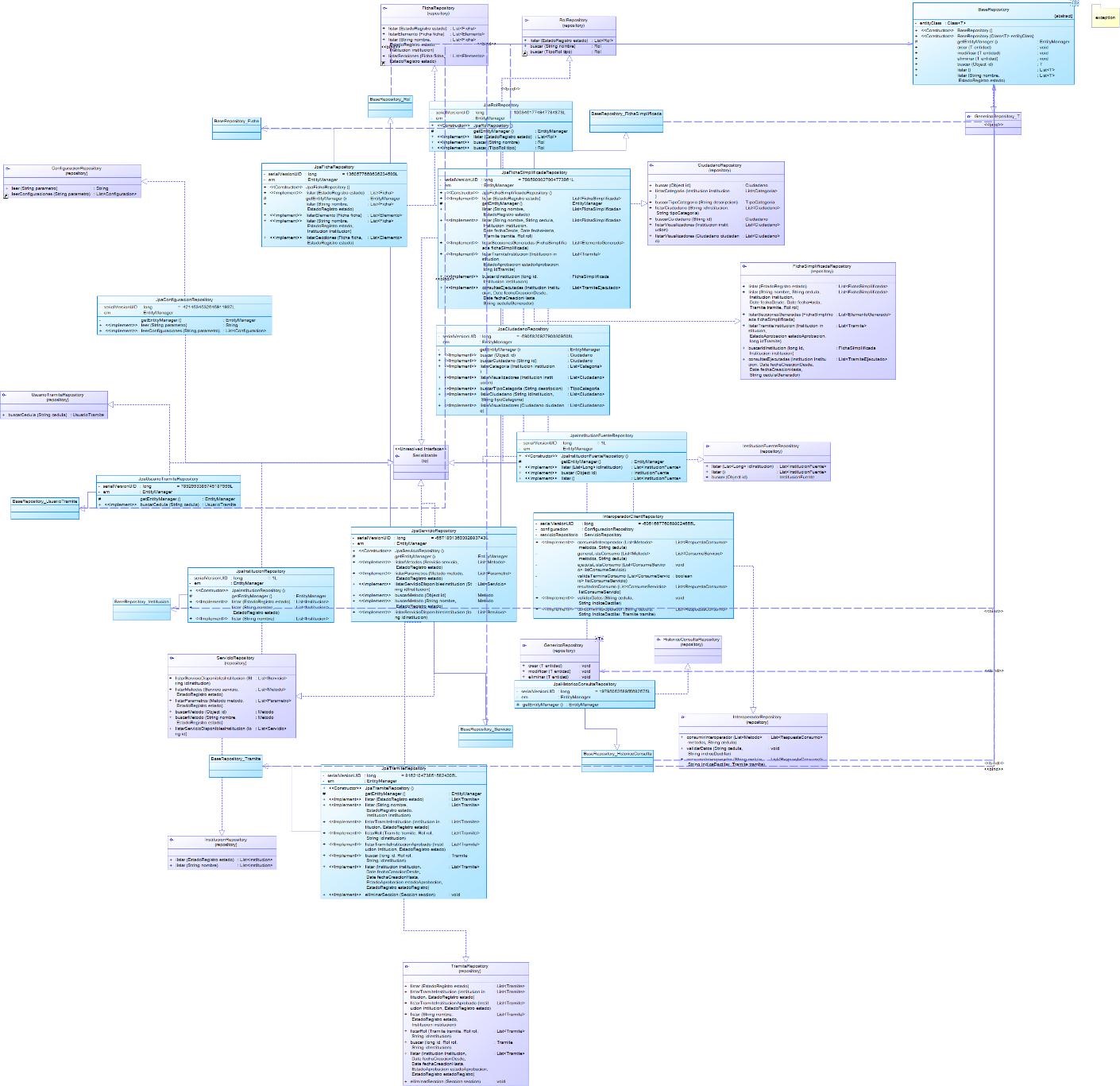


Interfaces Repositorios

#### 6.2.2 Capa de Infraestructura



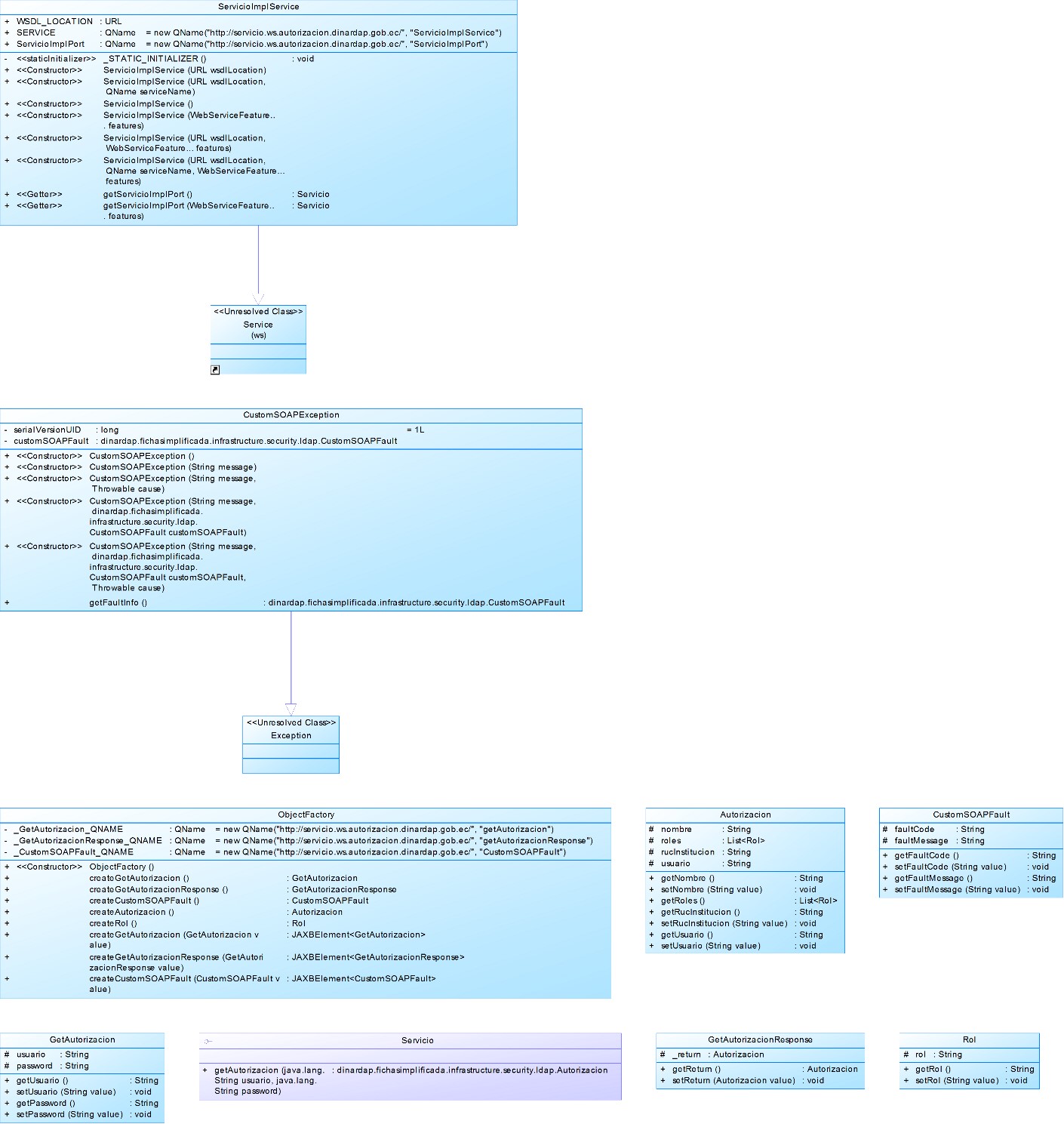
Repositorio.png



Repositorios



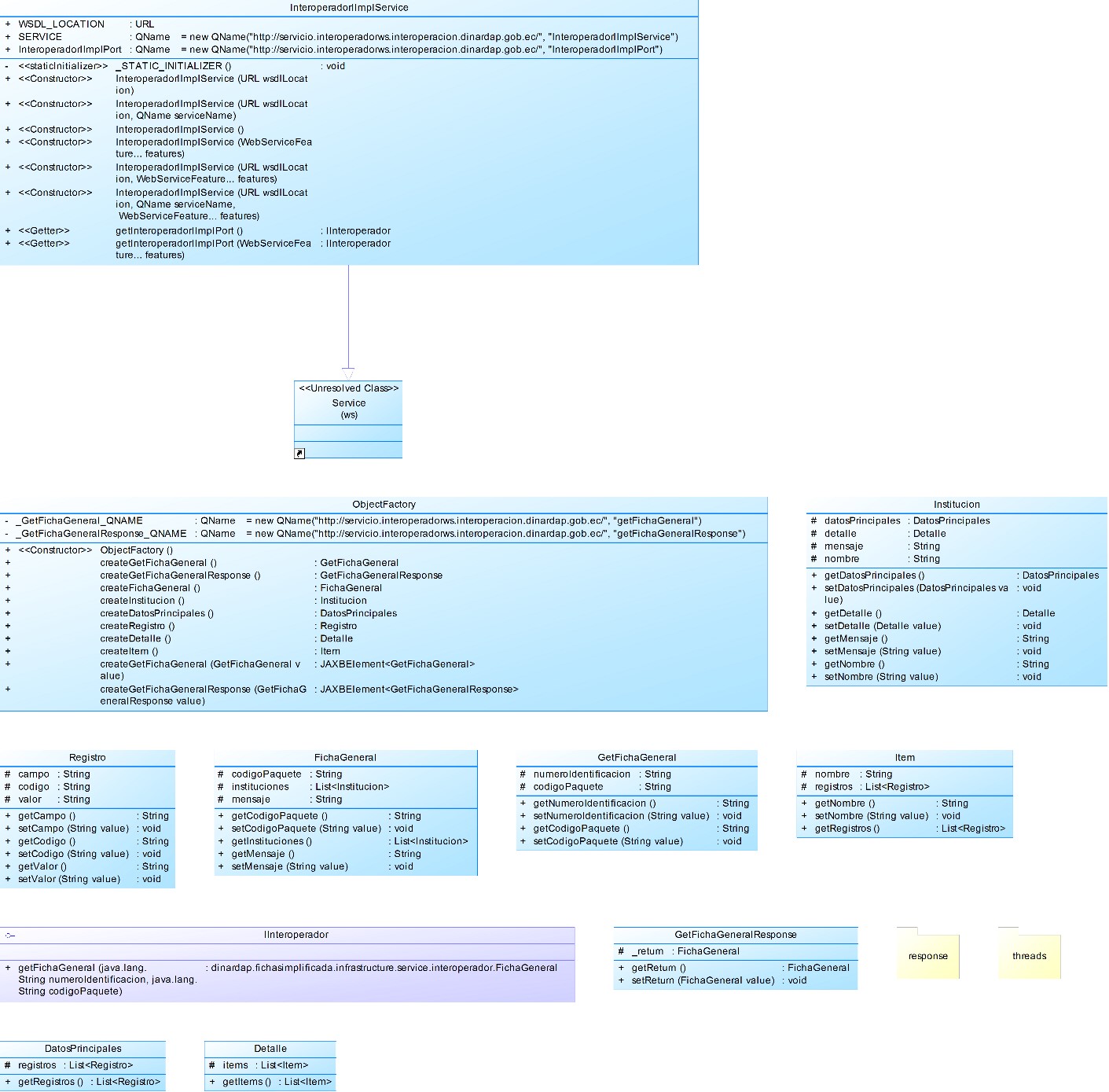
LDAP.png



Seguridad LDAP



Interoperabilidad.png



Interoperabilidad

#### 6.2.3 Capa de Aplicación



app.png

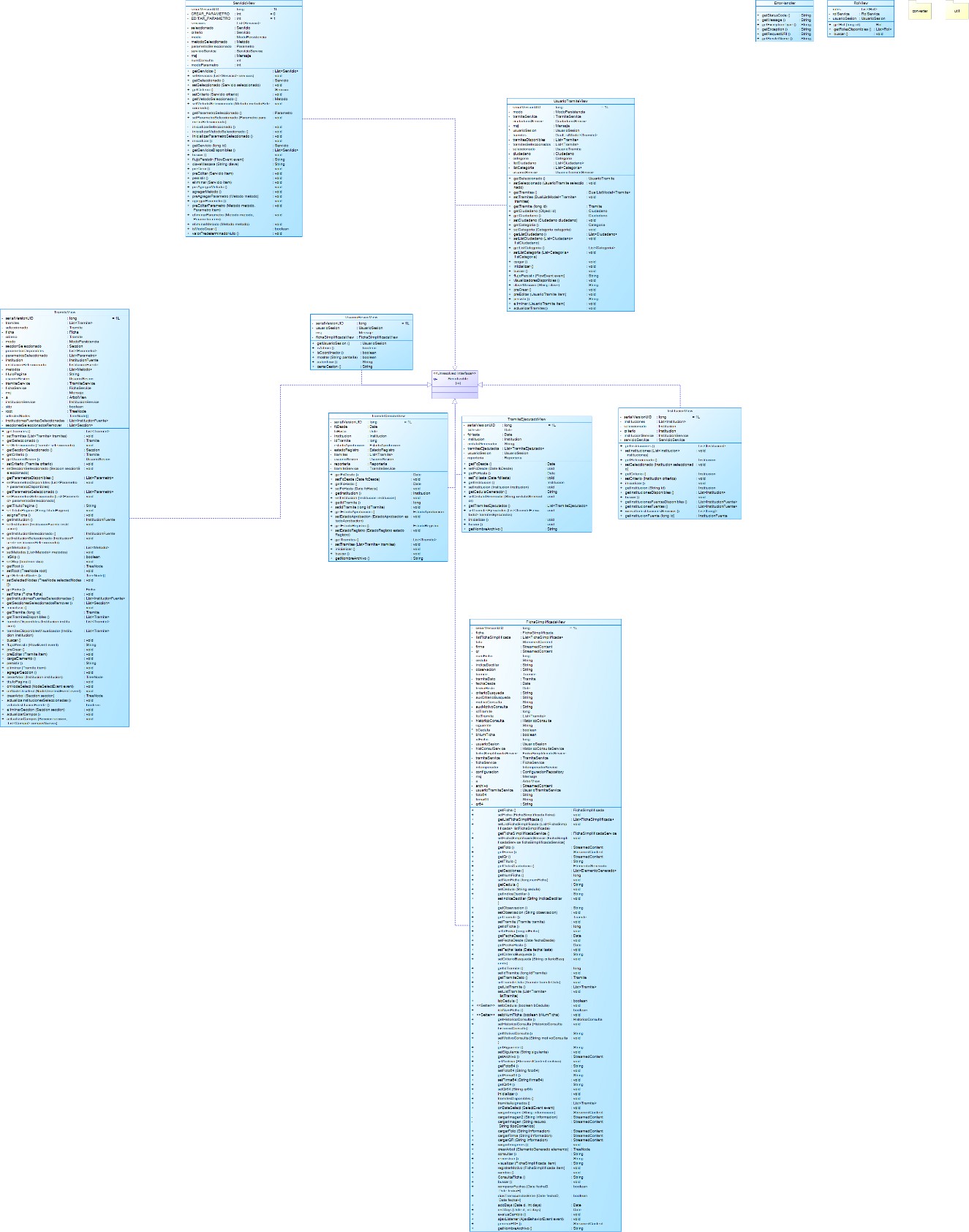


Capa de Aplicación

#### 6.2.4 Capa de UI



Vistas.png



Controlador

es