

# Casos Prácticos para Programadores



Documento elaborado por: Francisco José Cantero Villar

Versión: 1.4.1
Lugar y fecha: Sevilla, 28 de Noviembre de 2007

## 1 Casos Prácticos

A continuación se describen una serie de casos prácticos, para una vez comprendida la materia, aplicar los conocimientos recién adquiridos.

### 1) Creación del cliente de Port@firmas a partir del WSDL.

Nota: La sentencia en AXIS para generar las clases clientes a partir del fichero WSDL publicado es:

```
java org.apache.axis.wsdl.WSDL2Java fichero.wsdl
```

### 2) Adaptación del cliente para que lea las propiedades de un archivo de texto.

Nota: Dentro de los fuentes facilitados en el curso hay un ejemplo basado en un patrón proxy de cómo podemos configurar el cliente.

### 3) Crear una clase "Main" que cree una petición de firma dentro de Port@firmas.

Nota: Dentro del mismo código de ejemplo hay un caso de referencia, modificar para que sea enviado a nuestro propio usuario.

### 4) Crear peticiones, con más de un firmante y más de un documento, en modo cascada o paralelo.

Nota: Este ejercicio es muy similar al anterior pero permite observar el distinto comportamiento de la aplicación en un tipo u otro de firma.

### 5) Sobre una petición ya enviada, cambiar uno de los firmantes y eliminar el último.

Nota: Este ejercicio permite ver como se pueden cambiar los firmantes de una petición que aún no ha sido firmada o bien eliminar alguno de ellos.

### 6) Eliminar todos los firmantes de una petición entregada y luego eliminar la petición.

Nota: Este ejercicio permite ver como podemos deshacer un envío que no interesa mantener.

### 7) Crear una jsp que se invoque como retorno de Acción Web.

Nota: Revisar el manual de programadores de Port@firmas para identificar los parámetros que hay que recoger.

## 2 Casos Prácticos Avanzados

Esta relación de ejercicios planteados no profundizan tanto en la lógica de la aplicación sino que busca optimizar los accesos o hacer un control

- 1) Cachear en memoria las listas de usuarios, tipos de documento y estados.

Nota: En caso de usar un objeto estático sería recomendable que estos se actualizaran automáticamente cada cierto tiempo, y los accesos al mismo se hicieran de forma sincronizada.

- 2) Modificar los servicio de Usuarios para que se actualice con otros sistemas de información.

Nota: Se podría sincronizar con una LDAP o para hacerlo más sencillo, con un HashMap que cargue valores de un fichero.

### 3 Historia de versiones

VERSION	FECHA	AUTOR	DESCRIPCIÓN
1.3.0	21/03/2007	FJCV	Primera versión del documento.
1.4.0	19/06/2007	FJCV	Revisión del documento, no hay cambios.
1.4.1	28/11/2007	FJCV	Revisión del documento, no hay cambios.