



CDOCS (COMPRUEBA DOCUMENTOS)

CDOCS (Comprueba Documentos) es una aplicación divulgativa, de escritorio, gratuita y de Código Abierto. Para **Windows** y **Mac OS**, con el motor de ejecución **Adobe® AIR®** (**AIR**, es un acrónimo de *Adobe Integrated Runtime*).

Mediante el cálculo del dígito o dígitos de control, pretende comprobar la consistencia o veracidad de los números de algunos documentos de identificación personal, fiscal y laboral, de la Administración española. También la de algunos códigos de identificación financiera, en el ámbito internacional.

Los resultados ofrecidos por la aplicación son sólo orientativos, nunca deben considerarse exactos ni mucho menos vinculantes.

• Base de datos:

CDOCS incluye una base de datos **SQLite** (archivo, **entidades_financieras_ES.db**) con datos **públicos** sobre las Entidades Financieras con establecimiento en España, facilitados periódicamente por los **Registros de Entidades** del Banco de España.

• Programación:

CDOCS está programada con **lenguajes estándar del desarrollo Web**: **HTML**, **CSS**, **JAVASCRIPT** de propósito general y específico para la **API** de **AIR**. También **XML** y **ActionScript**.

Fue compilada, firmada y empaquetada para su distribución con **Adobe® Flex® 4.6 SDK**.

Puede utilizarse **Adobe® AIR® SDK**. Si no se considera necesario modificar el archivo **alivePDFlib.swf** que contiene algunas *clases* de la librería **ALIVEPDF**.

Adobe AIR utiliza **WebKit**, un motor de navegación (de código libre) que incluye, entre otros, un **analizador sintáctico de HTML** y un **intérprete de JAVASCRIPT**.

HTML

El archivo **index.html** (página principal), en el directorio raíz de aplicación y los incluidos en el directorio **/html** contienen esta clase de código, tipificado para **HTML5** `<!DOCTYPE html>` sólo en el caso de páginas completas.

Los archivos del subdirectorio **/forms**, encierran fragmentos que serán introducidos de forma interactiva en el documento principal mediante técnicas **AJAX**.

HOJA DE ESTILO CSS:

Los archivos **index.css** y **humane-original-foog.css** contienen el código **CSS₃** convencional.

En la programación **CSS** conviene definir algunas propiedades con la **sintaxis exclusiva para WebKit como proveedor**, para conseguir los efectos deseados.

JAVASCRIPT con la API (*Application Programming Interface*) de AIR:

En el archivo principal (comentado) **index.js**, se encuentra el código **JAVASCRIPT** con algunas de las propuestas de la extensa interfaz de **AIR**, como por ejemplo:

- Creación, posicionamiento y manejo de la **ventana nativa principal** y de **ventanas secundarias**, de los tipos “normal” y “utilidad”.
- Construcción de **menús**, emergente en el caso del principal y contextuales asociados a los iconos que aparecen en la bandeja del sistema o en el **“Dock” (Mac OS)**.
- Lectura de la **base de datos local**.
- Lectura del archivo descriptor (**XML**) y de otros archivos con este mismo formato.
- **Monitorización del estado de Internet** (precisa de **servicemonitor.swf**)
- Llamadas al **navegador (por defecto) instalado en el equipo** para la presentación de las páginas **web** (externas).
- Gestión del **sistema de ficheros**.
- **Apertura de archivos** con su programa determinado por defecto en el sistema.
- **Actualizaciones** desde un servidor remoto (descarga e instalación del paquete instalador **[.air]** con la versión solicitada).

- Visor local de **Código Fuente** (con **AIRSourceViewer.js**).
- Características “**capacidades**” del *software*, en el equipo del usuario.
- **Eventos** de conexión, error, progreso de descarga y estado de ventanas.

... etc.

Además del citado **index.js** se añade el archivo auxiliar **index-aux.js** con código **JAVASCRIPT** convencional, separado del principal (**index.js**) por motivos de legibilidad.

Recursos JAVASCRIPT de AIR incluidos en la aplicación:

Con el mencionado **AIRSourceViewer.js**, incluimos también: **AIRAliases.js**. Las clases *runtime* están organizadas en una estructura de paquete, y definidas con la siguiente sintaxis:

```
window.runtime.flash.desktop.NativeApplication
```

AIRAliases.js proporciona definiciones de “**alias**” que permiten acceder a las clases *runtime* sin tener que escribir definiciones tan largas. Por ejemplo, el acceso a la clase citada arriba, se consigue escribiendo solamente:

```
air.NativeApplication
```

En tiempo de desarrollo conviene utilizar **AIRIntrospector.js**, que ofrece numerosas funciones útiles y una interfaz gráfica para ayudar en la construcción y **depuración** de aplicaciones basadas en **HTML**.

Importante: Es necesario eliminar el acceso a **AIRIntrospector.js** y el código relacionado con él, antes de empaquetar y distribuir la aplicación.

servicemonitor.swf, **AIRSourceViewer.js**, **AIRAliases.js** y **AIRIntrospector.js**. Se encuentran dentro de los kits **AIR** y **Flex**.

JAVASCRIPT de otros autores:

humane.js junto con **humane-original-foog.css**. Proporciona avisos emergentes del tipo “**toast**”. De Marc Harter.

<http://wavded.github.com/humane-js/>

iban.js, para la verificación del código internacional **IBAN**.

<https://github.com/arhs/iban.js/tree/master>

- **humane.js** , **iban.js** y **ALIVEPDF**. Están publicados con **licencia MIT**.

PDF con **ALIVEPDF** (*ActionScript 3 Open-Source PDF Library*):

El archivo **alivePDFlib.swf**, tiene como origen el código **ActionScript** del guión **alivePDFlib.as**.

alivePDFlib.swf, junto con su correspondiente “envoltura” **JAVASCRIPT**, **alivePDFWrapper.js** y **alivePDFgenerator.js**. Permite la creación local de documentos con formato **PDF** que contienen entre otra información, los resultados obtenidos por la aplicación, en cada caso.

El código y el proceso de compilación del archivo **alivePDFlib.swf** se ha desarrollado siguiendo las sugerencias que aparecen en [este artículo](#) de Murray Hopkins.

Práctica rápida (más o menos) para conseguir un recurso **[.swf]** con algunas clases de **ALIVEPDF** para generar documentos **PDF** con contenido variable:

- Descargar [la última versión estable de las herramientas ALIVEPDF](#) (**AlivePDF 0.1.5 RC.zip**, cuando se edita este documento) y extraer el contenido del paquete.
- En el directorio **/bin** de **Adobe Flex SDK** creamos una carpeta con el nombre: **alivepdflib**.
- En el directorio **Sources/bin** del paquete **ALIVEPDF**, encontramos el archivo: **AlivePDF.swc**. Un archivo **SWC** es similar a **zip** (empaquetado y ampliado mediante el formato de archivos **PKZIP**) que genera la herramienta de edición de **Flash**.
- Copiar **AlivePDF.swc** y pegar en la carpeta **alivepdflib** que antes hemos creado en el directorio **/bin** de **Adobe Flex SDK**.
- Copiar la carpeta **assets** que (en este caso) contiene las imágenes (formato **.jpg**) y fuentes de letra (formatos **.ttf** y **.afm**) que deseamos “insertar” en el archivo final, en el directorio **/bin** de **Adobe Flex SDK**.
- En el mismo directorio (**/bin** de **Adobe Flex SDK**) incorporar también el archivo (**.as**) con el código **ActionScript** que señala los paquetes **flash** y **alivepdf** necesarios y los métodos **alivepdf** deseados. Para la importación y ensamblado final (el nombre de este archivo, con extensión **[.as]** será idéntico al de la *clase* principal programada en él). En el [código fuente de CDOCS](#), este archivo recibe el nombre de **alivePDFlib.as** y reside en el directorio **/as**.
- Después, desde la **línea de comandos** y situados en el directorio **/bin** de **Adobe Flex SDF** ejecutar el comando:

```
$ amxmlc -library-path+=alivepdflib alivePDFlib.as
```

Donde **alivepdflib** es la carpeta que contiene el archivo **AlivePDF.swc** y **alivePDFlib.as**, el guión **ActionScript**. (la carpeta **assets**, con las imágenes y

fuentes de letra y el archivo **alivePDFlib.as**, están situados dentro del directorio **/bin** de la estructura **Adobe Flex**, pero fuera de la carpeta **alivepdflib**, en este ejemplo).

Si no hay errores, la utilidad **amxmlc** (*MXML compiler*) genera el archivo **alivePDFlib.swf** con las librerías **Flex** y **AlivePDF** necesarias para construir documentos **PDF** “al vuelo”.

- Por último, (en nuestro caso) hay que crear una “superficie de contacto”, interfaz **JAVASCRIPT**, importando la librería **alivePDFlib** y reescribiendo los métodos adecuados, en este lenguaje, para conseguir el funcionamiento esperado (ver archivo **alivePDFWrapper.js**).

Para visualizar documentos **PDF** desde una aplicación **AIR**, es necesario que el equipo disponga de una versión actualizada (8.1 ó superior) del visor estándar (genuino) **Adobe READER**.

XML

El archivo descriptor: **cdocs-app.xml** (manifiesto).

Todas las aplicaciones de **AIR** requieren un **archivo descriptor**. El archivo descriptor es un documento **XML** especial que define las propiedades básicas de la aplicación.

CDOCS contiene otros archivos **XML** convencionales, como **version.xml** que sirve de registro de versiones (actualizaciones).

Código fuente de **CDOCS** (Versión 0.1.0 Beta) .Desde el siguiente enlace, en **GitHub**:



<https://github.com/fooghub/Cdocs-HTML-Javascript-AIR>

Puedes obtener la totalidad del código fuente para la producción de esta aplicación (incluye el paquete instalador de distribución, **cdocs.air**).

• Desarrollo de una aplicación AIR con HTML y JAVASCRIPT:

Herramientas necesarias:

JAVA Development Kit (**JDK**) o **JAVA** Runtime Environment (**JRE**).
Plataforma **JAVA** estándar para desarrolladores y motor de ejecución **JAVA** para usuarios finales, respectivamente. Que puedes descargar desde la página oficial de **Oracle**:

<http://www.oracle.com/technetwork/java/javase/downloads/>.

Adobe® AIR® SDK ó **Adobe® Flex® SDK**. Conjunto de herramientas de desarrollo de software para aplicaciones **AIR**.

Documentación:

Adobe proporciona una documentación, extensa y detallada, sobre el desarrollo de aplicaciones **AIR** para escritorio (**Windows y Mac**) o dispositivos móviles (**Android e iOS**):

[Adobe AIR documentation](#) (documentación).

[Adobe AIR FAQ](#) (preguntas frecuentes).

Una guía, en castellano y formato PDF: [Creación de aplicaciones de ADOBE AIR](#).

Estructura básica de una aplicación AIR con HTML y JAVASCRIPT:

El **archivo descriptor (XML)** y el **archivo principal (HTML ó SWF)** son los únicos estrictamente necesarios en este tipo de aplicaciones.

Como ejemplo, una plantilla comentada del **archivo descriptor: descriptor-sample.xml**, se encuentra en el directorio **samples** de los kits de desarrollo de software de **AIR** y **Flex**.

El **archivo descriptor** puede tener cualquier nombre. Al combinar la aplicación en un paquete, el nombre de este archivo cambia al de, **application.xml** y se coloca en un directorio especial en el paquete de distribución.

La etiqueta `<content>index.html</content>` del archivo descriptor encierra la ruta hasta el archivo principal de la aplicación, **index.html**, en este caso.

Lanzamiento en modo de pruebas de una aplicación AIR:

Para mayor comodidad, conviene que la variable del sistema **Path** (Variables de Entorno) contenga la **ruta** adecuada hasta el directorio **bin** de **Adobe AIR SDK** o **Adobe Flex SDK**.

Utiliza **AIR Debug Launcher (ADL)** para ejecutar (en modo de prueba) tanto aplicaciones basadas en **SWF** como las basadas en **HTML** durante la fase de desarrollo. Con **ADL** se puede ejecutar (con todas sus funciones operativas) una aplicación sin primero tener que empaquetarla e instalarla. De forma predeterminada, **ADL utiliza un motor de ejecución incluido con el SDK**, con lo cual no se necesita instalar el motor de ejecución por separado para utilizar **ADL**.

La orden en línea de comandos es:

```
$ adl cdocs-app.xml
```

Donde **cdocs-app.xml**, es el archivo descriptor de la aplicación.

Distribución de una aplicación AIR:

Firma digital de archivos de AIR:

Todos los archivos (paquetes de instalación) **AIR** [.air], **deben incluir necesariamente una firma digital** que identifica a su editor.

La **firma digital** se puede producir con un certificado proporcionado por una entidad emisora de certificados (**AC**) reconocida (opción comercial con un determinado coste económico) o con un certificado **no vinculado a ninguna entidad de verificación reconocida**.

Para la última alternativa citada, **AIR** ofrece la posibilidad de crear un **certificado de firma automática**, (con una validez de cinco años) con la herramienta multiusos: **ADT (AIR Developer Tool)** incluida en el **SDK** (precisa de **Java 1.5** o superior).

El formato (mínimo) de la orden, en línea de comandos, para generar un certificado de firma automática, es el siguiente:

```
$ adt -certificate -cn nombre-del-editor tipo-de-clave archivo-de-salida *contraseña*
```

El **tipo-de-clave** que se va a utilizar para el certificado puede ser **1024-RSA** o **2048-RSA**.

Por ejemplo:

```
adt -certificate -cn Foog.Software 1024-RSA FoogCert.p12 contraseNna
```

Los paquetes AIR admiten también certificados de firma digital **OpenSSL**.

La orden anterior genera el archivo **FoogCert.p12** (*Personal Information Exchange*), que utilizaremos en el paso siguiente.

Empaquetado de la aplicación:

Con **ADT** y su opción **-package** podemos combinarr un paquete de distribución e instalación con nuestro proyecto **AIR**. La sintaxis del comando, es:

```
$ adt -package -storetype pkcs12 -keystore  
[ruta/hasta/archivo/'.p12'] [nombre-del-paquete '.air']  
[archivo-descriptor] [archivo-de-contenidos] [archivos-o-  
directorios-auxiliares].
```

En nuestro ejemplo, la orden en línea de comandos será:

```
$ adt -package -storetype pkcs12 -keystore FoogCert.p12  
cdocs.air cdocs-app.xml index.html css db html images js md pdf  
swf xml
```

Durante el proceso de compilación **ADT** solicitará la entrada de la **contraseña** del certificado de firma.

La conclusión es **cdocs.air**, un paquete funcional e instalable en cualquier equipo **Windows** o **Mac** que disponga de **Adobe AIR** (*runtime*).

Licencias:

CDOCS es una aplicación divulgativa, sin ánimo comercial. La idea, las imágenes, los textos, el código elaborado por **Foog.Software**, el de otros (mencionados en el apartado “**JAVASCRIPT de otros autores**”) y **ALIVEPDF** se publican con una licencia de dominio público, **MIT License (MIT)**.

Los conjuntos de herramientas **Adobe AIR SDK** y **Adobe Flex SDK**, contienen **software propietario** (aunque gratuito) y su utilización está sujeta a las condiciones de una licencia corporativa de su autor, **ADOBE SYSTEMS INCORPORATED**. [Adobe AIR SDK - Warranty disclaimer and license agreement](#).

2004 – 2016 Foog.Software

Marzo 2016

www.foog.es