***WebService DDJJ***

**Documento de Arquitectura de Software**

Versión *0.1*

*30 de Enero de 2017*

**Historia del documento**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Descripción de los cambios** | **Fecha** | **Cambiado por:** | **Revisado por:** |
| **1.0** | **Documento Inicial** | **30/01/2017** | **Hernán Patrón Costas** |  |

**Tabla de contenidos**

[1 Introducción 4](#_Toc364954357)

[1.1 Definiciones, acrónimos y abreviaturas 4](#_Toc364954358)

[1.2 Referencias 4](#_Toc364954359)

[2 Requerimientos especiales 4](#_Toc364954360)

[3 Arquitectura tecnológica 5](#_Toc364954361)

[4 Arquitectura de la aplicación 5](#_Toc364954362)

[4.1 Módulos del sistema 5](#_Toc364954363)

[4.2 Vista de alto nivel 6](#_Toc364954364)

[4.3 Diagrama de clases 6](#_Toc364954367)

[4.4 Transaccionalidad de base de datos 7](#_Toc364954368)

[4.5 Decisiones tecnológicas de base 7](#_Toc364954370)

[4.6 Tecnologías y criterios de diseño 7](#_Toc364954372)

[4.7 Lenguaje y framework de aplicación 7](#_Toc364954374)

[4.8 Motor y framework de base de datos 8](#_Toc364954375)

[5 Manejo de errores 8](#_Toc364954379)

[5.1 Servicios 8](#_Toc364954382)

[5.1.1 Servicios que expone 8](#_Toc364954383)

[5.1.2 Servicios que consume 8](#_Toc364954384)

[7 Modelos de datos 8](#_Toc364954385)

[7.1 Modelo de entidad-relación 8](#_Toc364954386)

[7.2 Modelo de datos físico 8](#_Toc364954387)

[7.2.1 Diccionario de datos 8](#_Toc364954388)

# Introducción

El webservice desarrollado para el proyecto de declaraciones juradas tiene como objetivo la presentación de los declarantes, ministerios y declaraciones juradas pertinentes. En orden de realizar esto, nos basamos en dos webservices preexistentes: uno desarrollado por SADE, el cual se accede con un ID de transacción y devuelve una Declaración jurada, y otro de la Secretaría Legal y Técnica, que provee esos ID. Todo el servicio devuelve objetos del tipo JSON, basándose en el funcionamiento de una API REST.

## Definiciones, acrónimos y abreviaturas

|  |  |
| --- | --- |
| **Término** | **Definición** |
| DDJJ | Declaraciones Juradas |
| SADE | Sistema de Administración de Documentos Electrónicos |
| SLyT | Secretaría Legal y Técnica |
| WS | WebService |

## Referencias

|  |  |  |
| --- | --- | --- |
| **ID Referencia** | **Tipo de archivo** | **Nombre del archivo** |
| 0001 | Documento de Word | DocumentacionDDJJ.docx |
| 0002 | JSON | DDJJSadee.json |
| 0003 | CSV | declaracion590262.csv |
| 0004 | PDF | Documentacion Web Service DataBA.pdf |
| 0005 | Documento de Word | Webservice – Transaccion.docx |

# Requerimientos especiales

El WS debe poder acceder a las siguientes direcciones:

1. <http://euf.hml.gcba.gob.ar/dynform-web/transaccionService>
2. <http://10.79.0.72/ddjj/services/historico/list.php>

Así mismo, debe poder ser accedido por el ciudadano directamente.

# Arquitectura tecnológica

* Sistema Operativo: Red Hat 7.1 o CentOS similar
* Motor de Base de Datos: MariaDB 5.5
* Servidor Web: Apache con PHP 5.4

# Arquitectura de la aplicación

## Módulos del sistema

El WS está planeado para ser accedido por los ciudadanos. Para obtener la información se basa en dos webservices externos: uno de SLyT y otro de SADE, que son accedidos por SOAP. Así mismo, parte de la información obtenida es almacenada en una base de datos de MariaDB (ver diagrama 1).

## Vista de alto nivel



Diagrama 1: Vista de alto nivel

## Diagrama de clases

El WebService no utiliza más clases que las especificadas en el diagrama 1, salvo dos clases auxiliares:



La primera para generar los JSON y CSV (ver documentos 0002 y 0003) que serán devueltos al usuario y la segunda para acceder al webservice de la SLyT.

## Transaccionalidad de base de datos

La base de datos permite la actualización, inserción y selección de filas, dándose éstas en los siguientes métodos:

1. actualizarDDBB: Actualiza las bases de datos e inserta nuevas filas en caso de haber nuevos elementos que añadir.
2. Ministerios/cargos/declaraciones/declarantes: Permite seleccionar y mostrar los elementos de las tablas

No hay otros métodos que permitan la interacción con esta base.

## Decisiones tecnológicas de base

|  |  |  |  |
| --- | --- | --- | --- |
| **Capa** | **Servicio** | **Producto / Tecnología** | **Descripción** |
| Usuario | DDJJ | PHP |  |
| Obtención de información | SADE y SLyT | SOAP |  |
| Almacenamiento de información | - | MariaDB |  |

## Tecnologías y criterios de diseño

Para el desarrollo del WebService se eligió PHP para mayor facilidad de interacción con los webservices de la SLyT y SADE que utilizan el protocolo SOAP, ya que este lenguaje presenta interacción nativa con este protocolo.

Así mismo, se eligió MariaDB (SQL) como software de Base de Datos ya que tiene una gran facilidad de uso con PHP y es de fácil homologación.

## Lenguaje y framework de aplicación

PHP, sin ningún Framework particular.

## Motor y framework de base de datos

MariaDB.

# Manejo de errores

Como éste WebService actúa de intermediario entre los otros WS, el manejo de errores viene dado por ellos. Sin embargo, presenta errores por fallos en la conexión con la base de datos, con los webservices, por falta de parámetros, o errores en los datos.

## Servicios

### Servicios que expone

Ver documento 0001.

### Servicios que consume

Según lo visto, el WS utiliza otros dos WS que se pueden ver en los documentos 0004 y 0005.

# Modelos de datos

## Modelo de datos físico

### Diccionario de datos

Todos los datos usados se pueden ver en el diagrama 1 con sus respectivas tablas.