

# Tecnologías para la web

## Taller 3

Carlos Motta

En este taller debíamos crear un servicio web SOAP, adicionarlo a la aplicación que habíamos realizado en el taller 2 de los certificados de ICTAC y consumirlo utilizando 3 tecnologías diferentes.

Para el primer punto se tenía que crear tres servicios principales.

- Consultar si el usuario tiene certificado. Recibe un número de identificación y un tipo de identificación. Retorna true si el usuario se encuentra en la base de datos de certificados y false si no se encuentra.

La figura 1 muestra el código de la función que hace la consulta en la base de datos y la figura 2 muestra el código de la declaración del servicio web.

```
23 // Consulta a la base de datos para saber si el usuario se encuentra en la DB
24 function consultarCertificadoUsuarioDB($id, $tipoId){
25
26     $stripID = strip_tags($id);
27     $hscID = htmlspecialchars($stripID);
28
29     $stripType = strip_tags($tipoId);
30     $hscType = htmlspecialchars($stripType);
31
32
33     $inDB = false;
34     $conn = startConnection();
35
36     // Por alguna razon este query funciona en SoapUI, pero no en python, ruby y php
37     // $sql = 'SELECT documento, tipoDoc FROM Persona WHERE documento = ' . $hscID . ' and tipoDoc = ' . $hscType;
38
39     // Por alguna razon este query funciona en php, python y ruby, pero no en soapUI
40     $sql = 'SELECT documento, tipoDoc FROM Persona WHERE documento = ' . $hscID . ' and tipoDoc = ' . $hscType;
41
42     $result = $conn->query($sql);
43
44     if ($result->num_rows > 0) {
45         // output data of each row
46         $inDB = true;
47     }
48
49     closeConnection($conn);
50
51     return $inDB;
52 }
53
```

Figura 1. Función para consultar si el usuario tiene certificado .

The image shows a code editor with several tabs at the top: 'ejemplo1.lisp', 'ConsumirServicios.php', 'ConsumirServicios.rb', 'ExponerServiciosWebICTAC.php', 'dbManager.php', 'ictacDBserv (1).php', and 'db'. The active tab is 'ExponerServiciosWebICTAC.php'. The code is in PHP and defines a SOAP service. It starts with a namespace declaration and a WSDL configuration. Then, it registers a service named 'ConsultarCertificadoUsuario' with two parameters: 'ID' (xsd:string) and 'tipoID' (xsd:string). The return type is 'xsd:boolean'. A function 'ConsultarCertificadoUsuario' is defined, which calls a database function 'ConsultarCertificadoUsuarioDB' and returns a 'soapval' object.

```
1 <?php
2 require_once('lib/nusoap.php');
3 require('dbManager.php');
4 // namespace
5 $ns="http://mottz.info/ictac2015";
6 $server= new nusoap_server();
7 $server->configureWSDL('ExponerServiciosWebICTAC',$ns);
8
9
10 // Registrar servicios
11 $server->register('ConsultarCertificadoUsuario', array('ID'=>'xsd:string', 'tipoID'=>'xsd:string'),
12     array('return'=>'xsd:boolean'),
13     $ns);
14
15
16 function ConsultarCertificadoUsuario($ID, $tipoID){
17     $inDB = ConsultarCertificadoUsuarioDB($ID, $tipoID);
18
19     return new soapval('return', 'xsd:boolean', $inDB);
20 }
21
22
23
24
25
26
```

Figura 2. Declaración del servicio web.

- Consultar la información de un usuario. Recibe un número de identificación y un tipo de identificación. Retorna un objeto complejo (complexType) que tendrá los nombres y apellidos del usuario. Investigar como retornar un objeto complejo en NUSOAP

La figura 3 muestra el código de la función que hace la consulta en la base de datos y la figura 4 muestra el código de la declaración del servicio web.

```

53
54 // Consulta a la base de datos para obtener el nombre y apellido de la persona
55 function ConsultarInfoUsuarioDB($id, $tipoId){
56     $stripID = strip_tags($id);
57     $hscID = htmlspecialchars($stripID);
58
59     $stripType = strip_tags($tipoId);
60     $hscType = htmlspecialchars($stripType);
61
62     $conn = startConnection();
63
64     /*
65     // Por alguna razon este query funciona en SoapUI, pero no en python, ruby y php
66     $sql = 'SELECT nombre, apellido, documento, tipoDoc';
67     $sql .= ' FROM Persona WHERE documento = ';
68     $sql .= $hscID . ' and tipoDoc = ' . $hscType;
69     */
70
71     // Por alguna razon este query funciona en php, python y ruby, pero no en soapUI
72     $sql = 'SELECT nombre, apellido, documento, tipoDoc';
73     $sql .= ' FROM Persona WHERE documento = ';
74     $sql .= "''.$hscID.\"" . ' and tipoDoc = ' . "''.$hscType.\"";
75
76     $result = $conn->query($sql);
77
78     if ($result->num_rows > 0) {
79         $row = $result->fetch_assoc();
80     }
81     closeConnection($conn);
82
83     return $row;
84 }
85

```

Figura 3. Función para consultar la información de un usuario.

```
ejemplo1.lisp x ConsumirServicios.php x ConsumirServicios.rb x ExponerServiciosWebICTAC.php x dbManager.php x ictacDBserv (T
1 <?php
2 require_once('lib/nusoap.php');
3 require('dbManager.php');
4 // namespace
5 $ns="http://mottz.info/ictac2015";
6 $server= new nusoap_server();
7 $server->configureWSDL('ExponerServiciosWebICTAC',$ns);
8
9 // Adding complex type
10 $server->wsdl->addComplexType('contactInfo',
11     'complexType',
12     'struct',
13     'all',
14     '',
15     array('first_name' => array('name' => 'first_name', 'type' => 'xsd:string'),
16         'last_name' => array('name' => 'last_name', 'type' => 'xsd:string')
17     )
18 );
19
20
21 function ConsultarInfoUsuario($ID, $tipoID){
22     $res = ConsultarInfoUsuarioDB($ID, $tipoID);
23     $list = array('first_name' => $res['nombre'], 'last_name' => $res['apellido']);
24
25     return $list;
26 }
27
28
29
```

Figura 4. Declaración del servicio web.

- Consultar si un usuario ha descargado el certificado. Ajuste el código del taller anterior para que cuando un usuario genere el pdf de sus certificados se guarde en la aplicación que los certificados se descargaron. Luego construya un servicio que reciba el número de identificación y el tipo de documento y retornar true si el usuario ya descargó el certificado y false en caso contrario.

La figura 5 muestra el código de la función que hace la consulta en la base de datos y la figura 6 muestra el código de la declaración del servicio web.

```

86 // Consulta la base de datos para saber que certificados fueron descargados
87 function ConsultarDescargaCertificadoDB($id, $tipoId){
88     $stripID = strip_tags($id);
89     $hscID = htmlspecialchars($stripID);
90
91     $stripType = strip_tags($tipoId);
92     $hscType = htmlspecialchars($stripType);
93
94     $conn = startConnection();
95     $checked = false;
96
97     /*
98     // Por alguna razon este query funciona en SoapUI, pero no en python, ruby y php
99     $sql = "SELECT Persona_x_Actividad.descargado ";
100     $sql .= "FROM Persona inner join Persona_x_Actividad ";
101     $sql .= "WHERE Persona_x_Actividad.id_persona = Persona.id_persona ";
102     $sql .= "and Persona.documento = ";
103     $sql .= $hscID;
104     $sql .= " and Persona.tipoDoc = ";
105     $sql .= $hscType;
106     */
107
108     // Por alguna razon este query funciona en php, python y ruby, pero no en soapUI
109     $sql = "SELECT Persona_x_Actividad.descargado ";
110     $sql .= "FROM Persona inner join Persona_x_Actividad ";
111     $sql .= "WHERE Persona_x_Actividad.id_persona = Persona.id_persona ";
112     $sql .= "and Persona.documento = ";
113     $sql .= "".$hscID."" ;
114     $sql .= " and Persona.tipoDoc = ";
115     $sql .= "".$hscType."" ;
116
117     $result = $conn->query($sql);
118
119     if ($result->num_rows > 0) {
120         // output data of each row
121         while($row = $result->fetch_assoc()) {
122             if($row['descargado'] == 1){
123                 $checked = true;
124             }
125         }
126     }
127     closeConnection($conn);
128     return $checked;
129 }
130

```

Figura 5. Función para consultar si un usuario ha descargado un certificado.

```
ejemplo1.lisp x ConsumirServicios.php x ConsumirServicios.rb x ExponerServiciosWebICTAC.php ● dbManager.php x ictacDBserv (1).php x db.php
1 <?php
2 require_once('lib/nusoap.php');
3 require('dbManager.php');
4 // namespace
5 $ns="http://mottz.info/ictac2015";
6 $server= new nusoap_server();
7 $server->configureWSDL('ExponerServiciosWebICTAC',$ns);
8
9
10 $server->register('ConsultarDescargaCertificado', array('ID'=>'xsd:string', 'tipoID'=>'xsd:string'),
11 array('return'=>'xsd:boolean'),
12 $ns);
13
14
15 function ConsultarDescargaCertificado($ID, $tipoID){
16     $checked = ConsultarDescargaCertificadoDB($ID, $tipoID);
17     return new soapval('return', 'xsd:boolean', $checked);
18 }
19
20
21
22
23
24
25
```

Figura 6. Declaración del servicio web.

Para el segundo punto se utilizaron las siguientes tecnologías:

- PHP

En PHP se utilizó la extensión **NUSOAP** para crear el servicio web y para acceder a él como cliente. En la figura 7 se muestra el cliente que consume los 3 servicios y los muestra en pantalla.

```
1 <?php
2 require_once('/lib/nusoap.php');
3 $objClienteSOAP = new nusoap_client('http://mottz.info/ictac2015/ExponerServiciosWebICTAC.php?wsdl','wsdl');
4 $param = array('ID' => '1144052533', 'tipoID' => 'ID');
5
6 $resultado1 = $objClienteSOAP->call('ConsultarInfoUsuario', $param);
7 $resultado2 = $objClienteSOAP->call('ConsultarDescargaCertificado', $param);
8 $resultado3 = $objClienteSOAP->call('ConsultarCertificadoUsuario', $param);
9
10
11 echo '<br>';
12 print_r($resultado1);
13 echo '<br>';
14
15 echo '<br>';
16 print_r($resultado2);
17 echo '<br>';
18
19 echo '<br>';
20 print_r($resultado3);
21 echo '<br>';
22
23 ?>
24
```

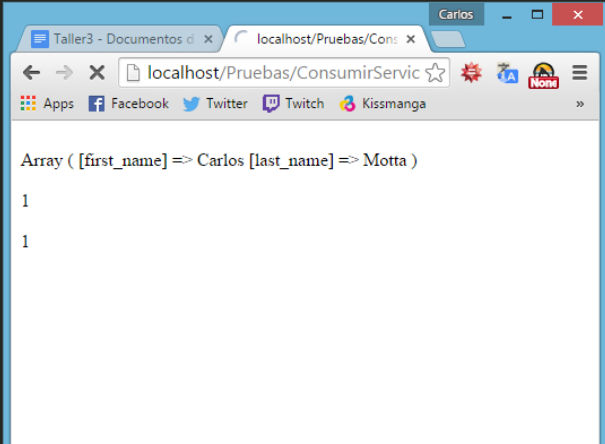


Figura 7. Cliente y consumidor de los servicios web en PHP.

### Requerimientos de instalación

Solo se necesita tener un servidor que entienda PHP y haber descargado **NUSOAP** y tenerlo en el mismo directorio del archivo de PHP.

#### ● Python

En Python se utilizo la libreria **SUDS** para consumir el servicio web de manera facil y rapida. En la figura 8 se muestra el script del cliente creado en Python que consume los 3 servicios.

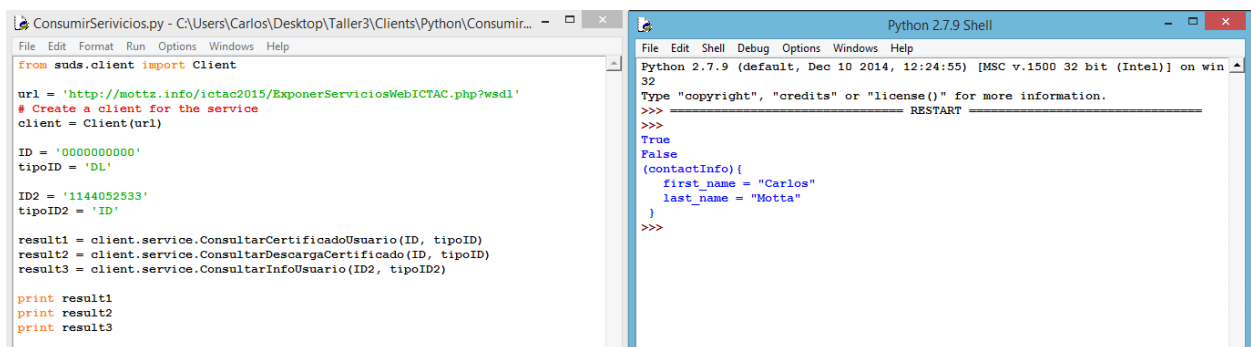


Figura 8. Cliente y consumidor de los servicios web en Python.

### Requerimientos de instalación

Solo se necesita tener Python instalado, haber descargado **SUDS** y haberlo colocado en la carpeta Lib del directorio donde se instaló Python

#### ● Ruby

En Ruby se utilizó un *gem* conocido como **SAVON** para consumir los 3 servicios web fácil de usar y sin tener que lidiar con **SOAP4R**, la implementación de SOAP para Ruby, la cual es muy ambigua, ineficiente y con muchas dependencias hasta el punto de tener que modificar archivos dentro de la carpeta de Ruby para que funcione correctamente. La figura 10 muestra el script del cliente creado en Ruby que consume los 3 servicios.

```
ejemplo1 x Consumir x Consumir x ExponerSe x dbManag x ictacObsen x db.php x poolPDF x Calculado x index.php x gener
1 require 'savon'
2
3 # create a client for the service
4 client = Savon.client(wsdl: 'http://mottz.info/ictac2015/ExponerServiciosWebICTAC.php?wsdl')
5
6 client.operations
7
8 # Consultar Certificado Usuario
9 response = client.call(:consultar_certificado_usuario, message: { id: "1144852533", tipoID: "ID" })
10 puts response.doc
11
12 # Consultar Descarga Certificado
13 response = client.call(:consultar_descarga_certificado, message: { id: "1144852533", tipoID: "ID" })
14 puts response.doc
15
16 # Consultar Informacion Usuario
17 response = client.call(:consultar_info_usuario, message: { id: "1144852533", tipoID: "ID" })
18 puts response.doc
```

```
Ins:xs:base="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XML
Schema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP
-ENC:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
<SOAP-ENV:Body>
  <ns1:ConsultarCertificadoUsuarioResponse xmlns:ns1="http://mottz.info/ictac2
015">
    <return xsi:type="xsd:boolean">true</return>
  </ns1:ConsultarCertificadoUsuarioResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XML
Schema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP
-ENC:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  <SOAP-ENV:Body>
    <ns1:ConsultarDescargaCertificadoResponse xmlns:ns1="http://mottz.info/ictac
2015">
      <return xsi:type="xsd:boolean">true</return>
    </ns1:ConsultarDescargaCertificadoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XML
Schema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xm
lns:tns="http://mottz.info/ictac2015" SOAP-ENC:encodingStyle="http://schemas.xmlso
ap.org/soap/encoding/"
  <SOAP-ENV:Body>
    <ns1:ConsultarInfoUsuarioResponse xmlns:ns1="http://mottz.info/ictac2015">
      <return xsi:type="tns:contactInfo">
        <first_name xsi:type="xsd:string">Carlos</first_name>
        <last_name xsi:type="xsd:string">Motta</last_name>
      </return>
    </ns1:ConsultarInfoUsuarioResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
C:\Users\Carlos\Desktop\Taller3\Clients\Ruby>
```

Figura 9. Cliente y consumidor de los servicios web en Ruby.

## Requerimientos de instalación

A diferencia de **SOAP4R**, **SAVON** es muy simple de instalar, primero se necesita tener Ruby instalado y luego a través del comando *gem install savon* se instala la librería y eso debería ser suficiente para que funcione. Sin embargo, en algunos computadores parece haber un problema intentando encontrar el repositorio de savon y para solucionar esto adicioné un repositorio para que pudiera buscarlo y descargarlo con facilidad, esto lo hice con el siguiente comando *gem sources -a http://rubygems.org*.