

## **Arquitecturas Orientadas a Servicios**

### **Actividad 2: Identificación y desarrollo de potenciales servicios**

#### **Descripción del proyecto**

El proyecto busca crear una aplicación web que permita consultar el estado de las multas o infracciones de tránsito en Cali mediante servicios web basados en SOAP

El proyecto cuenta con información estática para efecto de pruebas.

#### **Listado de servicios**

1. Multas por mes
  - Entrada: Mes
  - Salida: Listado de multas
2. Propietario de multa
  - Entrada: Placa, Mes
  - Salida: Objeto Propietario
3. Total deuda
  - Entrada: DNI
  - Salida: Valor numérico de multa
4. Detalle de multa
  - Entrada: DNI, Mes
  - Salida: Valor Multa, Objeto Propietario, Objeto Vehículo
5. Registro de multa
  - Entrada: Valor, Vehículo, Propietario
  - Salida: Confirmación
6. Consulta multa
  - Entrada: Placa
  - Salida: Boolean
7. Multas registradas por meses
  - Entrada: Arreglo de meses, Placa
  - Salida: Arreglo de enteros

## **Servicios implementados**

Para efectos de la práctica se implementaron los servicios 2,4,6,7:

### **2. Propietario Multa:**

El servicio permite conocer el infractor de una multa recibiendo como parámetros de entrada la placa del vehículo y el mes en el cual se registró la multa.

El objetivo es saber quien es el responsable de la infracción.

Datos de entrada:

- Placa: String, identificación del vehículo
- Mes: String, mes del año en el que se presentó la infracción

Datos de salida:

- Propietario: Objeto de tipo Propietario, contiene la información de identificación de la persona

### **4. Detalle de multa:**

El servicio permite obtener información detallada de la infracción tal como el valor de la multa, el infractor y el vehículo implicado a partir del DNI de una persona y el mes en el que se registró la multa

Datos de entrada:

- DNI: String, identificación de la persona a detallar información
- Mes: String, mes del año en el que se registró la multa

Datos de salida:

- Valor Multa: Integer, valor numérico de la multa
- Propietario: Objeto Propietario, contiene la información de identificación de la persona
- Vehículo: Objeto Vehículo, contiene la información de identificación del vehículo

### **6. Consulta multa:**

El servicio permite conocer si tiene una multa registrada en el sistema a partir de la placa del vehículo.

Datos de entrada:

- Placa: String, identificación del vehículo

Datos de salida:

- Confirmación: Boolean, falso o verdadero de acuerdo a la consulta

#### *7. Multas registradas por meses:*

El servicio permite conocer el valor de las multas registradas en los diferentes meses ingresados a partir de un arreglo de meses.

Datos de entrada:

- Arreglo de meses: String[], arreglo con los meses que se consultaran
- Placa: String, identificación del vehículo

Datos de salida:

- Arreglo de enteros: Integer[], arreglo con los valores numéricos de las posibles multas registradas en los meses ingresados, si no hay multa se ingresa 0

### **Información adicional**

El proyecto se realizó en Java con JWS y Wildfly para el despliegue.

Para probar el correcto funcionamiento se utilizó SoapUI y un cliente Java en donde se puede evidenciar el correcto envío de información. El cliente no se incluye en los fuentes

#### *Despliegue del proyecto*

Nombre del proyecto: Multas\_Web

Clase @Web Service: ServiciosMulta

Puerto Wildfly: 8080

URL\_WSDL: [http://localhost:8080/Multas\\_Web/ServiciosMulta?wsdl](http://localhost:8080/Multas_Web/ServiciosMulta?wsdl)

#### *Detalles de implementación*

Para poder utilizar los tipos de dato complejos fue necesario etiquetar las clases en XML para que pudieran traducirse en el WSDL.

Las etiquetas usadas fueron:

- @XmlAccessorType
- @XmlType
- @XmlElement

Las cuales se usan para detallar la clase, el nombre y los atributos, el último se utiliza para definir el elemento como requerido

#### *Ejemplos para prueba*

Para poder probar se definieron las siguientes colecciones de datos

placas = {"ZNL538","VCD709","CFN150","SSH485","HEP908"}

dni = {"1144067772","1143567812","1140034897","1144340675","31872462"}

meses = {"Enero","Marzo","Mayo","Abril","Noviembre"}