Astenor S.L.

Manual de Instalación

PROYECTO FICO

Octubre de 2010



Manual de Instalación

Índice

1 Introducción....4
2 Requerimientos para el despliegue....5
2.1 Equipo de Escaneo:....5
2.2 Equipo Servidor:....5
3 Instalación desde los fuentes....7
3.1 Código fuente y configuración....9
3.1.1 CAS / Single Sign On....10
3.1.2 FICO....10
3.1.3 Alfresco....11
3.1.4 LDAP....12
3.1.5 Otros....13
3.2 Compilar el código fuente....13
3.2.1 Compilación del cliente de Upload WEB....14
3.3 Instalación del servidor LDAP....15
3.4 Instalación del Servidor Web....20
3.4.1 Generación de certificados:....20
3.4.2 Configuración de módulos y sitios web....21
3.5 Instalación de la Base de Datos....22
3.6 Puesta a punto de las Bases de Datos....23
3.6.1 Alfresco....23
3.6.2 CAS 23
3.6.3 FICO....23
3.7 Arranque inicial del servicio....24
3.8 Instalación del cliente WEB de carga de ficheros....25

3.9 Instalación de la aplicación de OMR.....26
3.9.1 Instalación del software Opscan Insight.....26
3.9.2 Instalación de la aplicación de escaneado.....30
3.9.3 Peculiaridades de la instalación/configuración.....31
3.10 Instalación de LimeSurvey.....31

1 Introducción

Este documento pretende servir de guía para el despliegue de la aplicación FICO. Para más información sobre el proyecto y los detalles técnicos del aplicativo se recomienda la lectura del documento "Memoria Técnica" del proyecto.

FICO es un proyecto de código abierto basado en tecnologías y librerías del mismo tipo. Por este motivo, la documentación y código de las librerías de terceros y artículos relativos a los mismos se encuentran disponible para la descarga o lectura desde la página web de dichos terceros. La relación de estos componentes se encuentra detallada en la "Memoria Técnica" del proyecto.

Este documento va dirigido a Administradores de Sistemas o Técnicos de Sistemas y presupone un conocimiento y formación adecuados en la gestión de los sistemas operativos de Microsoft y Linux, así como conocimientos sobre administración de servidores web tales como Apache Web Server, servidores de aplicaciones como Apache Tomcat y bases de datos libres como PosgreSQL. No obstante se intenta describir cada paso de la instalación ejemplo, incluyendo la puesta a punto de estas herramientas sobre las que se supone un conocimiento previo.

Existen muchas maneras distintas de desplegar el aplicativo, al igual que combinaciones de componentes que se pueden reemplazar, por ello se describe un sistema ejemplo de entre los muchos posibles para realizar el despliegue del aplicativo.

2 Requerimientos para el despliegue

En este apartado se describirá la instalación mínima para el aplicativo. El sistema requiere de un mínimo de dos equipos, uno para controlar el escáner y otro como servidor de la aplicación.

2.1 Equipo de Escaneo:

- ✔ Sistema operativo: Windows XP
- ✔ Memoria RAM: 2 GB
- ✔ Procesador: Pentium Core Duo o AMD
- ✔ Disco duro: 40 GB
- ✔ Software requerido:
  - Scantools v4.60100
  - Scantools Plus App Dev: v7.10100

2.2 Equipo Servidor:

- ✔ Sistema Operativo: Cualquier Linux de 64 bits o MS Windows 2008.
- ✔ Memoria RAM: mínimo 2 GB, recomendado 4 GB en adelante.
- ✔ Procesador: mínimo 2 x Pentium Core Duo, recomendado Intel Quad Core. Con soporte para 64bits.
- ✔ Disco duro: 40 GB mínimo
- ✔ Software requerido junto al sistema operativo:
  - Apache Web Server 2.2.X o Nginx. (Módulos de apache requeridos: ssl, rewrite, expires, deflate, jk, php).
  - PHP 5.X.
  - OpenLDAP 2.4.X
  - PostgreSQL 8.3 o superior.

El resto del software necesario para instalar el aplicativo se instala desde la distribución de código fuente del proyecto, descargando algunos componentes desde Internet.

Otras opciones de despliegue que se podrían contemplar:

- ✔ Servidor exclusivo para base de datos
- ✔ Separar Alfresco y FICO

Los requerimientos del hardware varían en función del número de usuario concurrentes y del almacenamiento en disco requerido por Alfresco como gestor documental.

3 Instalación desde los fuentes

En este apartado se describirá como instalar y configurar el aplicativo a partir del código fuente. Se escoge Debian 5.0/Lenny como sistema operativo para el desarrollo y sobre el que se basa el ejemplo, pero puede portarse sin grandes quebraderos de cabeza a otras distribuciones de Linux o a MS Windows.
Esta parte de la documentación es bastante técnica y supone que el lector tiene unos conocimientos suficientes sobre Linux, lenguajes de scripting de shell (bash o sh), utilidades del sistema base y de la distribución Debian en concreto.
Supondremos que el sistema operativo base está instalado. Conviene verificar que el servidor se puede gestionar de forma remota y tiene algunos paquetes básicos:

```
apt-get install openssh-server build-essential subversion
```

También deberá existir un entorno de desarrollo para Java instalado en la máquina. Dicho entorno consiste en:

- ✔ JDK 1.6 de Sun instalada
- ✔ Apache Ant instalado
- ✔ Apache Maven instalado

Para ello lo mejor es seguir unos pasos similares a los que siguen:

```
echo "Creación de directorio para binarios Java de terceras partes"
mkdir /opt/java
cd /opt/java
echo "Descarga de binarios"
wget <ruta_binario_jdk_sun_de_mi_plataforma>
wget http://www.apache.org/dist/maven/binaries/apache-maven-2.2.1-bin.tar.gz
wget http://www.apache.org/dist/ant/binaries/apache-ant-1.7.1-bin.tar.gz
echo "Instalación JVM"
```

```
chmod +x <binario_jvm>
./<binario_jvm>
ln -s <directorio_jvm> java-6-sun
echo “Instalación Ant”
tar xvzf apache-ant-1.7.1-bin.tar.gz
ln -s apache-ant-1.7.1 ant
echo “Instalación Maven”
tar xvzf apache-maven-2.2.1-bin.tar.gz
ln -s apache-maven-2.2.1 maven
echo “Configuración del perfil de desarrollo”
echo -e “# Entorno de desarrollo Java” >> /etc/profile
echo -e “JAVA_HOME=/opt/java/java-6-sun” >> /etc/profile
echo -e “ANT_HOME=/opt/java/ant” >> /etc/profile
echo -e “MAVEN_HOME=/opt/java/maven” >> /etc/profile
echo -e “PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$MAVEN_HOME/bin:$PATH” >> /etc/profile
echo -e “export JAVA_HOME ANT_HOME MAVEN_HOME PATH” >> /etc/profile
```

El perfil de desarrollo suele incluirse en los ficheros /etc/profile o /etc/bashrc o ~/.bashrc o ~/.bash_profile, dependiendo de cómo se acceda al equipo y con qué usuario.

Tras estos pasos podemos comprobar si todo ha ido bien:

```
echo “Recargamos el perfil”
source /etc/profile
echo “Comprobaciones”
which java
java -version
which ant
ant -version
which mvn
mvn -version
```

3.1 Código fuente y configuración

El código fuente estará disponible empaquetado en un fichero comprimido o directamente en el repositorio subversion de desarrollo. Se supone que se descomprimirá o efectuará un checkout en un determinado directorio que denominaremos **<code>**.

Los directorios principales del proyecto son:

- **alfresco** : contiene las personalizaciones de Alfresco y el script para **ant** de despliegue de Alfresco.
- **cas** : contiene las personalizaciones de Jasig CAS, el sistema de SSO, y el fichero de proyecto de **maven** pom.
- **conf** : contiene ficheros de configuración de ejemplo para Apache, OpenLDAP y generación de certificados auto-firmados SSL.
- **doc** : contiene manuales y documentación del proyecto.
- **ldif** : contiene un juego de datos de prueba para OpenLDAP
- **limesurvey** : contiene la distribución de limesurvey personalizada para el proyecto, con las modificaciones necesarias en su código fuente para albergar las encuestas de las empresas.
- **omr** : contiene los ficheros de configuración para el escáner OMR.
- **sql** : contiene los scripts de la base de datos de FICO.
- **src**: contiene los fuentes de FICO.

En el propio directorio principal se encontrarán varios documentos README*.txt con información para los desarrolladores, junto al fichero de proyecto maven pom de FICO.

La configuración del proyecto se realiza mediante la edición de un único fichero de propiedades. Para configurar el proyecto se copia el fichero default.properties en un nuevo fichero que debe llamarse build.properties, y se edita convenientemente.

A continuación describiremos las principales opciones de configuración para los diferentes componentes de la aplicación que se deben revisar:

3.1.1 CAS / Single Sign On

- **cas.host**: Nombre de host completo del servidor de Single Sign On. Conviene que sea el nombre de host que emplearán los clientes para acceder a la aplicación.
- **cas.host.url**: URL absoluta de acceso a la aplicación CAS, conviene que contenga un nombre de host en lugar de una IP.
- **cas.database.url**: URL JDBC para acceso a la base de datos, si la base de datos reside en el mismo servidor que las aplicaciones, no es necesario modificar este parámetro.
- **cas.database.user**: nombre de usuario para acceder a la base de datos de CAS.
- **cas.database.password**: contraseña del usuario de base de datos de CAS.

3.1.2 FICO

- **fico.host**: Nombre de host completo del servidor de FICO. Conviene que sea el nombre de host que emplearán los clientes para acceder a la aplicación.
- **fico.host.url**: URL absoluta de acceso a la aplicación FICO, conviene que contenga un nombre de host en lugar de una IP.
- **fico.secure.host.url**: URL absoluta de acceso a la aplicación FICO, mediante protocolo seguro.
- **fico.database.url**: URL JDBC para acceso a la base de datos, si la base de datos reside en el mismo servidor que las aplicaciones, no es necesario modificar este parámetro.
- **fico.database.user**: nombre de usuario para acceder a la base de datos de FICO.

- **fico.database.password**: contraseña del usuario de base de datos de FICO.

3.1.3 Alfresco

- **alfresco.destination**: ruta completa al directorio donde se instalará Alfresco que viene con una distribución de Apache Tomcat 6.0.X.
- **alfresco.host**: Nombre de host completo del servidor de Alfresco. Conviene que sea el nombre de host que emplearán los clientes para acceder a la aplicación.
- **alfresco.host.url**: URL absoluta de acceso a la aplicación Alfresco, conviene que contenga un nombre de host en lugar de una IP.
- **alfresco.api.url**: URL absoluta de acceso a los servicios web de Alfresco.
- **alfresco.api.user**: Nombre del usuario administrador de Alfresco. Debería ser "wsagent" y no debe cambiarse este valor.
- **alfresco.api.password**: Contraseña del usuario administrador de Alfresco, es la contraseña que se empleará para instalar Alfresco. Durante la vida de la aplicativo no es conveniente cambiar la contraseña de este usuario dado que se emplea para acceder mediante servicios web y afecta a la configuración de varios módulos de la aplicación. El mejor momento para escoger una contraseña es antes de realizar la instalación.
- **alfresco.database.url**: URL JDBC para acceso a la base de datos, si la base de datos reside en el mismo servidor que las aplicaciones, no es necesario modificar este parámetro.
- **alfresco.database.user**: nombre de usuario para acceder a la base de datos de Alfresco.
- **alfresco.database.password**: contraseña del usuario de base de datos de Alfresco.

3.1.4 LDAP

- **ldap.url**: URL de acceso al servidor LDAP.
- **ldap.bind.dn**: DN del usuario con acceso de escritura y permisos para modificar entradas LDAP como administrador.
- **ldap.bind.password**: constraseña del usuario administrador de LDAP.
- **ldap.user.search.base**: RDN de la rama LDAP donde se encuentran definidos los usuarios. Se emplea para realizar búsquedas.
- **ldap.group.search.base**: RDN de la rama LDAP donde se encuentran definidas las organizaciones. Se emplea para realizar búsquedas.
- **ldap.user.format**: formato DN de los usuarios en LDAP. Se emplea para realizar búsquedas.
- **ldap.group.format**: formato DN de las organizaciones en LDAP. Se emplea para realizar búsquedas.

3.1.5 Otros

- **user.language**: idioma codificado ISO 639, empleado para fijar el idioma por defecto de la aplicación.
- **user.country**: país codificado ISO 3166, empleado para fijar el idioma por defecto de la aplicación.
- **mail.host**: nombre completo o IP del servidor de correo SMTP.
- **log4j.mail.host**: nombre completo o IP del servidor de correo empleado para enviar los mensajes de error de aplicación.
- **log4j.mail.to**: dirección de correo electrónico del usuario o grupo de usuarios responsables de administrar y mantener a aplicación.
- **log4j.mail.from**: dirección de correo electrónico del remitente de los mensajes de error de aplicación.

3.2 Compilar el código fuente

Una vez configurado el código fuente, se puede proceder a su compilación y preparación para su uso. Para ello es necesario que el sistema tenga acceso a

internet y funcione la resolución de nombres DNS, debido a que se descargan paquetes desde Internet para generar el aplicativo. Las instrucciones necesarias para compilar el código fuente son:

```
echo “Posicionamiento en el directorio de fuentes”
cd <ruta_al_directorio_de_fuentes>
echo “Compilación de FICO”
mvn package
echo “Compilación de CAS”
cd cas
mvn package
echo “Compilación de Alfresco”
cd ../alfresco
ant deploy
echo “Distribución de aplicaciones con el Tomcat de Alfresco”
cd ..
ln -s target/fico <ruta_despliegue_alfresco>/tomcat/webapps/fico
ln -s target/cas <ruta_despliegue_alfresco>/tomcat/webapps/cas
echo “Script de inicio de servicio”
cp alfresco/target/conf/etc/init.d/tomcat.sh /etc/init.d
update-rc.d tomcat.sh defaults 50
```

3.2.1 Compilación del cliente de Upload WEB

La compilación del cliente web se efectúa por separado del código fuente de la aplicación. Mediante el comando mvn assembly:assembly se crea un fichero .jar que se deberá de copiar en el ordenador que contendrá al aplicación de escaneo OMR.

```
mvn assembly:assembly
[INFO] Building jar: target/fico-upload-client.jar
```

3.3 Instalación del servidor LDAP

Para instalar OpenLDAP en Debian es necesario ejecutar:

```
apt-get install slapd ldap-utils
dpkg-reconfigure slapd
```

Durante ese proceso un instalador para consola preguntará ciertos parámetros de configuración. Conviene crear un usuario de administración de LDAP, y a ser posible que no se llame admin. Para más información es interesante leer los artículos:

- ✔ http://wiki.debian.org/LDAP/OpenLDAPSetup
- ✔ http://www.debuntu.org/ldap-server-and-linux-ldap-clients

Si queremos ajustar la dirección IP en la que escucha el servidor de LDAP, deberemos editar el fichero /etc/default/slapd y modificar la variable SLAPD_SERVICES tras lo que necesitaremos volver a arrancar el servicio:

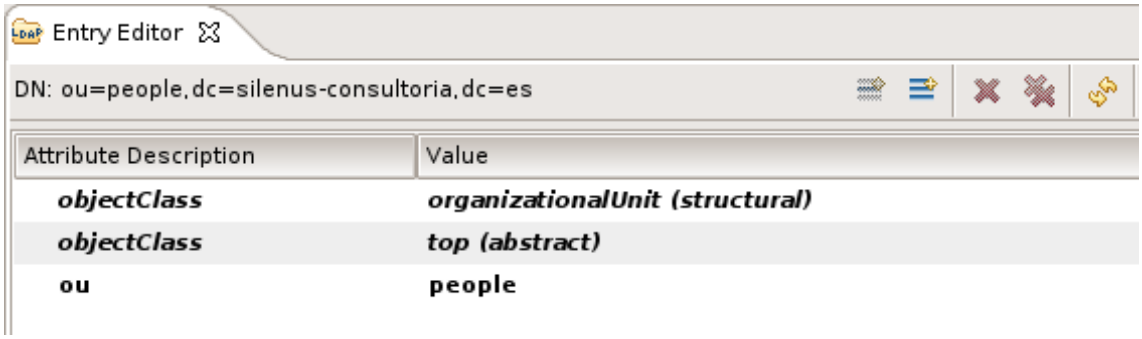
```
/etc/init.d/slapd restart
```

Ahora podemos comprobar que todo funciona correctamente con el siguiente comando, en el que reemplazaremos adecuadamente los valores por los apropiados a nuestro caso:

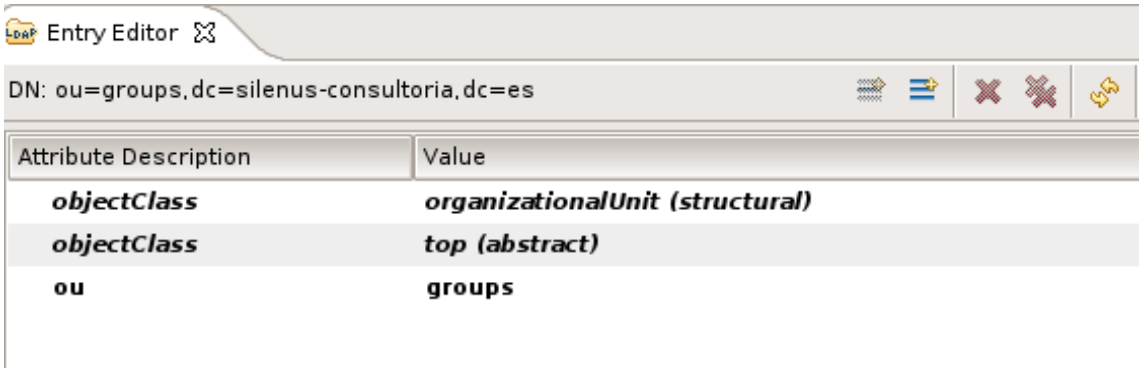
```
ldapsearch -v -H 'ldap://localhost' -b "dc=silenus-consultoria,dc=es" -x  -D
"cn=manager,dc=silenus-consultoria,dc=es" -w <password>
```

El siguiente paso es preparar los datos mínimos para que cuando se arranque el servicio del aplicativo, todo funcione correctamente. Para esto, recomendamos emplear un cliente de LDAP para editar el directorio. Un posible cliente para distintos servidores LDAP es Apache DS, que dispone de versiones para diferentes plataformas y sistemas operativos. Los datos mínimos que deberemos añadir son:

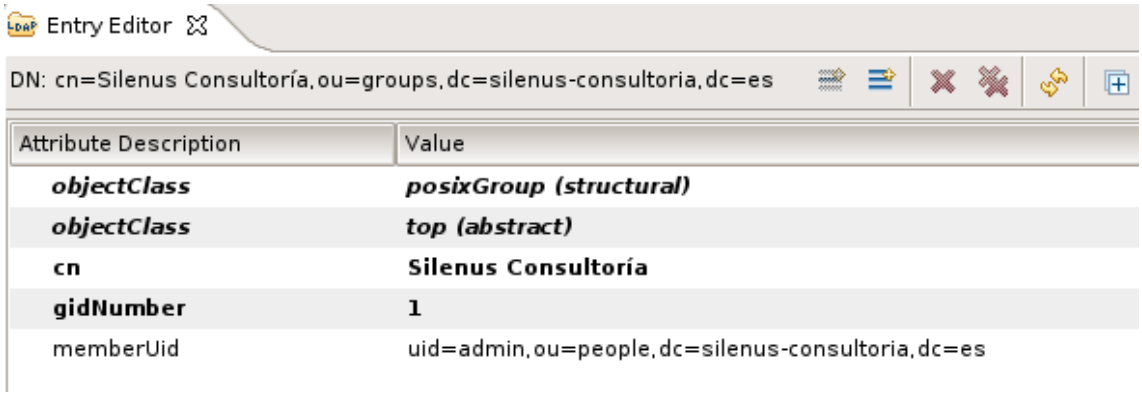
- ✔ Unidad organizativa para usuarios



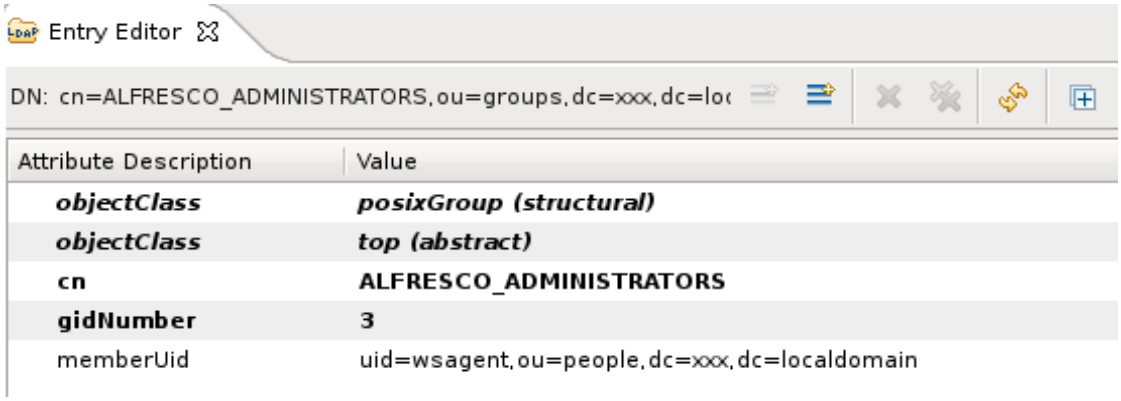
- Unidad organizativa para grupos



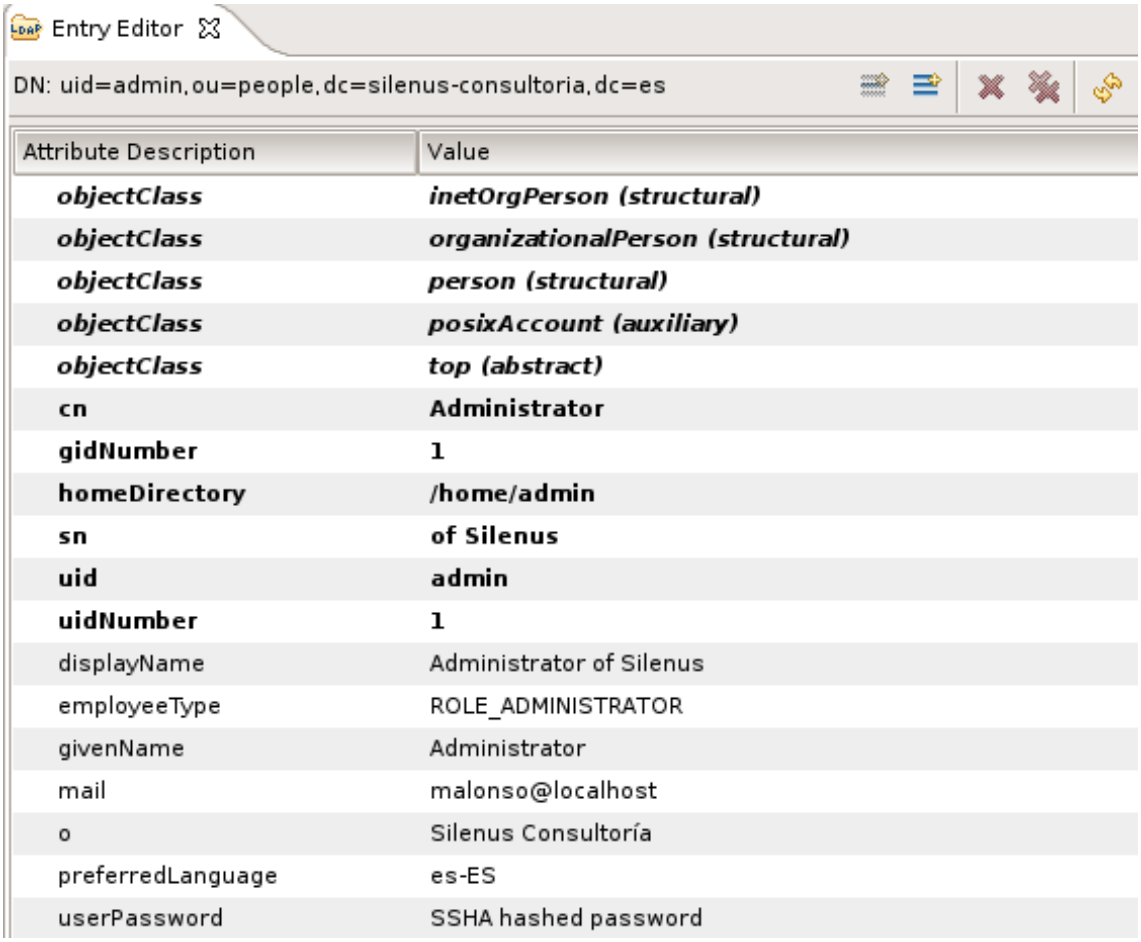
- Organización de administradores globales de la plataforma.



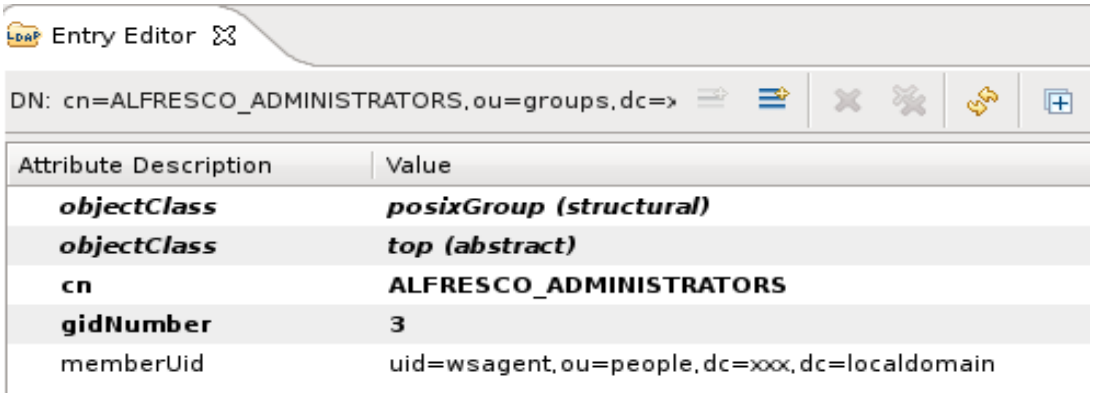
- Organización de administradores de Alfresco.



- ✔ Usuario administrador global de la plataforma.



- ✔ Usuario de Alfresco para Web Services.



En este punto tendremos un servidor de ldap básico instalado y configurado, pero es necesario terminar su configuración. El montaje que nos interesa es el siguiente: tener un servidor de LDAP como repositorio de usuarios y organizaciones y permitir a los propios usuarios puedan modificar sus contraseñas, y al usuario administrador darle permisos de modificación sobre los datos relevantes. **La contraseña empleada para el usuario administrador deberá ser idéntica a la de configuración del usuario de alfresco.**

Para configurar slapd, el demonio de OpenLDAP, podemos inspirarnos en el fichero /etc/ldap/slapd.conf y añadirle una serie de directivas.
Lo siguiente configura los tipos de contraseñas por defecto y el nivel de acceso de seguridad al servidor de ldap.

```
# Global
password-hash {SSHA}
disallow bind_anon
sasl-secprops noanonymous,minssf=56
loglevel     256
```

Dentro de la sección específica para nuestro dominio de usuarios añadiremos otros parámetros y configuraremos las ACL. Deberemos incluir en dichas ACL los permisos para el administrador global de la plataforma.

```
# Indice uid
```

```
index        uid eq
# ACL para contraseñas
access to attrs=userPassword,shadowLastChange
        by dn="cn=manager,dc=silenus-consultoria,dc=es" write
       by dn="uid=admin,ou=people,dc=silenus-consultoria,dc=es" write
        by anonymous auth
        by self write
        by * none
# ACL general
access to *
        by dn="cn=admin,dc=silenus-consultoria,dc=es" write
       by dn="uid=admin,ou=people,dc=silenus-consultoria,dc=es" write
        by * read
```

Tras esto, comprobaremos si el usuario que hemos creado tiene acceso:

```
ldapsearch -v -H 'ldap://localhost' -b "dc=silenus-consultoria,dc=es" -x  -D
"uid=admin,ou=people,dc=silenus-consultoria,dc=es" -w <password>
```

3.4 Instalación del Servidor Web

Para instalar el servidor web apache 2.2 en Debian:

```
echo -e “Instalación Servidor Apache”
apt-get install apache2   apache2-doc apache2-mpm-prefork apache2-utils
apache2.2-common libapache2-mod-apreq2 libapache2-mod-auth-plain libapache2-
mod-encoding libapache2-mod-jk libapache2-mod-php5  libapache2-mod-proxy-html
echo -e “Habilitamos los módulos necesarios”
a2enmod ssl
a2enmod jk
a2enmod php5
a2enmod encoding
a2enmod expires
a2enmod headers
a2enmod deflate
echo -e “Reiniciamos Apache”
/etc/init.d/apache2 restart
```

3.4.1 Generación de certificados:

Cuando se instala un sitio web seguro, es necesario disponer de certificados criptográficos para encriptar el tráfico que emplea SSL. Con el fin de facilitar la creación de estos certificados se emplean unos scripts que provienen de la distribución de OpenVPN. Este conjunto de scripts se denomina easy-rsa y está incluida en el código del proyecto.

```
echo -e “Instalación de openssh”
apt-get install openssl
echo -e “Instalación easy-rsa”
mkdir -p /etc/ssl
cp -r <directorio_fuentes>/conf/etc/ssl/easy-rsa /etc/ssl
cd /etc/ssl/easy-rsa
```

En este punto editamos el fichero /etc/ssl/easy-rsa/vars que contiene los parámetros básicos para la generación de certificados y, a continuación, procedemos a la generación de certificados:

```
cd /etc/ssl/easy-rsa
source vars
./clean-all
echo -e “Generación de Autoridad Certificadora”
./build-ca
echo -e “Generación de Certificado de servidor”
./build-key-server <nombre-de-host.completo.org>
echo -e “Certificados para apache”
mkdir -p /etc/apache2/ssl
cp keys/ca.crt keys/<nombre-de-host.completo.org>.* /etc/apache2/ssl
```

Ahora es necesario añadir estos certificados al repositorio de certificados de la JVM:

```
keytool -import -noprompt -v -trustcacerts -alias FICO_CA -file
/etc/apache2/ssl/ca.crt -keystore ${JAVA_HOME}/jre/lib/security/cacerts
-storepass changeit
keytool -import -noprompt -v -trustcacerts -alias FICO_HOST -file
/etc/apache2/ssl/<nombre-de-host.completo.org>.crt -keystore $
{JAVA_HOME}/jre/lib/security/cacerts -storepass changeit
```

3.4.2 Configuración de módulos y sitios web

Algunos módulos de apache requieren configuración, tal es el caso del módulo jk, capaz de comunicar Apache Http con Tomcat mediante el protocolo AJP. Para configurar este módulo y los sitios:

```
cd <directorio_fuentes>
cp alfresco/target/etc/apache2/*.* /etc/apache2
cp  alfresco/target/etc/apache2/mods-available/*.conf /etc/apache2/mods-
available
cp  alfresco/target/etc/apache2/sites-available/*.conf /etc/apache2/sites-
available
ln -s /etc/apache2/mods-available/jk.conf /etc/apache2/mods-enabled/jk.conf
echo “Activación de sitios web”
a2dissite default*
a2ensite default.conf default-ssl.conf
echo “Reinicio del servidor web”
```

```
/etc/init.d/apache2 restart
```

3.5 Instalación de la Base de Datos

Para instalar e inicializar el cluster de postgresql.

```
apt-get install postgresql-8.3 postgresql-contrib-8.3
mkdir -p /var/pgsql/data
chown postgres /var/pgsql/data
su - postgres
/usr/lib/postgresql/8.3/bin/initdb -E utf8 -U postgres -D /var/pgsql/data
exit
/etc/init.d/postgresql-8.3 restart
```

Ahora crearemos un usuario de administración y configuraremos su acceso sin contraseña desde la cuenta local de usuario de administración:

```
su - postgres
createuser -s -d -r -l -e -P sa
exit
echo “Acceso sin prompt de contraseña”
echo “localhost:5432:*:sa:<password>” >> ~/.pgpass
chmod 0666 ~/.pgpass
```

Para verificar que el acceso a la base de datos es correcto:

```
psql -U sa -H localhost template1
```

En caso de cualquier duda o problema, debe consultarse el manual de la base de datos: http://www.postgresql.org/docs/8.3/interactive/index.html

3.6 Puesta a punto de las Bases de Datos

3.6.1 Alfresco

Para crear la base de datos:

```
createdb -E UNICODE -O sa -e -U sa -h localhost alfresco
```

Durante el proceso de arranque de tomcat se rellenarán las tablas y se terminará su configuración.

3.6.2 CAS

Para crear la base de datos:

```
createdb -E UNICODE -O sa -e -U sa -h localhost cas
```

Durante el proceso de arranque de tomcat se rellenarán las tablas y se terminará su configuración.

3.6.3 FICO

Para crear la base de datos:

```
createdb -E UNICODE -O sa -e -U sa -h localhost surveys
createlang -e -U sa -h localhost plpgsql surveys
```

Si queremos cargar la base de datos con un juego de datos de prueba:

```
cd <directorio_de_fuentes>
chmod +x *.sh
./test_model.sh
```

Si queremos crear una base de datos básica, con las tablas limpias a excepción del usuario administrador y la organización de administración:

```
cd <directorio_de_fuentes>
echo "Creación basica de tablas"
psql -U sa -h localhost -f ./sql/create.sql surveys
echo "Creación del modelo de archivado"
psql -U sa -h localhost -f ./sql/archive/archive.sql surveys
echo "Datos mínimos"
psql -U sa -h localhost -f ./sql/data/min-data.sql surveys
```

```
psql -U sa -h localhost -f ./sql/data/councils.sql surveys
echo "Creación del modelo de análisis"
psql -U sa -h localhost -f ./sql/analysis/analysis.sql surveys
psql -U sa -h localhost -f ./sql/analysis/indexes.sql surveys
echo "Creación del modelo de carga"
psql -U sa -h localhost -f ./sql/load/load.sql surveys
```

3.7 Arranque inicial del servicio

En este momento el sistema debería estar correctamente configurado y estaríamos en disposición de arrancarlo. Los servicios de LDAP, PosgreSQL y Apache Web deberán estar arrancados.

```
cd <ruta_despliegue_alfresco>/tomcat/logs
echo “Re-arranque de LDAP”
/etc/init.d/slapd restart
echo “Re-arranque de PostgreSQL”
/etc/init.d/postgresql-8.3 restart
echo “Re-arranque de Apache Web Server”
/etc/init.d/apache2 restart
echo “Arranque de Tomcat”
/etc/init.d/tomcat.sh start
echo “Monitorización de registros de arranque”
tail -f catalina.out
```

Si el arranque ha tenido éxito, indica que la configuración es válida y se han creado las tablas que faltaban para Alfresco y CAS.

Ya puede entrar en la aplicación en la URL que haya configurado con el usuario administrador.

3.8 Instalación del cliente WEB de carga de ficheros

La instalación se efectuará mediante el copiado de los siguientes ficheros a una carpeta en la maquina destino que contendrá el escáner OMR.

- <wrksvn>/target/fico-upload-client.jar
- <wrksvn>/src/main/java/es/silenus/fico/client/*.properties

a continuación se edita el fichero configuration.properties para fijar los valores de configuración del equipo donde está el OMR y donde se encuentra Detecta:

- ***omr.directory*** : Directorio donde se encontraran los ficheros resultado del escaneo.
- **host.url** : Dirección de acceso a Detecta.
- **user.name** : Nombre de usuario para autenticarnos en el sistema.
- **user.password** : Contraseña de acceso al sistema Detecta

Accedemos al directorio y ejecutamos el cliente de carga para verificar su conectividad con el sistema

```
java -Dlog.home=. -Dlog4j.configuration=file:client-log4j.properties -jar
fico-upload-client.jar configuration.properties
```

Con el fin de que se ejecute de forma periódica se crea una tarea en el sistema, bien en windows mediante el planificador de tareas bien en linux mediante el cron.

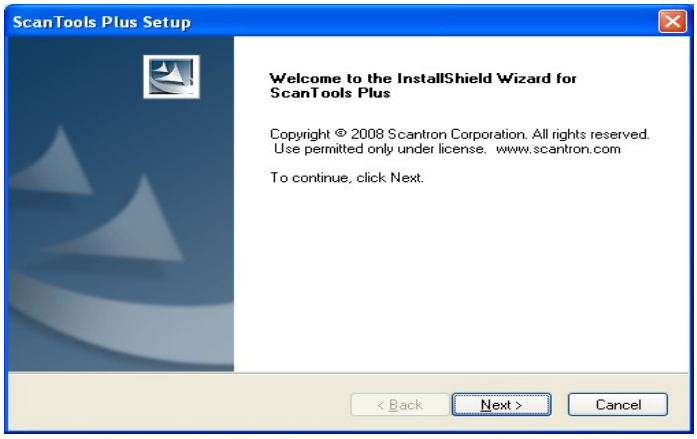
Se aconseja crear un fichero de comandos para facilitar su ejecución y poderlo invocar con comodidad.

3.9 Instalación de la aplicación de OMR

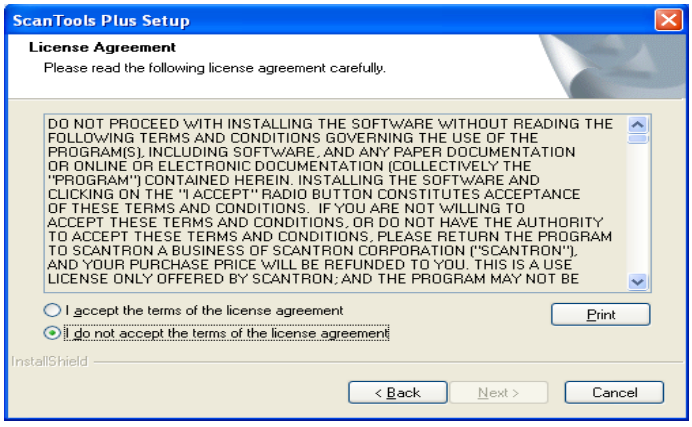
3.9.1 Instalación del software Opscan Insight

Introducimos el cd del aplicativo en el lector CD del equipo donde estará conectado el scanner. Accedemos a él e iniciamos la instalación del software invocando el programa setup.exe.

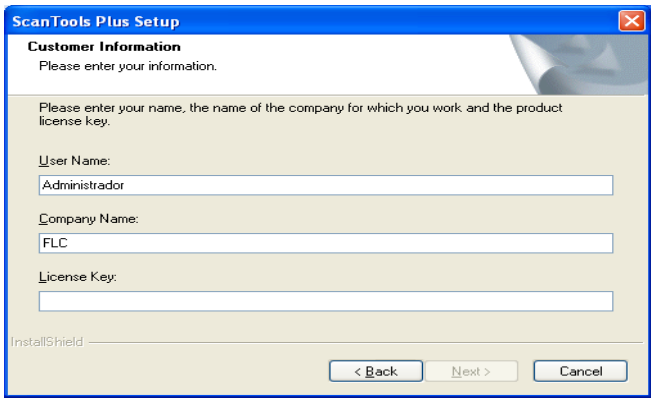
Aparecerá una pantalla de bienvenida que nos indicará que se procederá a la instalación.



El siguiente paso consiste en la aceptación de la licencia del programa.

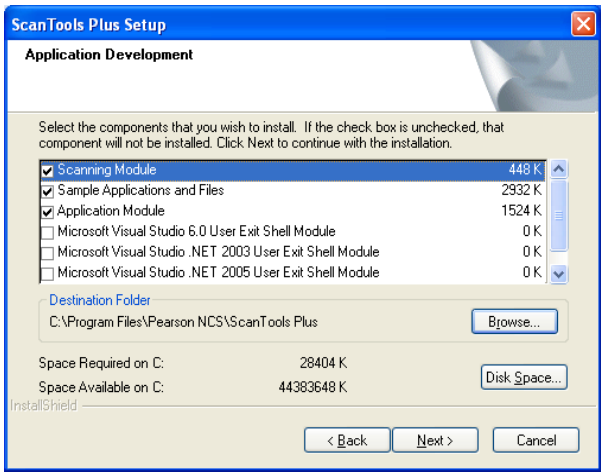


Una vez aceptada se nos pide que introduzcamos unos datos basicos de identificación y el número de licencia que viene pegado en la bolsa donde está almacenado el CD.

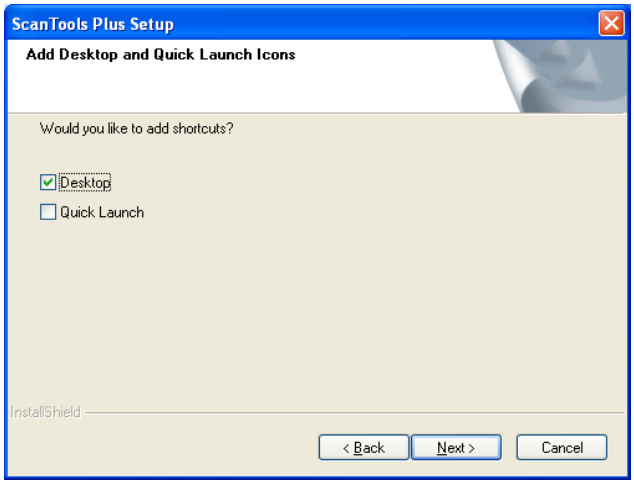


Confirmamos que los datos son correctos y pasamos a la selección de los elementos a instalar.

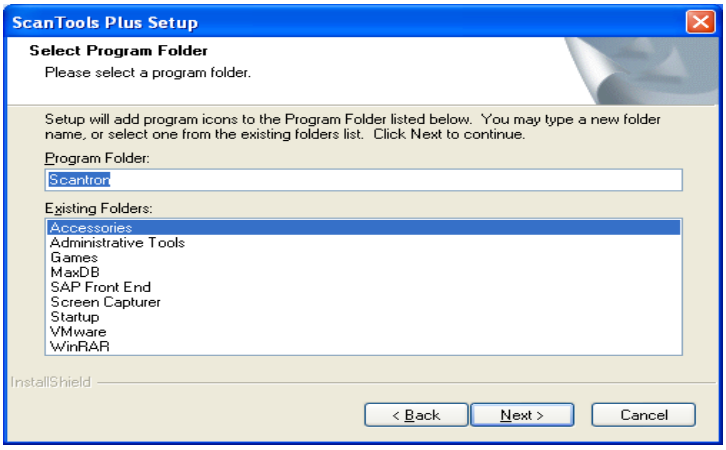




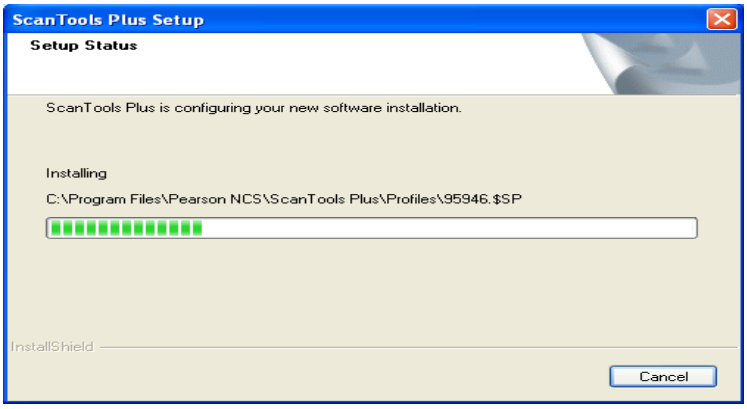
El siguiente paso nos pregunta si deseamos añadir un acceso directo en el escritorio y/o en la barra de inicio rápido. Por comodidad indicamos que añada el acceso directo al escritorio.



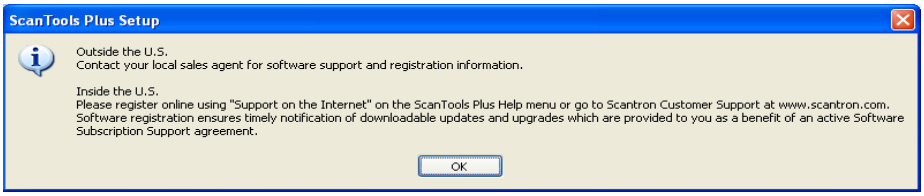
A continuación introducimos el grupo de programas donde pondremos la aplicación. Podemos asignar uno ya creado o bien crear uno nuevo para tal efecto.



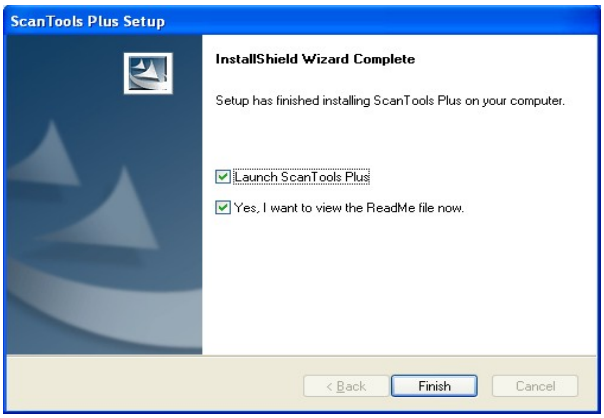
Una vez aceptado y configurado todo se procede a la instalación de la aplicación.



Acabada la instalación se muestra un texto sobre la licencia del producto, soporte y atención al cliente.



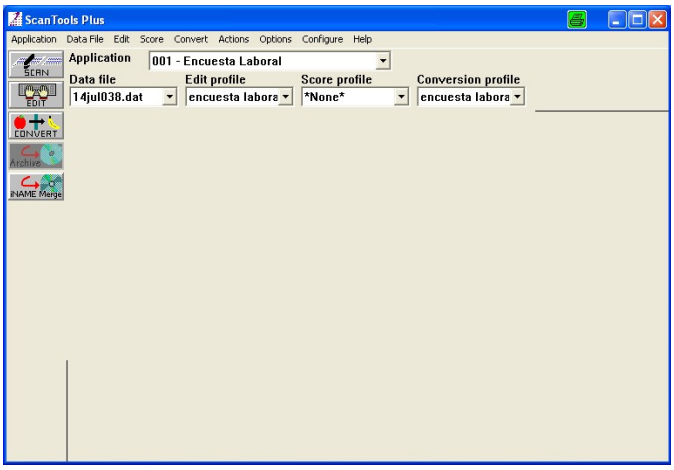
Ya para finalizar se nos presenta dos opciones como ultimo paso, lanzar la aplicación y ver el fichero de cambios.



3.9.2 Instalación de la aplicación de escaneado

La aplicación de OMR es un protocolo de escaneado que se ejecuta dentro de la aplicación del scanner Insight. Para instalar la aplicación se accede al menú configure > Install Application. Localizamos los ficheros *.$ep y *.sds.

Una vez finalizada la misma procedemos a seleccionar la aplicación, el "Data File", el "Edit Profile" y el "Conversion Profile".

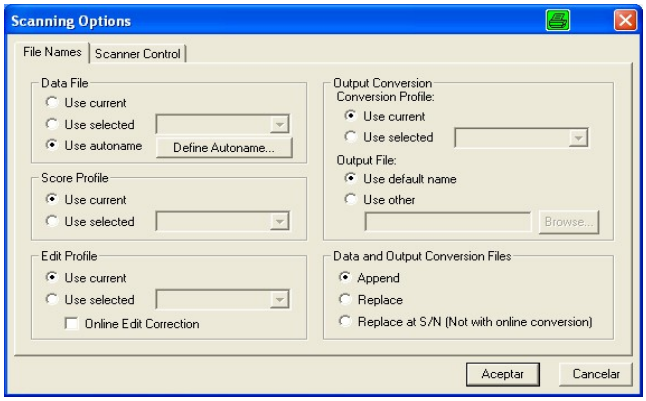


En el caso del "Data File", no aparecerá uno por defecto así que crearemos uno mediante el Menu Data File > New, se especificará un nombre (ej: fundación) y ya se procederá a efectuar una prueba de escaneo.

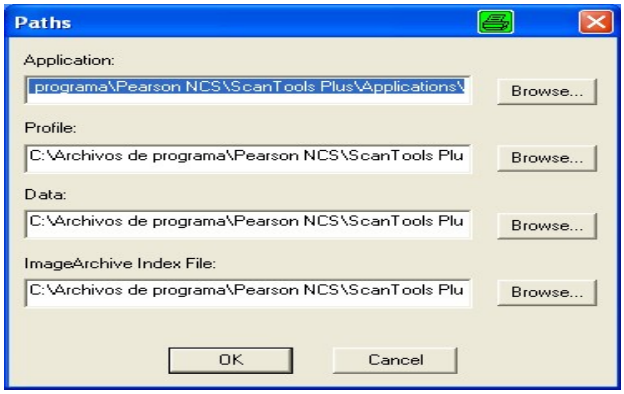
Estos ficheros contienen toda la información sobre el escaneado de las encuestas y su conversión a un fichero *.csv.

3.9.3 Peculiaridades de la instalación/configuración

- Existe la posibilidad de configurar la aplicación para que se cree un fichero de datos por cada escaneo de documentos que se realice configurando la aplicación mediante Options > Scanning Options en la pestaña "File Names", seccion "Data File" opción "Use Autoname".



- Se ha detectado que en algunos casos no copia el fichero Z001.ini en la carpeta de instalación de las aplicaciones, con lo que habría que copiarlo a mano. Podemos acceder a ella mediante la opción del menú Configure> Paths > Application



3.10 Instalación de LimeSurvey

Para instalar limesurvey, es necesario tener instalados:

- ✔ Apache Web Server 2.2
- ✔ Módulo PHP
- ✔ Base de datos PostgreSQL 8.3

Lea los apartados anteriores del manual si desea instalar estos componentes. A continuación:

```
cd <directorio_fuentes>
cp limesurvey/limesurvey-182/config.php limesurvey/limesurvey-182/config-local.php
```

Edite el fichero config-local.php conforme a lo descrito en el manual de LimeSurvey: http://docs.limesurvey.org/tiki-index.php?page=English+Instructions+for+LimeSurvey

Y ejecute los siguientes comandos cuando haya terminado la configuración de LimeSurvey:

```
createdb -E UNICODE -O sa -e -U sa -h localhost limesurvey
./install_limesurvey.sh limesurvey sa <password> /var/www  local
```

Ahora navegue con su explorador hasta http://<host>/limesurvey/admin/install/ y termine la configuración.

Para concluir borre los directorios del instalador:

```
cd /var/www/limesurvey
mv admin/install admin/.install
```