

P/. icearchiva
D/. **Manual de integración**

Control documental

Versión del documento	Fecha	Autor	Modificaciones/Comentarios
1.0	28/10/2013	keensoft http://www.keensoft.es	Versión inicial del documento

Contenido

1/.Introducción.....	3
2/.Frameworks utilizados.....	3
3/.Generación de la solución.....	4
4/.Módulos de la aplicación.....	4
4.1/.ice-web.....	4
4.2/.ice-auth-filter.....	4
4.3/.Ice-wsdl.....	4
4.4/.Ice-webservice.....	4
4.5/.Ice-tasks.....	4
4.6/.Ice-cmis-client.....	4
4.7/.Ice-commons.....	4
4.8/.Ice-services.....	4
4.8.1/.Ice-services.interface.....	4
4.8.2/.Ice-services.lta.impl.....	5
4.8.3/.Ice-services.aip.impl.....	5
4.8.4/.Ice-services.tsa.impl.....	5
4.8.5/.Ice-services.ks.jks.....	5
4.8.6/.Ice-services.esign.dummy.....	5
4.8.7/.Ice-services.search.dummy.....	5
5/.Elección de componentes mediante profiles.....	5
6/.Propiedades de configuración.....	6
7/.Pruebas de la aplicación.....	6

1/. Introducción

icearchiva es una aplicación de repositorio electrónico que tiene el cometido de almacenar la información de los expedientes del modo adecuado para que su validez legal permanezca en el tiempo.

Tal como se establece en el ENS y ENI explícitamente en su capítulo X “Recuperación y conservación del documento electrónico”, así como en las NTI, se debe garantizar la preservación y la validez jurídica del documento electrónico a lo largo de su vida útil. Si el documento nace electrónico se tiene que preservar electrónicamente con las garantías mencionadas.

Un repositorio digital de larga duración se compone de diferentes sistemas integrados por una aplicación que provee el servicio, como es el caso de icearchiva.

icearchiva gestiona todas las operaciones que debe facilitar un repositorio de larga duración de acuerdo al estándar LTAP y permite al integrador decidir qué sistemas o aplicaciones desea utilizar para cada uno de los difentes servicios.

Los servicios proporcionados por icearchiva son:

- Núcleo de repositorio y procesos automáticos de preservación.
- Servicios web para exponer las operaciones de repositorio de acuerdo al estándar LTAP.
- Autenticación.
- Acceso a repositorios CMIS.
- Soporte multi entidad. (Multitenancy)
- Conexión reduandante a TSA.
- Empaquetado en formato METS.
- Gestión de certificados mediante Java KeyStore.
- Gestión de procesos.

Los servicios mínimos que se requieren son:

- Servicio de firma.
- Servicio de sellos de tiempo. (TSA)
- Repositorio CMIS.
- Servicio de indexación y búsqueda.

pudiendo realizar nuevas implementaciones de los facilitados para adecuarse a las necesidades de la integración.

2/. Frameworks utilizados

Los frameworks utilizados y las diferentes versiones de los mismos pueden ser consultados en el archivo 'pom.xml' del componente raíz 'ice-modules'. Concretamente son:

```
<!-- Dependencies versions -->
<springframework.version>3.2.3.RELEASE</springframework.version>
<slf4j.version>1.6.1</slf4j.version>
<slf4j-log4j12.version>1.7.5</slf4j-log4j12.version>
<log4j.version>1.2.16</log4j.version>
```

```
<junit.version>4.11</junit.version>  
<jetty.version>6.1.26</jetty.version>  
<cxf.version>2.7.4</cxf.version>  
<wss4j.version>1.6.10</wss4j.version>  
<bouncycastle.version>1.46</bouncycastle.version>  
<opencmisVersion>0.8.0</opencmisVersion>  
<opencmisBindingVersion>0.8.0</opencmisBindingVersion>
```

3/. Generación de la solución

La aplicación se genera mediante la herramienta MAVEN.

Para generar el 'war' de icearchiva, ejecutar desde el raíz del proyecto:

```
mvn clean package -DskipTests=true -Ploc,tomcat
```

Donde 'loc' es el entorno para el cual se cogerán los archivos de propiedades y 'tomcat' el servidor.

4/. Módulos de la aplicación

4.1/. ice-web

Módulo que crea la aplicación web. Contiene el archivo de propiedades para todos los módulos.

4.2/. ice-auth-filter

Filtro de la aplicación web para acceso a las rutas de los webservices. Se facilita interfaz e implementación dummy para poder personalizarlo.

4.3/. Ice-wsdl

Contiene el WSDL del servicio web.

4.4/. Ice-webservice

Contiene los web services que con los que icearchiva proporciona su servicio.

4.5/. Ice-tasks

Contiene las tareas de mantenimiento del repositorio.

4.6/. Ice-cmis-client

Contiene el cliente de acceso al servidor CMIS.

4.7/. Ice-commons

Contiene clases comunes al resto de módulos, como las encargadas de gestionar la 'Multientidad'.

4.8/. Ice-services

Contiene los servicios propiamente de repositorio LTA.

4.8.1/. Ice-services.interface

Contiene los interfaces que se deben implementar para generar el nucleo de la aplicación:

- AIPService – Servicio de empaquetado de la importación.
- ESignService – Servicio de firma.
- KeyStoreService – Servicio de almacenamiento de certificados.
- LtaService – Servicio de archivo longevo.
- SearchService – Servicio de búsqueda .
- TsaService – Servicio de sellos de tiempo.

4.8.2/. Ice-services.lta.impl

Implementación facilitada para el servicio de archivo longevo.

4.8.3/. Ice-services.aip.impl

Implementación facilitada para el empaquetado de los paquetes. Se utiliza el estandar METS.

4.8.4/. Ice-services.tsa.impl

Implementación facilitada para consulta de TSAs para la obtención de sellos de tiempo.

4.8.5/. Ice-services.ks.jks

Implementación facilitada para la gestión de certificados mediante un keystore de Java.

4.8.6/. Ice-services.esign.dummy

Implementación de firma vacía. Deberá generarse la implementación apropiada para el sistema de firma que se desee usar.

4.8.7/. Ice-services.search.dummy

Implementación de servicios de búsqueda vacía. Deberá generarse la implementación apropiada para el servicio de indexación y búsqueda que se desee usar.

5/. Elección de componentes mediante perfiles

Para la selección de los componentes que formarán parte de la aplicación, se deben informar estos en el archivo 'spring-profiles.properties' del módulo 'ice-web'. Por ejemplo:

```
spring.profiles=auth-dummy,esign-dummy,task-java,ks-local,search-dummy
```

La definición del perfil al que pertenece cada uno de los componentes se hace en la definición del Bean de Spring de cada uno de los módulos. Por ejemplo, en el módulo 'ice-services.search.dummy':

```
<?xml version='1.0' encoding='UTF-8'?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd">

    <beans profile="search-dummy">
        <bean id="searchService"
              class="org.icearchiva.lta.service.search.DummySearchService" />
    </beans>
```

</beans>

6/. Propiedades de configuración

Las propiedades de configuración de cada uno de los módulos se incluyen todas en el archivo 'ice-archiva.properties' del componente 'ice-web'.

7/. Pruebas de la aplicación

En el componente 'ice-web' se dispone de clases de TEST para probar la combinación de implementaciones seleccionada.