

# Programación en el entorno Android

Master en Ingeniería Informática - Programación de Dispositivos Móviles

---

## Práctica 4

### Introducción

Aprovechando toda la interfaz de la práctica 3, se trata de acceder a los datos ubicados en un Servidor externo usando los protocolos SOAP y JSON.

De las tres vistas que contiene la práctica (clientes, productos y pedidos), al menos una tiene que ser gestionada usando SOAP y al menos una tiene que ser gestionada usando JSON.

Los alumnos que implementen todas las conexiones al servicio usando un único protocolo obtendrán 8 puntos como máximo. Solo si se implementan los dos protocolos se podrá alcanzar la nota máxima.

### Interfaz

URL del Servicio: <http://appstip.iatext.ulpgc.es/ventas/server.php>

NAMESPACE: <urn://ulpgc.masterii.moviles>

### TABLA CUSTOMERS

**int InsertCustomer(string name, string address).**

Descripción: Inserta un cliente en la base de datos. No se podrán insertar dos clientes con el mismo nombre.

Devuelve: El *IDCustomer*.

**array QueryCustomers()**

Descripción: Solicita la lista completa de clientes ordenados por el nombre.

Devuelve: *array of <IDCustomer, name, address>*.

### **bool UpdateCustomer(int IDCustomer, string name, string address)**

Descripción: Actualiza los campos *name* y *addres* del cliente *IDCustomer*.

Devuelve: *true* si se ha actualizado correctamente y *false* en caso contrario.

### **bool DeleteCustomer(int IDCustomer)**

Descripción: Borra el cliente *IDCustomer* de la base de datos.

Devuelve: *true* si se ha eliminado correctamente y *false* en caso contrario.

## **TABLA PRODUCTS**

### **int InsertProduct(string name, string description, float price).**

Descripción: Inserta un producto en la base de datos. No se podrán insertar dos productos con el mismo nombre.

Devuelve: El *IDProduct*.

### **array QueryProducts()**

Descripción: Solicita la lista completa de productos ordenados por el nombre.

Devuelve: *array of <IDProduct, name, description, price>*.

### **bool UpdateProduct(int IDProduct, string name, string description, float price)**

Descripción: Actualiza los campos *name* y *description* del cliente *IDProduct*.

Devuelve: *true* si se ha actualizado correctamente y *false* en caso contrario.

### **bool DeleteProduct(int IDProduct)**

Descripción: Borra el producto *IDProduct* de la base de datos.

Devuelve: *true* si se ha eliminado correctamente y *false* en caso contrario.

## **TABLA ORDERS**

### **int InsertOrder(string code, string date, int IDCustomer, int IDProduct, int quantity).**

Descripción: Inserta una orden de compra en la base de datos. No se podrán insertar dos órdenes con el mismo código.

Devuelve: El *IDOrder*.

### **array QueryOrders()**

Descripción: Solicita la lista completa de productos ordenados por el nombre.

Devuelve: *array of <IDCustomer, IDProduct, customerName, productName, code, price, quantity, date>*.

### **bool UpdateOrder(int IDOrder, int IDCustomer, int IDProduct, string code, date date, int quantity)**

**Descripción:** Actualiza los campos *IDCustomer*, *IDProduct*, *code*, *date*, *quantity* de la orden.

**Devuelve:** *true* si se ha actualizado correctamente y *false* en caso contrario.

### **bool DeleteOrder(int IDOrder)**

**Descripción:** Borra la orden *IDOrder* de la base de datos.

**Devuelve:** *true* si se ha eliminado correctamente y *false* en caso contrario.

## Notas JSON

Para el caso de JSON, los argumentos deben tener el nombre correcto de la especificación de cada método. Ejemplo para los argumentos de *InsertCustomer()*:

```
JSONObject jsonObject = new JSONObject();  
jsonObject.put("name", "Mi nombre");  
jsonObject.put("address", "Mi dirección");
```

Las respuestas son objetos que tienen dos campos: *fault* y *data*.

- *Fault* a cero indica que todo ha ido bien y lo que devuelve cada método está en el campo *data*.
- *Fault* a uno indica error y en el campo *data* habrá un *null*.

Ejemplo de una *QueryCustomers()* en JSON:

<http://appstip.iatext.ulpgc.es/ventas/server.php?QueryCustomers>

```
{ "fault": 0,  
  "data":  
  [ { "IDCustomer": "1", "name": "Fulano", "address": null },  
    { "IDCustomer": "6", "name": "PerenganoJSON", "address": "desconocida" },  
    { "IDCustomer": "4", "name": "PerenganoSOAP", "address": "desconocida" },  
    { "IDCustomer": "3", "name": "Zutano", "address": "Perojo, 1" }  
  ] }
```

Para el caso SOAP, la respuesta es similar. Les llega un vector con un campo de nombre "result" y otro con "return". Les adjunto el ejemplo de una *QueryCustomers()* en SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>  
<SOAP-ENV:Envelope  
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:ns1="urn://ulpgc.masterii.moviles"  
  xmlns:ns2="http://xml.apache.org/xml-soap"  
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ns1:QueryCustomersResponse>
<return SOAP-ENC:arrayType="ns2:Map[9]" xsi:type="SOAP-ENC:Array">
  <item xsi:type="ns2:Map">
    <item>
      <key xsi:type="xsd:string">IDCustomer</key>
      <value xsi:type="xsd:string">14</value>
    </item>
    <item>
      <key xsi:type="xsd:string">name</key>
      <value xsi:type="xsd:string">1412413</value>
    </item>
    <item>
      <key xsi:type="xsd:string">address</key>
      <value xsi:type="xsd:string">es</value>
    </item>
  </item>
  <item xsi:type="ns2:Map">
    <item>
      <key xsi:type="xsd:string">IDCustomer</key>
      <value xsi:type="xsd:string">20</value>
    </item>
    <item>
      <key xsi:type="xsd:string">name</key>
      <value xsi:type="xsd:string">Andreíta</value>
    </item>
    <item>
      <key xsi:type="xsd:string">address</key>
      <value xsi:type="xsd:string">Bajo Santo, 9</value>
    </item>
  </item>
</return>
</ns1:QueryCustomersResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Pueden visualizar el contenido de la base de datos en esta dirección:

<http://appstip.iatext.ulpgc.es/ventas/mostrar.php>

### Resultado

Como resultado de la práctica, se entregará un fichero comprimido que contendrá el proyecto de la práctica completo, con todas sus carpetas y archivos, listo para ser usado.