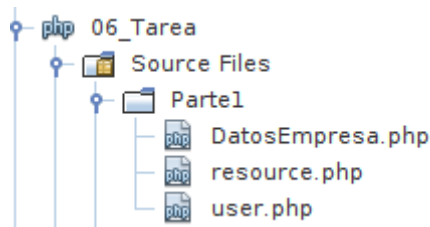


Plan de pruebas DWES06: Servicios Web:

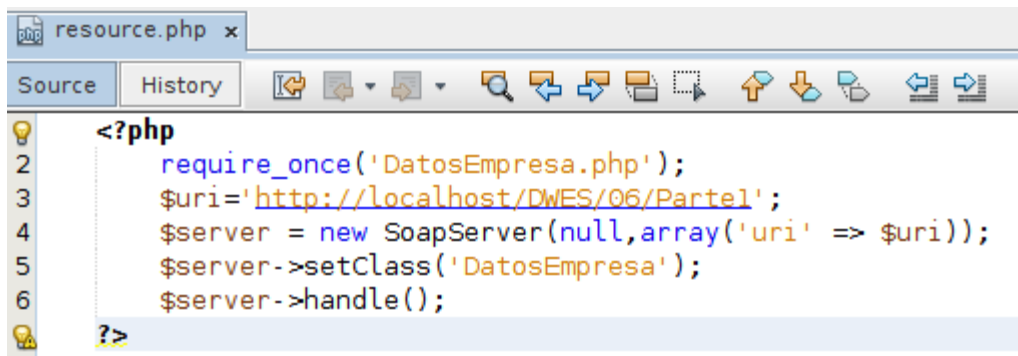
Parte 1:

Servicio Web sin WSDL:

Tenemos 3 ficheros: DatosEmpresa.php, resource.php y user.php



resource.php contiene el script que ejecuta el servicio web:



user.php contiene el cliente que se va a conectar al servicio y llamar a las funciones que contiene:

```
<h2>Servicios Web SOAP sin WSDL</h2>
<h3>Autor: Iker Macaya Faber</h3>
<?php
$url='http://localhost/DWES/06/Parte1/resource.php';
$uri='http://localhost/DWES/06/Parte1';
$cliente = new SoapClient(null,array('location'=>$url,'uri'=>$uri));
// Ejemplo de getFechaNacimiento del empleado 10002
$fecha_nac=$cliente->getFechaNacimiento(10002);
print_r("Fecha de nacimiento del empleado seleccionado ".$fecha_nac."<br>");
print_r("<br>");
```

DatosEmpresa.php que contiene las 4 funciones publicadas en el servicio:

```
public function getFechaNacimiento($cod) {
    $sql = "SELECT fecha_nac FROM empleados WHERE emp_no =" . $cod . " ";
    $result = self::ejecutaConsulta($sql);
    $fecha_nac=$result->fetch();
    $fecha = $fecha_nac[0];
    return $fecha;
}
```

Comprobación de las 4 funciones publicadas en el servicio:

getFechaNacimiento: función que recibe el código de un empleado como parámetro y devuelve su fecha de nacimiento:

```
public function getFechaNacimiento($cod) {
    $sql = "SELECT fecha_nac FROM empleados WHERE emp_no = ".$cod." ";
    $result = self::ejecutaConsulta($sql);
    $fecha_nac=$result->fetch();
    $fecha = $fecha_nac[0];
    return $fecha;
}
```

La llamada desde el cliente:

```
// Ejemplo de getFechaNacimiento del empleado 10002
$fecha_nac=$cliente->getFechaNacimiento(10002);
print_r("Fecha de nacimiento del empleado seleccionado ".$fecha_nac."<br>");
print_r("<br>");
```

Hacemos una llamada a la función para saber la fecha de nacimiento del empleado 10002:

Fecha de nacimiento del empleado seleccionado 1964-06-02

getDepartamentos: función sin parámetros que devuelve una matriz o array con los códigos de todos los departamentos:

```
public function getDepartamentos() {
    $sql = "SELECT emp_no FROM empleados ";
    $result = self::ejecutaConsulta($sql);
    $emp_no=$result->fetch();
    $emp_no = $emp_no[0];
    return $emp_no;
}
```

La llamada desde el cliente:

```
$dptos=$cliente->getDepartamentos();
print_r("Array de codigos de departamentos: ");
print_r("<br>");
print_r($dptos);
print_r("<br>");
```

En este caso no es necesario pasar parámetros a la función y nos devuelve los códigos de todos los departamentos:

Array de codigos de departamentos:

Array ([0] => d009 [1] => d006 [2] => d005 [3] => d002 [4] => d008 [5] => d001 [6] => d004 [7] => d003 [8] => d007)

getEmpleadosDepartamento: función que recibe el código de un departamento y devuelve una matriz o array con los códigos de los empleados de ese departamento:

```
public function getEmpleadosDepartamento($cod) {  
    $sql = "SELECT emp_no FROM dept_emp WHERE dept_no = ".$cod." ";  
    $result = self::ejecutaConsulta($sql);  
    $empleados = array();  
    while ($f = $result->fetch())  
        array_push( $empleados , $f[0]);  
    return $empleados;  
}
```

La llamada desde el cliente:

```
$empleados=$cliente->getEmpleadosDepartamento("'d005'");  
print_r("Array de codigos de empleados del dpto. seleccionado: ");  
print_r("<br>");  
print_r($empleados);  
print_r("<br>");
```

En este ejemplo llamamos a la función consultando los empleados del departamento con código d005:

Array de codigos de empleados del dpto. seleccionado:

Array ([0] => 10001 [1] => 10006 [2] => 10008 [3] => 10012 [4] => 10014

getFechaDesdeEmpleadoDept: función que recibe el código de un departamento y el código de un empleado y devuelve la fecha desde la cual el empleado está en ese departamento:

```
public function getFechaDesdeEmpleadoDept($dpto, $emp) {  
    $sql = "SELECT fecha_desde FROM dept_emp WHERE dept_no = ".$dpto." AND emp_no=".$emp." ";  
    $result = self::ejecutaConsulta($sql);  
    $f = $result->fetch();  
    $fecha = $f[0];  
    return $fecha;  
}
```

La llamada desde el cliente:

```
$fechacontrato=$cliente->getFechaDesdeEmpleadoDept("'d008'", 10007);  
print_r("La fecha de inicio de trabajo en el departamento es: ".$fechacontrato);
```

En este ejemplo hacemos la llamada a la función para saber la fecha del empleado 10007, del departamento d008:

La fecha de inicio de trabajo en el departamento es: 1989-02-10

Visión del servicio con la respuesta de las 4 peticiones hechas por el cliente:

Servicios Web SOAP sin WSDL

Autor: Iker Macaya Faber

Fecha de nacimiento del empleado seleccionado 1964-06-02

Array de codigos de departamentos:

Array ([0] => d009 [1] => d006 [2] => d005 [3] => d002 [4] => d008 [5] => d001 [6] => d004 |

Array de codigos de empleados del dpto. seleccionado:

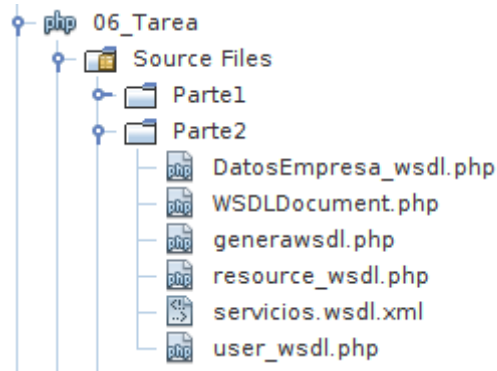
Array ([0] => 10001 [1] => 10006 [2] => 10008 [3] => 10012 [4] => 10014 [5] => 10018 [6] =:

La fecha de inicio de trabajo en el departamento es: 1989-02-10

Parte 2:

Servicio Web con WSDL:

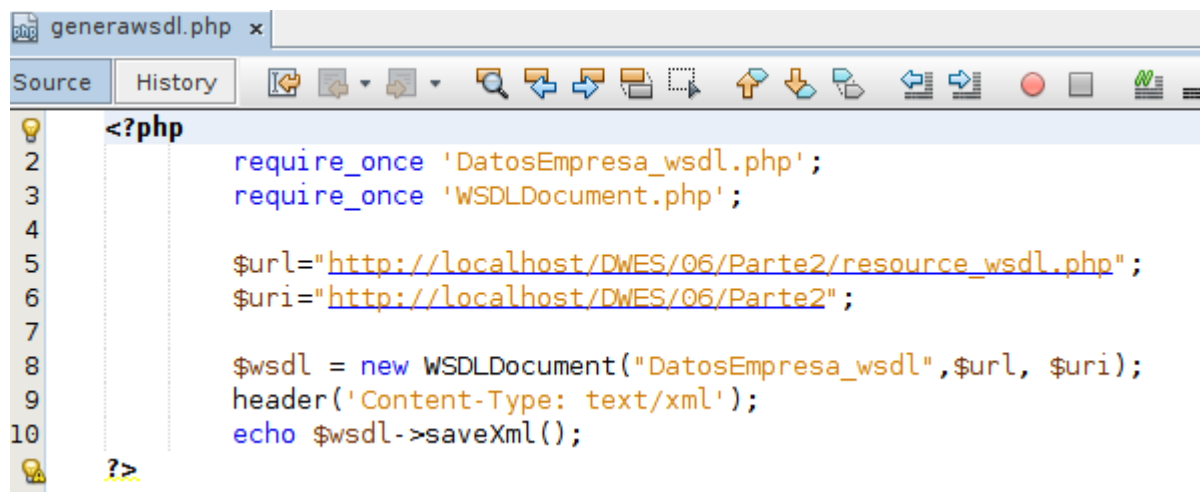
Tenemos 6 ficheros: DatosEmpresa_wsdl.php, generawsdl.php, resource_wsdl.php, servicios.wsdl.xml, user_wsdl.php y WSDLDocument.php.



DatosEmpresa_wsdl.php: Funciones que va a publicar el servicio, comentado para poder generar el xml:

```
<?php
class DatosEmpresa_wsdl {
    /** Metodo para hacer la conexion a la BD y ejecutar las consultas
     *
     * @param string $sql
     * @return string
     */
    public function ejecutaConsulta($sql) {
        $servername = "localhost";
        $dbname="tarea6";
        $username = "dwes";
        $password = "abc123.";
    }
}
```

generawsdl.php: Fichero que nos va a permitir, junto con WSDLDocument.php, generar: servicios.wsdl.xml



```
generawsdl.php x
Source History
<?php
2     require_once 'DatosEmpresa_wsdl.php';
3     require_once 'WSDLDocument.php';
4
5     $url="http://localhost/DWES/06/Parte2/resource_wsdl.php";
6     $uri="http://localhost/DWES/06/Parte2";
7
8     $wsdl = new WSDLDocument("DatosEmpresa_wsdl",$url, $uri);
9     header('Content-Type: text/xml');
10    echo $wsdl->saveXml();
?>
```

resource_wsdl.php: servicio pasado a generawsdl.php para generar el xml:

```
resource_wsdl.php x
Source History
<?php
2     require_once('DatosEmpresa_wsdl.php');
3     $server = new SoapServer(null,array('uri' => ''));
4     $server->setClass('DatosEmpresa_wsdl');
5     $server->handle();
?>
```

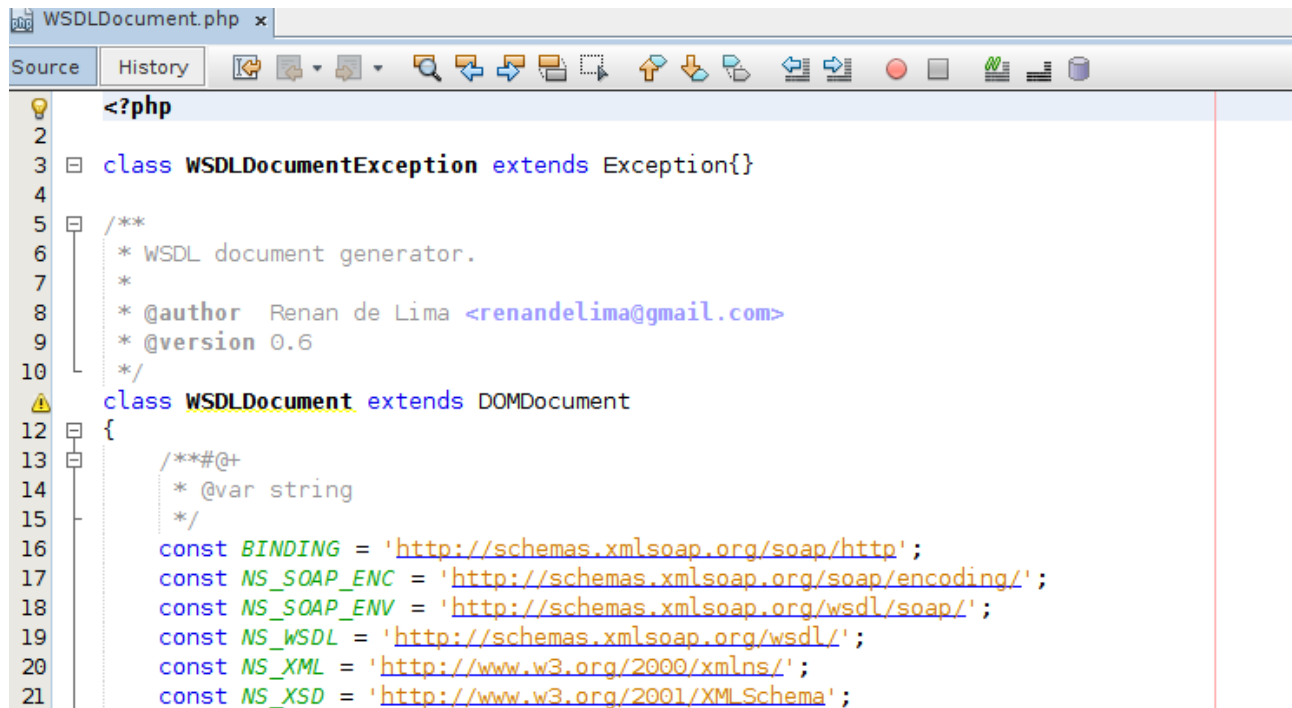
servicios.wsdl.xml: fichero generado a partir de generawsdl.php:

```
servicios.wsdl.xml x
Source History
1 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
2                   xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/"
3                   xmlns:soap-env="http://schemas.xmlsoap.org/wsdl/soap/"
4                   xmlns:tns="http://localhost/DWES/06/Parte2"
5                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6                   targetNamespace="http://localhost/DWES/06/Parte2">
7     <wsdl:types>
8         <xsd:schema targetNamespace="http://localhost/DWES/06/Parte2" />
9     </wsdl:types>
10    <wsdl:message name="ejecutaConsultaRequest">
11        <wsdl:part name="sql" type="xsd:string" />
12    </wsdl:message>
```

user_wsdl.php: cliente que va a hacer las llamadas a las 4 funciones. Ahora al servicio le llamamos a través de servicios.wsdl.xml:

```
user_wsdl.php x
Source History
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
2 http://www.w3.org/TR/html4/loose.dtd">
3 <!-- Desarrollo Web en Entorno Servidor
4 Tema 6: Servicios SOAP
5 Tarea 6: user.php
6 @author: Iker Macaya Faber -->
7 <html>
8 <head>
9     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
10    <title>DWES: Tarea 06</title>
11 </head>
12 <body>
13     <h2>Servicios Web SOAP con WSDL</h2>
14     <h3>Autor: Iker Macaya Faber</h3>
15     <?php
16         $cliente = new SoapClient("servicios.wsdl.xml");
```

WSDLDocument.php: Herramienta usada para generar el servicio:



```
<?php
2
3 class WSDLDocumentException extends Exception{}
4
5 /**
6  * WSDL document generator.
7  *
8  * @author Renan de Lima <renandelima@gmail.com>
9  * @version 0.6
10 */
11 class WSDLDocument extends DOMDocument
12 {
13     /**#@+
14      * @var string
15      */
16     const BINDING = 'http://schemas.xmlsoap.org/soap/http';
17     const NS_SOAP_ENC = 'http://schemas.xmlsoap.org/soap/encoding/';
18     const NS_SOAP_ENV = 'http://schemas.xmlsoap.org/wsdl/soap/';
19     const NS_WSDL = 'http://schemas.xmlsoap.org/wsdl/';
20     const NS_XML = 'http://www.w3.org/2000/xmlns/';
21     const NS_XSD = 'http://www.w3.org/2001/XMLSchema';
```

Comprobación de las 4 funciones publicadas en el servicio:

****** la implementación es igual que en la parte1, las llamadas son diferentes porque cambiamos de parametros y sobre todo en el cliente como se ha indicado anteriormente al servicio se le llama a través del fichero generado servicios.wsdl.xml ******

getFechaNacimiento: función que recibe el código de un empleado como parámetro y devuelve su fecha de nacimiento.

Llamada a la función:

```
$fecha_nac=$cliente->getFechaNacimiento(10024);  
print_r("Fecha de nacimiento del empleado seleccionado ".$fecha_nac."<br>");  
print_r("<br>");
```

En este ejemplo vamos a consultar la fecha de nacimiento del empleado 10024:

Fecha de nacimiento del empleado seleccionado 1958-09-05

getDepartamentos: función sin parámetros que devuelve una matriz o array con los códigos de todos los departamentos.

Llamada a la función:

```
$dptos=$cliente->getDepartamentos();  
print_r("Array de codigos de departamentos: ");  
print_r("<br>");  
print_r($dptos);
```

El resultado es igual que en la parte 1:

Array de codigos de departamentos:

Array ([0] => d009 [1] => d006 [2] => d005 [3] => d002 [4] => d008 [5] => d001 [6] => d004 [7] => d003 [8] => d007)

getEmpleadosDepartamento: función que recibe el código de un departamento y devuelve una matriz o array con los códigos de los empleados de ese departamento.

Llamada a la función:

```
$empleados=$cliente->getEmpleadosDepartamento("'d008'");  
print_r("Aray de codigos de empleados del dpto. seleccionado: ");  
print_r("<br>");  
print_r($empleados);
```

En este ejemplo veremos los códigos de empleado del departamento d008:

**Array de codigos de empleados del dpto. seleccionado:
Array ([0] => 10007 [1] => 10015 [2] => 10019)**

getFechaDesdeEmpleadoDept: función que recibe el código de un departamento y el código de un empleado y devuelve la fecha desde la cual el empleado está en ese departamento.

Llamada a la función:

```
$fechacontrato=$cliente->getFechaDesdeEmpleadoDept("'d003'", 10005);  
print_r("La fecha de inicio de trabajo en el departamento es: ".$fechacontrato);
```

En este ejemplo estamos consultado la fecha de inicio del trabajador 10005 en el departamento d003:

La fecha de inicio de trabajo en el departamento es: 1989-09-12

Aquí todas la peticiones hechas por el cliente al servicio:

Servicios Web SOAP con WSDL

Autor: Iker Macaya Faber

Fecha de nacimiento del empleado seleccionado 1958-09-05

Array de codigos de departamentos:

Array ([0] => d009 [1] => d006 [2] => d005 [3] => d002 [4] => d008 [5] => d001

Array de codigos de empleados del dpto. seleccionado:

Array ([0] => 10007 [1] => 10015 [2] => 10019)

La fecha de inicio de trabajo en el departamento es: 1989-09-12