

Manual de Huellas de Luz

11 de marzo de 2015

Índice

Sistema de información para archivos de imágenes	3
Arquitectura del sistema	4
Configuración Común	4
SSH (Security Shell)	4
Samhain	4
Logwatch	4
Correo Electrónico	5
Servidor 1	5
Apache httpd con acceso público	5
Sitio Huellas de Luz	5
Sitio Fotógrafos y Editores Franceses en México	6
Subversion (svn)	7
Gitweb	8
Acceso anónimo a Git vía HTTP	8
Base de datos de publicaciones	9
Acceso autentificado de Git por ssh	10
Denyhosts	10
Snort	10
Servidor 2	10
Tomcat/Huellas de Luz	10
Actualización de contenido	11
Base de datos de publicaciones	13
Sitio de colaboración interna	13
Manual para instalación local del Pescador	14
Requerimientos	14
Sistema Operativo: Debian GNU/Linux 7 (Wheezy)	14
Java JDK 7 y JRE 1.7	14
Maven 3.0.5	15

GitHub	17
ImageMagick	17
Navegador web compatible con GWT (Firefox)	18
Instalación del Pescador en un ambiente local	19
Instalación de Eclipse y plugins	21
Instalación del plugin de Maven	22
Instalación del plugin de Google (GWT) para Eclipse	23
Configuración de Eclipse	24
Importación del proyecto a Eclipse	24
Configuración del ambiente	25
Pescador o Huellas de Luz	28
Estructura	28
Red semántica	31
Red Semántica de una Colección Archivística	31
Red Semántica de una Imagen Archivística	38
Sitio Huellas de Luz	46
Mapa del sitio	46

Sistema de información para archivos de imágenes

Para permitir el acceso a imágenes patrimoniales y la búsqueda en los datos relacionados, el sitio *Huellas de Luz* emplea un sistema informático llamado **Pescador**. Este sistema —desarrollado por el proyecto *Preservación de Imágenes, Sistemas de Información, Acceso e Investigación*— organiza los datos del catálogo con base en la Web Semántica, lo cual permitió la creación de diversas funciones de búsqueda y despliegue de la información.

Se han escrito [varios textos](#) acerca del sistema y los aspectos del proyecto relacionados con las ciencias de la computación. El Pescador es software libre, y su código fuente puede descargarse mediante la herramienta Git, bajo los términos de la licencia GPL, desde esta dirección: <http://lais.mora.edu.mx/gitrepo/pescador.git>

La difusión del código bajo esta modalidad responde al objetivo de fomentar las colaboraciones amplias en el desarrollo de sistemas informáticos para archivos y en las investigaciones relacionadas. Sin embargo, debe notarse que en su estado actual de desarrollo el sistema Pescador no es fácil de utilizar o adaptar a otro contexto archivístico; hacerlo requiere de conocimientos detallados de diversas tecnologías que el sistema utiliza. No obstante, por favor no duden en [escribirnos](#) con cualquier propuesta de colaboración o pregunta al respecto.¹

¹ Información de <http://lais.mora.edu.mx/huellasdeluz/#contenido;id=MXIMHDL-AcercaDelSitio-Pescador>

Arquitectura del sistema

El sistema Pescador cuenta con 2 servidores: *lais.mora.edu.mx* y *lais-interno.mora.edu.mx*. Éstos cuentan con algunas aplicaciones que son iguales en ambos para el acceso remoto, así como otras para seguridad. A continuación se detallarán las aplicaciones compartidas y posteriormente las características singulares de cada servidor.

Configuración Común

SSH (Security Shell)

El mecanismo principal para la configuración y administración de los servidores por vía remota es *ssh* (*Security Shell*).

El acceso remoto tiene varias restricciones de seguridad, ésta se encuentran el archivo */etc/ssh/sshd_config*. Sólo es posible el acceso a usuarios especificados con el parámetro **AllowUsers** y no se permite el acceso por ssh al usuario **root**. Se recomienda que no esté activada la opción de autenticación por contraseña (**PasswordAuthentication**), sin embargo por ahora está activada en el servidor *lais.mora.edu.mx*, a petición del Departamento de Informática, ya que ellos no han podido ingresar mediante la llave pública/privada.

Para el servidor *lais.mora.edu.mx*, el ssh se encuentra restringido por el programa *Denyhosts*, que será abordado posteriormente.

Samhain

Samhain es un programa de seguridad que monitorea continuamente los archivos básicos de sistema y hace notificaciones de cambios vía correo electrónico al usuario **root**. Después de realizar cualquier instalación o modificaciones en los paquetes, así como cualquier cambio de configuración, es necesario actualizar la base de datos de Samhain, para evitar que las modificaciones se vuelvan a notificar al reiniciarse el servidor. Para hacerlo, deben ejecutarse los siguientes comandos como **root**:

```
/etc/init.d/samhain stop  
/usr/sbin/samhain -t update --foreground -m none  
/etc/init.d/samhain start
```

Una vez por semana *Samhain* avisa que ha recargado su configuración. Esto se debe a la rotación de las bitácoras del sistema.

Logwatch

Se trata de un programa que hace un resumen diario de varias bitácoras del sistema y lo envía por correo electrónico al usuario **root**.

Correo Electrónico

Se utiliza *Postfix* como agente de transporte de correos. Hay dos mecanismos para el reenvío de correos para `root` a cuentas externa. Primero, el archivo `/etc/aliases` permite reenviar los correos para `root` a un usuario normal del mismo servidor. Luego, en la carpeta `~/.forward` del usuario normal, se especifican las direcciones de correo externas a las cuales deben reenviarse los mensajes.

Servidor 1

Nombre:	lais.mora.edu.mx
Dirección en la red interna:	10.1.1.232
Dirección pública:	201.148.84.213
Puertos para conexiones externas:	80 (http), 20 (ssh)

Nota: No es posible reiniciar este servidor por vía ssh. Sólo el personal del Departamento de Informática puede hacerlo mediante los controles del sistema de servidores virtuales.

Apache httpd con acceso público

En este servidor se ejecuta un *Apache httpd*, que sirve de conducto para la difusión de todos los sitios Web públicos del **LAIS** (Laboratorio Audiovisual de Investigación Social).

Algunos sitios se generan y se configuran directamente desde este Apache, mientras otros en realidad se generan de otros procesos, en este servidor o en el otro. Entonces para varios sitios, este Apache funciona como proxy para canalizar datos entre el usuario y los procesos que hacen el trabajo verdadero.

El archivo principal de configuración es `/etc/apache2/sites-available/default`, el cual queda ligado (*linked*) como `/etc/apache2/sites-available/000-default`.

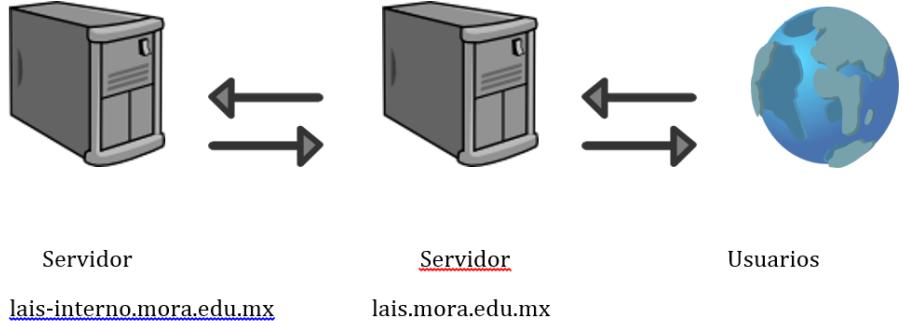
Es importante tener en cuenta que el orden de las directrices (*rules*) en la configuración de *Rewrite*² es primordial.

Sitio Huellas de Luz

El sitio **Huellas de Luz** se encuentra en el servidor *lais-interno.mora.edu.mx*, mientras que *lais.mora.edu.mx* es un proxy para la difusión pública del sitio.

De manera esquemática puede representarse de la siguiente manera:

²Información detallada del módulo Rewrite: <http://httpd.apache.org/docs/2.0/misc/rewriteguide.html>



```
Líneas de configuración de Apache relevantes:  
  
writeRule ^/huellasdeluz([^\/]*) /huell  
writeCond %{REQUEST_URI} !huellasdeluz  
writeRule ^(.*) /huellasdeluz/ [R]  
oxyPass /huellasdeluz http://lais-interr  
:1122/webclient-0.3-SNAPSHOT  
oxyPassReverse /huellasdeluz http://lais  
mx:1122/webclient-0.3-SNAPSHOT
```

Para más detalles sobre la configuración de Huellas de Luz, véase más adelante, en la sección sobre el servidor *lais-interno.mora.edu.mx*.

La dirección principal de acceso a este sitio es <http://lais.mora.edu.mx/huellasdeluz>. Note que este Apache *httpd* redirecciona a dicha URL cualquier solicitud que no reconozca.

Sitio Fotógrafos y Editores Franceses en México

Dentro de este proyecto existió este sitio, sin embargo ahora se encuentra fuera de funcionamiento, pese a ello se incluye en este manual la información proporcionada por el desarrollador para cualquier referencia que sea necesaria. Dicho sitio se genera mediante un servidor Ruby que corre en esta máquina y que no es visible públicamente. Este *Apache httpd* es un proxy para la difusión pública del sitio.

Líneas de configuración Apache relevantes:

```
RewriteRule ^/ff/web/catalogo(.*) http://localhost:3000/web/  
    ui$1 [P]  
RewriteCond %{REQUEST_URI} !images_ui  
RewriteRule ^/ff/(.*) http://localhost:3000/$1 [P]
```

El servidor *Ruby* se levanta automáticamente mediante esta instrucción de arranque:

/etc/init.d/pescador

El comando anterior se ejecuta automáticamente al arrancar o apagar el servidor. Se encuentra linkeado en las carpetas /etc/rc[0-6].d/

Note que el sitio tarda varios minutos en estar disponible después del arranque, debido a la necesidad de cargar todos los datos en la memoria RAM. El proceso mismo del servidor es un proceso *ruby* cuyo PID se registra en */var/run/pescador.pid*

El sitio opera con una versión anterior de Pescador (0.2), la cual se encuentra instalado en la carpeta

```
/var/local/pescador-fotografos-franceses/
```

Se trata de una copia instalado desde el depósito de código *svn*, con la versión en la rama *produccion_ff*. Las dependencias Java de esta versión del sistema se encuentran instaladas en */usr/local/lib* y linkeadas en

```
/var/local/pescador-fotografos-franceses/rails/vendor/java/
```

La dirección principal de acceso a este sitio es <http://lais.mora.edu.mx/ff/> (Sitio fuera de funcionamiento)

Nota: es imprescindible la diagonal al final de la dirección

Posibles problemas:

Esta versión del sistema Pescador contiene algunos errores fatales que tiran el proceso ruby. En el caso de que este sitio se encuentre fuera de línea, probablemente se puede levantar nuevamente con el comando (ejecutado como **root**):

```
/etc/init.d/pescador start
```

Por otra parte, el sistema depende de bibliotecas *Rails* que se instalaron desde el sistema de paquetes *Debian*, cuya actualización puede a veces causar problemas.

Subversion (svn)

Este depósito de código fuente contiene el código de la versión anterior de Pescador (0.2). Este *Subversion* se instaló desde el sistema paquetes *Debian*. Se comunica con los usuarios mediante *HTTP* a través del *Apache httpd*. Usa esta vía tanto para desplegar las páginas Web que muestran el código fuente, como para interactuar con los programadores mediante la herramienta de línea de comandos *svn* (así como otras interfaces para descargar e ingresar código fuente).

Línea de configuración Apache relevante:

```
RewriteRule ^/svn(.*) /svn$1 [L]
```

Hay configuraciones más detalladas en */etc/apache2/mods-enabled/dav_svn.conf* y */etc/apache2/mods-enabled/dav_svn.load*

El depósito en se encuentra instalado en

```
/var/local svn/
```

La dirección principal de acceso a las páginas que despliegan el código fuente desde Subversion es: <http://lais.mora.edu.mx/svn/>

Gitweb

A partir de la versión 0.3 del sistema Pescador, se usa un depósito de código *Git* (en lugar de *Subversion*, que se empleaba con la versión anterior). Gitweb es una interfaz web de Git escrita en Perl para el despliegue de páginas Web con los contenidos del depósito de código fuente *Git* (cuya configuración general se describe más adelante). Gitweb forma parte de Git y se ha instalado desde el sistema de paquetes Debian.

Líneas de configuración Apache relevantes:

```
# aliases suggested in Gitweb on Debian tutorial
Alias /gitweb.css /usr/share/gitweb/gitweb.css
Alias /git-favicon.png /usr/share/gitweb/git-favicon.png
Alias /git-logo.png /usr/share/gitweb/git-logo.png
Alias /git /var/local/git
ScriptAlias /gitweb.cgi /usr/lib/cgi-bin/gitweb.cgi

# these settings also suggested for Gitweb
SetEnv GITWEB_CONFIG /etc/gitweb.conf
HostnameLookups On
UseCanonicalName Off
ServerSignature Off

<directory /var/local/git>
    AllowOverride None
    Options FollowSymlinks
    Order Allow,Deny
    Allow From All

    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-f
    #RewriteCond %{REQUEST_FILENAME}!-d
    RewriteRule ^.* /gitweb.cgi/$0 [L,PT]
</directory>

RewriteCond %{REQUEST_URI} !git
```

Nota: esta última directriz aparece en dos lugares en el archivo de configuración.

Nota: no es relevante el siguiente archivo de configuración:

```
/etc/apache2/conf.d/gitweb
```

La dirección principal de acceso a las páginas que despliegan el código fuente desde Git es: <http://lais.mora.edu.mx/git/>

Acceso anónimo a Git vía HTTP

Existe un segundo mecanismo de acceso a Git vía HTTP, el acceso anónimo mediante la herramienta de línea de comandos `git` (u otra herramienta

equivalente). Este mecanismo permite descargar de manera anónima todos los contenidos del depósito de código fuente, pero no permite ingresar código fuente ni modificar los contenidos del depósito.

Al igual que Gitweb, este mecanismo de acceso forma parte de Git y se ha instalado desde el sistema de paquetes Debian.

Líneas de configuración Apache relevantes:

```
# for anonymous git access over http
SetEnv GIT_PROJECT_ROOT /var/local/git
SetEnv GIT_HTTP_EXPORT_ALL
ScriptAlias /gitrepo/ /usr/lib/git-core/git-http-backend/

# require authentication for http git access
<LocationMatch "^/gitrepo/.*/git-receive-pack$">
    AuthType Basic
    AuthName "Git Access"
    Require group committers
</LocationMatch>

RewriteCond %{REQUEST_URI} !gitrepo
```

Nota: esta última directriz aparece en dos lugares en el archivo de configuración.

Nota: a propósito, no existen usuarios con autorización para modificar el depósito por esta vía.

Comando para descargar de manera anónima los contenidos del depósito Git:

```
git clone http://lais.mora.edu.mx/gitrepo/pescador.git
```

Base de datos de publicaciones

En realidad este sitio se genera desde *lais-interno.mora.edu.mx*. Entonces *lais.mora.edu.mx* es un proxy para la difusión pública del sitio.

Líneas de configuración Apache relevantes:

```
RewriteRule ^/publicaciones/?$ /publicaciones/index.php/
    topics/single/4 [R]
RewriteCond %{REQUEST_URI} !publicaciones

ProxyPass /publicaciones http://lais-interno.mora.edu.mx/
    aigaion2
ProxyPassReverse /publicaciones http://lais-interno.mora.edu
    .mx/aigaion2
```

Para más detalles sobre la configuración de la base de datos de publicaciones, véase más adelante, en la sección sobre el servidor *lais-interno.mora.edu.mx*.

La dirección principal de acceso a este sitio es: <http://lais.mora.edu.mx/publicaciones>

Acceso autenticado de Git por ssh

Para los desarrolladores que contribuyen al código de Pescador, el acceso al depósito de código *Git* en el servidor se hace mediante *ssh*.

Para cada desarrollador con autorización para modificar el depósito de código, debe crearse un usuario en el servidor. Luego, el desarrollador debe crear un par de llaves pública/privada, y debeadirse su llave pública al archivo `~/.ssh/authorized_keys` (en el directorio `home` del servidor). Asimismo, en el archivo `/etc/passwd`, debe modificarse el *shell* asignado a este usuario, para que sea *git-shell*. Esto impide que el usuario pueda entrar al servidor para ejecutar comandos distintos a los permitidos por *git*. (Por cierto, tampoco es necesario que dicho usuario tenga una contraseña para su cuenta en el servidor.)

Denyhosts

Se trata de un sistema que evita el hackeo del servidor mediante ataques de fuerza bruta *ssh*. Después de que un determinado número de intentos de acceso fallidos mediante ssh desde una dirección IP determinada, se bloquean los accesos *ssh* desde esa dirección.

Nota: si no se puede acceder desde una máquina determinada, se debe revisar si su dirección ha quedado dentro de la lista de direcciones bloqueadas. Esto puede pasar incluso para direcciones de la red interna del Instituto Mora. Para más información, consulte la documentación de *Denyhosts*.

Snort

Se trata de un sistema de seguridad que monitorea las comunicaciones de red del servidor. Envía un reporte diario al usuario `root`. Arroja un número importante de intentos de hackeo, así como bastantes resultados positivos erróneos, incluyendo algunas actividades normales de monitoreo de red que realiza el mismo Instituto Mora.

Servidor 2

Nombre:	lais-interno.mora.edu.mx
Dirección en la red interna:	10.1.1.42
Dirección pública:	201.148.84.214
Puertos para conexiones externas:	80 (http), 20 (ssh), 1122 (Tomcat http) 1143 (Depurador GWT)

Tomcat/Huellas de Luz

El sitio **Huellas de Luz** se genera mediante el servlet Java de Pescador 0.3, que corre dentro de Tomcat y se encuentra disponible en el puerto 1122. Si bien

es posible acceder a este servicio directamente desde fuera del Instituto Mora, la ruta de acceso normal es a través de *lais.mora.edu.mx*, que sirve de proxy.

El sitio se encuentra instalado en el directorio
`/var/lib/tomcat7/webapps/webclient-0.3-SNAPSHOT/`

Este directorio es generado automáticamente por Tomcat a partir del archivo
`webclient-0.3-SNAPSHOT.war`

Sin embargo, para el funcionamiento correcto del sitio, es necesario copiar manualmente varias cosas a este directorio, después de que Tomcat lo genera.

El sitio se levanta automáticamente al iniciarse el servidor. Note que tarda varios minutos en activarse el sitio, ya que el sistema carga todos los datos a la memoria RAM al iniciarse.

El archivo de configuraciones del sitio es:

`/var/lib/tomcat7/pescador-setup.properties`

Tal como se especifica en este archivo, los datos del sitio propiamente dicho están en

`/home/lais-usuario-dev-2/git/pescador/data/hd1/`

Este directorio se encuentra dentro de una copia de trabajo del depósito de código Git, por lo que las modificaciones a los datos también son controladas por dicho depósito.

Las imágenes históricas que difunde el sitio están en

`/var/lib/tomcat7/webapps/webclient-0.3-SNAPSHOT/imagenes/`

Otros archivos disponibles para su descarga desde el sitio (específicamente, artículos en PDF) están en

`/var/lib/tomcat7/webapps/webclient-0.3-SNAPSHOT`

Si no se modifican los contenidos del sitio, no hay necesidad de cambiar nada en esta carpeta.

Actualización de contenido

A continuación se detallan los procedimientos para actualizar el sitio.

En el caso de la actualización de las páginas de texto estático del sitio (por ejemplo, en “Acerca del sitio”) es suficiente modificar los archivos con la extensión `.txt` en el directorio `/home/lais-usuario-dev-2/git/pescador/data/hd1/`. En este caso no es necesario reiniciar el demonio Tomcat para que se actualicen los contenidos en el sitio. No olvide mantener los contenidos del depósito Git en *lais.mora.edu.mx* sincronizados con los de esta copia de trabajo.

En el caso de actualizarse los artículos en PDF que pueden descargarse desde el sitio, es suficiente copiarlos a la carpeta `/var/lib/tomcat7/webapps/webclient-0.3-SNAPSHOT`. Tampoco es necesario reiniciar el Tomcat. No olvide dejar una copia actualizada en

`/home/lais-usuario-dev-2/git/pescador/data/hdl/` también y sincronizar los contenidos del depósito Git en `lais.mora.edu.mx` con esta copia de trabajo.

En el caso de modificar el código fuente del sistema, el procedimiento de actualización es más complejo.

Primero debe recompilarse el sistema y generar un nuevo archivo `.war`. Desde la carpeta

```
/home/lais-usuario-dev-2/git/pescador/java/
```

Y como usuario normal, ejecute el siguiente comando:

```
mvn3 clean install
```

(El comando `mvn3` es un enlace hacia la versión 3 de Maven, instalada en la carpeta `/usr/local/bin/mvn3`. El sistema Pescador requiere esta versión de Maven, y el sistema de paquetes de Debian sólo ofrece una versión desactualizada.)

Después, como `root` (o como `lais-usuario-dev-1`, que es lo mismo), ejecute los siguientes comandos:

```
rm -rf /var/lib/tomcat7/webapps/webclient-0.3-SNAPSHOT*
cp /home/lais-usuario-dev-2/.m2/repository/mx/org/pescador
/webclient/0.3-SNAPSHOT/webclient-0.3-SNAPSHOT.war /var/
lib/tomcat7/webapps/
/etc/init.d/tomcat7 restart
```

Esto hará que Tomcat reinicie y vuelve a desplegar el servlet, sin embargo no se levantará correctamente el sitio, debido a que todavía no se han copiado los contenidos requeridos a la carpeta del servlet. Entonces, ejecute los siguientes comandos, nuevamente como `root` o `lais-usuario-dev-1`:

```
cp -r /home/lais-usuario-dev-2/imagenes/ /var/lib/tomcat7/
webapps/webclient-0.3-SNAPSHOT
cp ./data/hdl/*pdf /var/lib/tomcat7/webapps/webclient-0.3-
SNAPSHOT
/etc/init.d/tomcat7 restart
```

Así volverá a iniciar el servlet de Pescador con todos los contenidos disponibles, y se levantará correctamente el sitio. Puede monitorearse el proceso de inicio de Pescador mediante el comando:

```
tail -f -n 100 /var/log/tomcat7/catalina.out
```

Se puede acceder a la administración del demonio Tomcat, para controlar los servlets (incluyendo el de Pescador), desde la siguiente dirección:

<http://lais-interno.mora.edu.mx:1122/manager/html>

Possible problema: De vez en cuando, por motivos que desconozco, el servlet no inicia después de una actualizarse (con el procedimiento que se acaba de describir). La solución que he encontrado es detener y reiniciar Tomcat, o en última instancia, volver a desplegar el servlet y sus contenidos.

Para información sobre la configuración del acceso público al sitio Huellas de Luz mediante la dirección <http://lais.mora.edu.mx>, véase la sección anterior, sobre el Servidor 1.

Base de datos de publicaciones

La base de datos de publicaciones usa el sistema de gestión bibliográfico *Aigaion*. Este sistema, basado en php y MySQL, se genera desde el Apache httpd en este servidor. Si bien es posible acceder a este servicio directamente desde fuera del Instituto Mora, la ruta de acceso normal es a través de lais.mora.edu.mx, que sirve de proxy.

El sistema se encuentra instalado en el directorio

`/var/www/aigaion2/`

Aigaion no se instaló desde el sistema de paquetes Debian, ya que no está disponible por esta vía. Se descargó directamente del sitio de los autores del programa. Note que es necesario descargar el paquete para el idioma español por separado.

Algunos aspectos de la configuración pueden controlarse mediante la interfaz Web, al acceder como usuario administrador. Otros aspectos de la configuración deben modificarse en el archivo

`/var/www/aigaion2/aigaionengine/config/config.php`

La carpeta de adjuntos para el sistema es:

`/home/lais-usuario-dev-2/aigaion/adjuntos/`

Las configuraciones base actual del Apache httpd en este servidor permiten que funcione correctamente este sistema. Es necesario tomar en cuenta las necesidades de éste antes de realizar cualquier modificación a dichas configuraciones.

Para información sobre la configuración del acceso público a la base de datos de publicaciones mediante la dirección <http://lais.mora.edu.mx>, véase la sección anterior, sobre el Servidor 1.

Sitio de colaboración interna

Se encuentra instalado en el servidor el sistema *EGroupware*. Este sistema fue instalado desde el sistema de paquetes Debian. La configuración para Apache (establecido por el paquete Debian) se encuentra en

`/etc/apache2/conf.d/egroupware`

La configuración principal del sitio se realiza mediante la interfaz Web, al acceder como usuario administrador.

La dirección para acceder al sitio es:

<http://lais-interno.mora.edu.mc/egroupware>

Manual para instalación local del Pescador

Requerimientos

Sistema Operativo: Debian GNU/Linux 7 (Wheezy)

Para la descarga de Debian, recurrir a la página: www.debian.org

Para instalar Debian junto con Windows, se recomienda seguir este tutorial:

<http://dodebian.blogspot.mx/2011/11/install-debian-alongside-windows-guide.html>

http://dodebian.blogspot.mx/2011/11/install-debian-alongside-windows-guide_19.html

http://dodebian.blogspot.mx/2011/11/install-debian-alongside-windows-guide_6728.html

En el equipo actual (Mayo 2014), Dell Precision T1700, necesita un controlador específico para la tarjeta de red en Debian Wheezy 7.0. Abajo se especifica qué hacer para hacerla funcionar.³

Es necesario utilizar el controlador *e1000e* para conexiones de red de Intel®Gigabit. El controlador soporta Conexiones por PCI Express.

1. Descargar el controlador *e1000e-<x.x.x>.tar.gz*, *jx.x.xj* es la versión:

https://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=15817

2. Mover el archivo *e1000e.tar.gz* al directorio */usr/local/src/e1000e*

3. Extraer el controlador del archivo tar con el comando:

```
tar zxf e1000e-<x.x.x>.tar.gz
```

4. Dirigirse al directorio src:

```
cd e1000e-<x.x.x>/src/
```

5. Compilar el módulo del controlador:

```
# make install
```

6. Cargar el módulo de la siguiente manera:

```
# modprobe e1000e
# modprobe e1000e insmod e1000e
```

Java JDK 7 y JRE 1.7

Instalación de JDK 7.

En la línea de comandos ejecutar las siguientes instrucciones con privilegios de root:⁴

³La fuente original es: <http://www.intel.com/support/network/sb/cs-032514.htm>

⁴Para más información: <http://stackoverflow.com/questions/15543603/installing-java-7-oracle-in-debian-via-apt-get>
<http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-debian.html>

```

$ su root
# echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu
      precise main" | tee /etc/apt/sources.list.d/webupd8team-
      java.list
# echo "deb-src http://ppa.launchpad.net/webupd8team/java/
      ubuntu precise main" | tee -a /etc/apt/sources.list.d/
      webupd8team-java.list
# apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
      EEA14886
# apt-get update
# apt-get install oracle-java7-installer
# exit

```

En necesario definir variables de entorno para Java⁵. Ejecutar las siguientes instrucciones con privilegios de root (adminsitrador).

```
# vi /etc/profile
```

Introducir en ese archivo las siguientes dos líneas, que definen la variable de entorno JAVA_HOME, y agregarla al PATH con los comandos:

```
export JAVA_HOME=/usr/java/jdk1.5.0_07/bin/java
export PATH=$PATH:/usr/java/jdk1.5.0_07/bin
```

Guardar y cerrar el archivo, después se ejecuta alguna de las dos líneas que abajo aparecen para activar las propiedades de PATH.

```
# source /etc/profile
```

o

```
# . /etc/profile
```

Maven 3.0.5

No utilizar versiones superiores. A partir de la versión 3.1 el gem-maven-plugin no está soportado aún.⁶

Para instalar seguir las indicaciones:

1. Bajar Maven 3.0.5 (archivo *bin.tar.gz*):
<http://maven.apache.org/download.cgi>
2. Mover el archivo *apache-maven-3.0.5-bin.tar.gz* a la carpeta */usr/local/apache-maven*.
Ejecutar el siguiente comando para extraer la distribución de Maven, se creará la carpeta *apache-maven-3.0.5*

⁵Para más información: [http://www.cyberciti.biz/faq/linux-unix-set-java_home-path-variable/](http://www.cyberciti.biz/faq/linux-unix-set-java-home-path-variable/)

⁶Para más información recurrir a <http://stackoverflow.com/questions/8826881/maven-install-on-mac-os-x/19564500#19564500>
<https://cwiki.apache.org/confluence/display/MAVEN/AetherClassNotFound>

```
tar zxf apache-maven-3.0.5-bin.tar.gz
```

3. Añadir la variable de entorno M2_HOME

```
$ export M2_HOME=/usr/local/apache-maven/apache-maven-3.0.5
```

4. Añadir la variable de entorno M2

```
$ export M2=$M2_HOME/bin
```

5. Añadir la variable de entorno MAVEN_OPTS para especificar propiedades de la JVM

```
$ export MAVEN_OPTS="-Xms256m -Xmx512m"
```

Esta variable de entorno puede ser utilizada para proporcionar opciones adicionales.

6. Añadir la variable de entorno M2 a la variable PATH

```
$ export PATH=$M2:$PATH
```

7. Agregar las siguientes dos líneas al archivo /etc/enviroment con privilegios de administrador

```
M2_HOME=/usr/local/apache-maven/apache-maven-3.0.5  
PATH=$PATH:$JAVA_HOME/bin:$M2_HOME/bin
```

8. Ejecutar lo siguiente en la línea de comandos.

```
$ source /etc/enviroment
```

9. Cerrar e iniciar sesión.

10. Ejecutar el siguiente comando en consola para verificar que se ha instalado Maven de forma correcta

```
$ mvn --version
```

Otra manera de definir las variables de entorno para Java y Maven es creando o modificando el archivo `~/.bash_profile`, al que deben agregársele las siguientes líneas:

```
JAVA_HOME=/usr/lib/jvm/java-7-oracle  
export JAVA_HOME
```

```
M2_HOME=/usr/local/apache-maven/apache-maven-3.2.1  
export M2_HOME
```

```

M2=$M2_HOME/bin.
export M2

PATH=$PATH:$JAVA_HOME/bin:$M2
export PATH

```

Una vez que se ha guardado el archivo, se corre la siguiente línea de comando:

```
# source ~/.bash_profile
```

Por último, es necesario cerrar la sesión y reiniciarla.

GitHub

Para instalar GitHub⁷ primero se deben instalar todas las dependencias, esto se hace a través de la consola de comandos:

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext
      libz-dev libssl-dev
```

Para hacer la descarga de Git, debe bajarse de la siguiente página:

<http://git-scm.com/download>

Una vez descargado el archivo .tar.gz, debe descomprimirse e instalarse con los siguientes comandos: ⁸

```
$ tar -zxf git-1.7.2.2.tar.gz
$ cd git-1.7.2.2
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

Finalmente se actualiza Git a sí mismo con el siguiente comando git:

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

Para la configuración, se recomienda seguir los pasos mencionados en la siguiente liga, allí se detalla cómo crear un usuario con los privilegios necesarios:

<http://git-scm.com/book/en/Getting-Started-First-Time-Git-Setup>

ImageMagick

El sistema *Pescador* contiene imágenes que despliega en el portal, dichas imágenes tienen distintas dimensiones y es necesario que sean manipuladas. Debido a que todo se hace de forma automática, o sin intervención de alguna persona, se requiere de un software como *ImageMagick* que sea capaz de manipular imágenes vía línea de comandos.

Para la instalación se deben seguir estas indicaciones:

⁷Para una referencia completa de Git, se recomienda el libro en línea Pro Git: <http://git-scm.com/book/en/v2>

⁸<http://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

1. Descargar el archivo de ImageMagick de la siguiente liga (Esta es la última versión disponible en el momento en que fue escrito este documento).

<http://www.imagemagick.org/download/ImageMagick-6.8.9-0.tar.gz>

2. Descomprimir el archivo y compilarlo con los comandos: ⁹

```
$ tar xvzf ImageMagick.tar.gz  
$ cd ImageMagick-6.8.9  
$ ./configure  
$ make
```

3. Con privilegios de administrador, ejecutar las siguientes instrucciones que harán la instalación, configuración y verificación.

```
$ sudo make install  
$ sudo ldconfig /usr/local/lib  
$ /usr/local/bin/convert logo: logo.gif
```

Navegador web compatible con GWT (Firefox)

Se requiere un navegador web compatible con el plugin de Google GWT llamado “Google Web Toolkit Developer Plugin” pero debido a que dicho plugin es poco compatible con las versiones recientes de Firefox y Chrome (navegadores comunes en Linux) por lo que se recomienda instalar una versión compatible (antigua) de Firefox.

Debido a que los comandos `apt-get` y `yum` no permiten instalar múltiples versiones del mismo navegador, se describen los pasos para instalar una versión antigua de Firefox sin que altere la versión reciente instalada en Linux: ¹⁰

1. Ir a la siguiente página y decidir que versión de Firefox se desea:

<ftp://ftp.mozilla.org/pub.mozilla.org/firefox/releases/>

Se utilizará esta versión como modelo a seguir:

<ftp://ftp.mozilla.org/pub.mozilla.org/firefox/releases/24.8.0esr/>

2. Estando en la carpeta contenedora de la descarga (por ejemplo `/home/usuario`), descomprimir el archivo:

```
tar -zxvf firefox-9.0.1.tar.bz2
```

`-z` significa filtrar el archivo a través de gzip

`-x` extraer

`-v` verbose

`-f` nombre de archivo

⁹Para mayor información se sugiere consultar la fuente original: <http://www.imagemagick.org/script/install-source.php#unix>

¹⁰La explicación original se encuentra en la siguiente dirección y es válida para distribuciones Linux basada en Debian: <http://ubuntuforums.org/showthread.php?t=1907045>

3. Colocar firefox en su propio directorio:

```
mv firefox firefox-24.8.0
```

4. Hacer un respaldo de tu actual directorio de perfil:

```
cp -r /home/usuario/.mozilla /home/usuario/.mozilla3
```

Si usas Seamonkey o Thunderbird, está información también estará ahí.

5. Crear un nuevo perfil para no corromper el anterior perfil:

```
firefox -CreateProfile "firefox-24.8.0" /home/usuario/.mozilla/firefox/firefox-24.8.0"
```

Este es mi nombre de perfil **firefox-24.8.0** y este es mi directorio de perfil
/home/usuario/.mozilla/firefox/firefox-24.8.0

6. Ejecutar firefox 24.8.0 desde el directorio donde se colocó:

```
/home/usuario/firefox-24.8.0/firefox -no  
-remote -P firefox-24.8.0
```

/home/usuario/firefox-24.8.0/firefox es el directorio de instalación
-no-remote permite utilizar firefox 24.8.0 y versiones antiguas de firefox
-P es la opción de perfil
firefox-24.8.0 es el nombre del perfil

De esta manera ya es posible descargar el plugin de GWT para Firefox en la siguiente dirección:

<https://code.google.com/p/google-web-toolkit/downloads/list>

Se selecciona una versión adecuada (con extensión xpi), se ejecuta el navegador como se menciona, se selecciona del menú principal de Firefox la opción *Complementos* y después *Extensiones*. Se arrastra el archivo del plugin GWT a esta ventana, se aceptan las condiciones y se instalará automáticamente.

Instalación del Pescador en un ambiente local

Una vez cubierto los requerimientos de la sección anterior, es posible hacer la descarga del código del Pescador, que se encuentra en un repositorio Git alojado en el servidor 1 ya mencionado. Si es la primera vez que se descargará el Pescador, la siguiente línea puede ser omitida, de lo contrario es necesario limpiar el repositorio de Maven con:

```
$ rm -rf ~/.m2/
```

El comando para la descarga anónima de los contenidos del repositorio Git es el siguiente:

```
$ git clone http://lais.mora.edu.mx/gitrepo/pescador.git
```

Como resultado tendremos una carpeta llamada **pescador** desde el directorio donde el comando fue ejecutado. Ahora dirigirse a la carpeta **java**:

```
$ cd pescador/java
```

Desde ese directorio ejecutar el siguiente comando:

```
$ mvn clean install
```

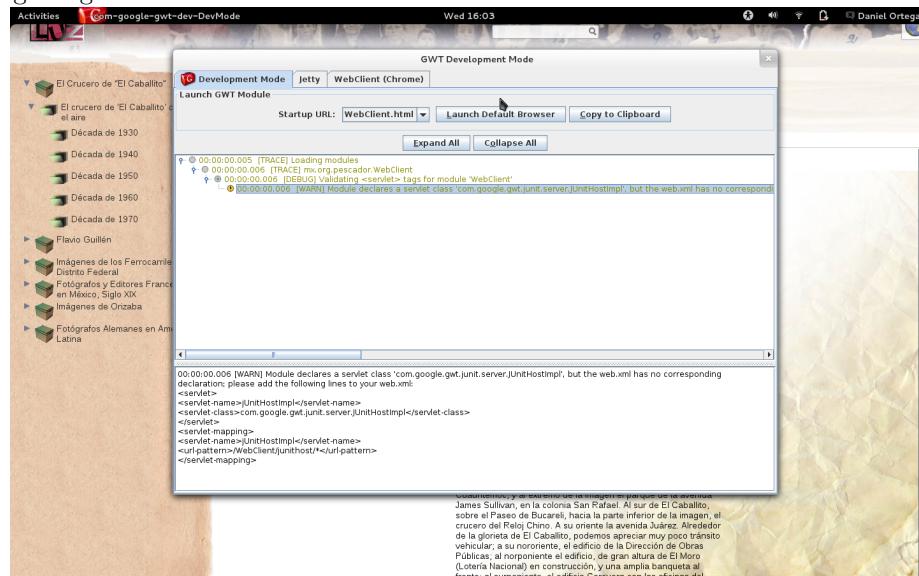
Es importante verificar que la compilación no tenga errores y los tres proyectos Maven que están allí contenidos, hayan pasado el proceso con éxito. Algo así debe aparecer:

```
[INFO] -----  
[INFO] Reactor Summary:  
[INFO]  
[INFO] Pescador Maven Parent ..... SUCCESS [0.527s]  
[INFO] Pescador Server ..... SUCCESS [9.714s]  
[INFO] Pescador WebClient ..... SUCCESS [1:28.349s]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----
```

Una vez hecho lo anterior, es posible construir el Pescador desde línea de comandos usando Maven, para ello es necesario moverse al directorio **pescador/java/webclient** y ejecutar el siguiente comando:

```
$ mvn gwt:run
```

El resultado será que se abre la consola de GWT, como se muestra en la imagen siguiente:



Desde allí se puede iniciar la aplicación web en el botón “Launch Default Browser”. Hasta este punto, ya se tiene funcionando el Pescador, sólo que no contiene las imágenes que están en línea.

Fototecas Digitales Acerca del sitio Publicaciones y vínculos Búsqueda

Grupo MXIM-1-3-4 Década de 1940

2 de 5 en El crucero de 'El Caballito' desde el aire

Ficha Contiene: 30 unidades

Código de referencia: MXIM-1-3-4-1
Fecha: ca. 1940
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 1794

Código de referencia: MXIM-1-3-4-2
Fecha: 1943
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 2546

Código de referencia: MXIM-1-3-4-3
Fecha: 1946
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 4116

Código de referencia: MXIM-1-3-4-4
Fecha: 06 de noviembre de 1949
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 5968

Código de referencia: MXIM-1-3-4-5
Fecha: 07 de junio de 1949
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 6097

Código de referencia: MXIM-1-3-4-6
Fecha: 1941
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, caja "Ciudad de México No. 4", vuelo 150, negativo 376

Código de referencia: MXIM-1-3-4-7
Fecha: 1947
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, caja "Ciudad de México No.5", vuelo 241.1, negativo 347

Código de referencia: MXIM-1-3-4-8
Fecha: 1947
Productor: Compañía Mexicana de Aerofoto (México)
Referencia de procedencia: México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, vuelo 318.2

Fotografía MXIM-1-3-4-1

1 de 30 en Década de 1940

Ficha

Área de identificación

Código de referencia	MXIM-1-3-4-1
Referencia de procedencia	México, Ingenieros Civiles Asociados, Fundación Ingenieros Civiles Asociados, negativo 1794
Fecha	ca. 1940
Autores	Productor
Compañía Mexicana de Aerofoto (México)	
Grupo al que pertenece en esta fototeca	

Estructura y contenido

Estructura Formal	Paisaje urbano
Ángulo de la cámara	Aéreo oblicuo
Fotografía tomada desde	suroriental
Fotografía tomada hacia	noroeste
Descripción	Hacia la parte superior se ubica la Escuela Nacional de Maestros, la parte inferior de la imagen abarca hasta el crucero del Reloj Chino, en el extremo derecho la estación de trenes de Buenavista y en el izquierdo la calle 8 de Octubre. Se aprecia cerca del ángulo inferior izquierdo la glorieta de El Caballito, en la esquina poniente del Paseo de la Reforma. Al norte de El Caballito, la glorieta y panteón de San Fernando, hacia el ángulo superior derecho, al noroeste el Panteón de Dolores, en la parte inferior, se aprecia, seguido en la parte superior de la imagen, al noroeste, Mascarones, Santa María la Rivera y, la Escuela Nacional de Maestros, también se distingue una parte de Azcapotzalco. Al poniente de la glorieta el Moro, en la parte superior, se aprecia la parte posterior del edificio del Frontón México. En dirección suroriental, sobre el Paseo de la Reforma, vemos la glorieta de Colón y la de Cuauhtémoc, y en la parte de la glorieta de Colón la estatua de Andrew James Sully, y en la colonia San Pedro, Al sur de El Caballito, sobre el Paseo de Bucareli, hacia la parte inferior de la imagen, el crucero del Reloj Chino. A su oriente la avenida Juárez. Alrededor de la glorieta se observan edificios de vivienda, un importante vial, a su noroeste, el edificio de la Dirección de Obras Públicas; al noroeste el edificio, de gran altura de El Moro, instrucción, y una amplia avenida que conecta al

127.0.0.1:8888/WebClient.html?gwt.codesvr=127.0.0.1:9997#contenido=id=hd1rep:MXIM-1-3-4-1/de_listado_id=VPayC8

Instalación de Eclipse y plugins

Para poder hacer una correcta depuración del Pescador, es necesaria de una herramienta o IDE. Para este cometido se emplea Eclipse con diferentes plugins. Primero se debe bajar Standard Eclipse más reciente, que hasta ahora es la 4.3.2,

para Linux 64 bit. Aquí se encontrará la última versión.

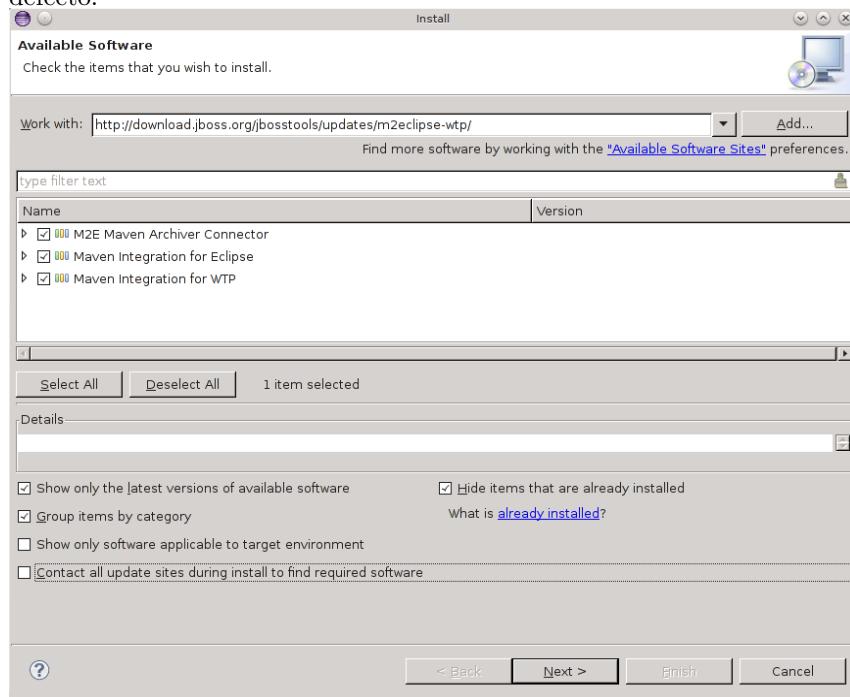
<https://www.eclipse.org/downloads/>

Una vez que ya se cuenta con el archivo `eclipse-estándar-kepler-SR2-linux-gtk-x86_64.tar.gz`, debe extraerse en la carpeta donde se desee alojar Eclipse, no es necesaria una instalación, sólo la extracción.

Instalación del plugin de Maven

Una vez iniciado Eclipse, seguir estos pasos:

1. Seleccionar **Help > Install New Software...** y se abrirá una ventana.
2. En el campo “Work with” se debe ingresar la siguiente liga y presionar enter:
`http://download.jboss.org/jbosstools/updates/m2eclipse-wtp/`
3. Se abrirá la siguiente ventana. Y se deben seleccionar los componentes allí marcados. Las opciones inferiores deben dejarse como aparecen por defecto.

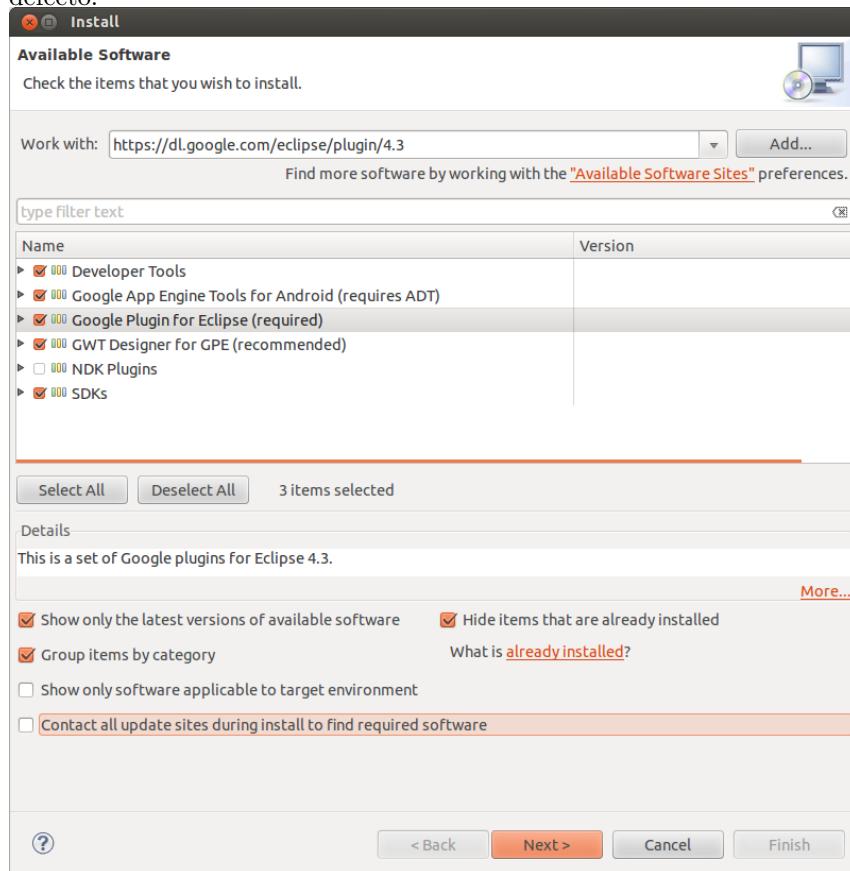


4. Presionar **Next** y aceptar las condiciones. Durante el proceso de instalación pedirá autorización para instalar software que no está firmado y se debe aceptar y permitir. Por último, se debe reiniciar Eclipse.

Instalación del plugin de Google (GWT) para Eclipse

Una vez iniciado Eclipse, seguir estos pasos¹¹:

1. Seleccionar Help > Install New Software... y se abrirá una ventana.
2. En el campo “Work with” se debe ingresar la siguiente liga y presionar enter:
`https://dl.google.com/eclipse/plugin/4.3`
3. Se abrirá la siguiente ventana. Y se deben seleccionar los componentes allí marcados. Las opciones inferiores deben dejarse como aparecen por defecto.



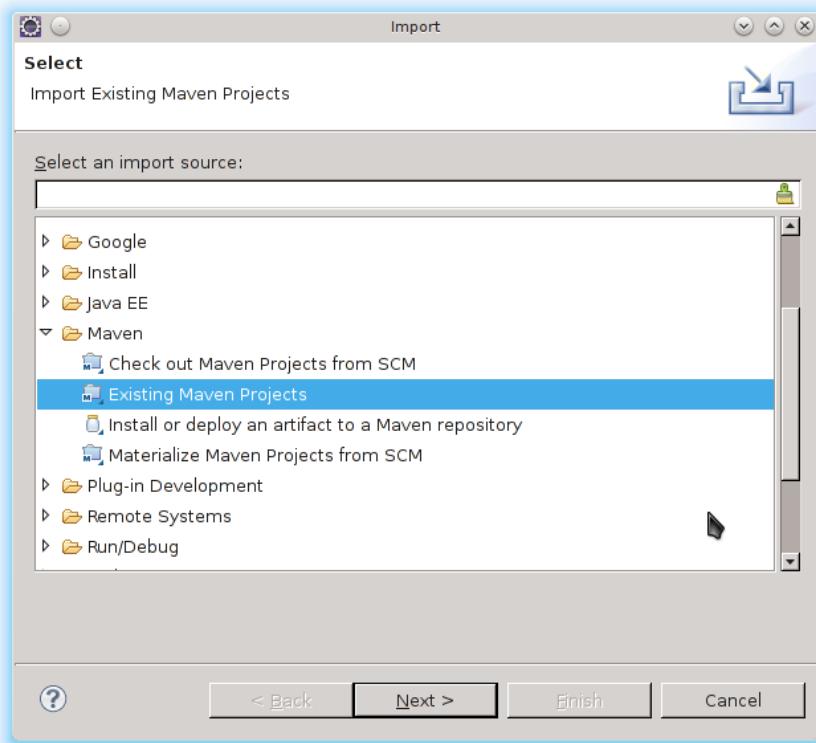
4. Presionar Next y aceptar las condiciones. Durante el proceso de instalación pedirá autorización para instalar software que no está firmado y se debe aceptar y permitir. Por último, se debe reiniciar Eclipse.

¹¹Más información en: <https://developers.google.com/eclipse/docs/install-eclipse-4.3>

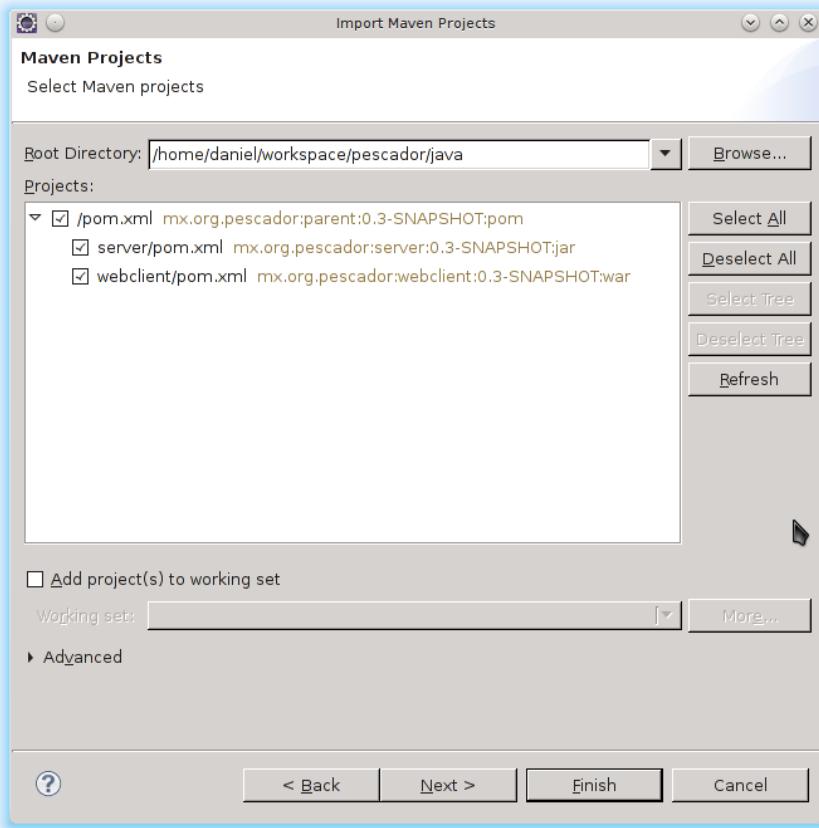
Configuración de Eclipse

Importación del proyecto a Eclipse

Para cargar el proyecto a Eclipse, éste se debe importar. En la ventana de diálogo debe escogerse un Proyecto existente de Maven (File > Import > Existing Maven Projects) y continuar.



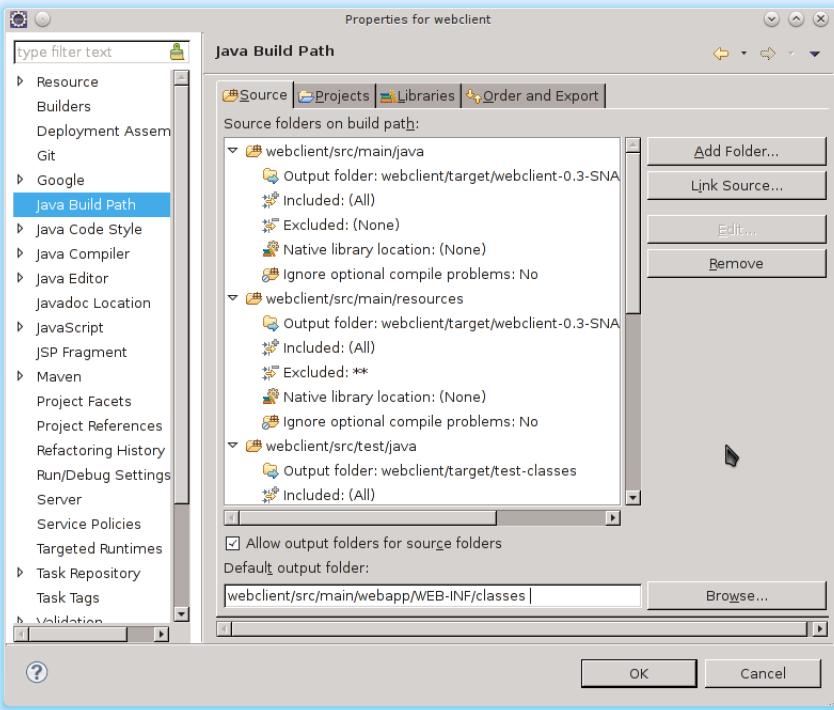
El siguiente paso es escoger el directorio principal (Root Directory) que contiene el archivo `pom.xml` del módulo “Parent”, este directorio es `/pecador/java` que está en el proyecto descargado del repositorio git.



Como se aprecia en la imagen superior, hay un proyecto principal y dos módulos: **sever** y **webclient**. Se presiona continuar para verificar que todas las dependencias estén resueltas, si no es así, Eclipse solicitará permiso para la instalación de lo requerido. Por último se presionará finalizar.

Configuración del ambiente

Una vez que se tienen los módulos **Parent**, **Webclient** y **Server** como proyectos separados en Eclipse, se deben seguir estos pasos para hacer unas modificaciones al ambiente. Los cambios se harán a las propiedades del proyecto **Webclient** y aparecerá la pantalla que se muestra a continuación:



Los pasos son:

1. Ir a **Google > Web Toolkit**. La opción “*Use Google Web Toolkit*” debe estar seleccionada y se debe seleccionar un SDK de GWT válido.
2. Ir a **Google > Web Application**. Seleccionar como carpeta WAR a **src/main/webapp**, y desactivar la opción “*Launch and deploy from this directory*”.
3. Ir a **Java Build Path**.
 - Definir **webclient/src/main/webapp/WEB-INF/classes** como “*Default output folder*”
 - Desactivar la opción “*Allow output folders for source folders*”
 - En la pestaña **Source**: Quitar las exclusiones de las carpetas fuente **webclient/src/main/resources** y **webclient/src/test/resources**
 - Agregar como carpeta fuente **webclient/target/generated-sources/apt** y **webclient/target/generated-sources/gwt**
4. Ir a **Java Compiler > Annotation Processing**. Activar la opción “*Enable project specific setting*”

5. Ir a Java Compiler > Annotation Processing > Factory Path. Activar la opción “*Enable project specific setting*”. Agregar el archivo .jar de `gwt-mpv-apt`, que se descarga automáticamente por Maven. Éste se encuentra en la carpeta `~/.m2/repository/org/gwttmpv/gwt-mpv-apt/2.2.0` o donde esté el repositorio local de las descargas de Maven.

Después de los pasos anteriores, desaparecerán los errores que aparecen en la compilación, esto se debe a que como se agregan carpetas al ambiente, es posible detectar las clases Java faltantes.

Ahora se puede ejecutar el proyecto **Webclient** como una aplicación Web.

Pescador o Huellas de Luz

Como se explicó al principio de este documento, el proyecto en su conjunto se llama **Huellas de Luz**, esto incluye la investigación social, el proceso de catalogación y el desarrollo informático. Las alusiones hechas a **Huellas de Luz** en este documento hacen referencia directa al **Pescador**, que es propiamente el desarrollo informático.

A lo largo de este documento se han abordado la estructura de los servidores, los requerimientos y la instalación del sistema. Ahora se especificará información con respecto al proyecto, su estructura, la web semántica y otros detalles.

Estructura

El sistema **Pescador** está compuesto por 3 proyectos Maven en Java: *Parent*, *Server* y *WebClient*. Maven cumple la función de resolver todas las dependencias, además de que permite la separación de un proyecto general en módulos, como en este caso.

El proyecto *Parent* sólo contiene a los otros dos, los cuales realmente tienen la funcionalidad del sistema.

El proyecto *Server* tiene la función de generar la web semántica con todos los datos de las imágenes, así como de hacer las búsquedas.

El proyecto *WebClient* tiene el código que se ejecuta en los exploradores, a través del plugin GWT de Google.

La carpeta llamada *Pescador*, que debe ser descargada del servidor a través del repositorio *Git*, contiene todos los archivos del proyecto. Las principales carpetas son:

Java Contiene todo el código de los 3 proyectos Maven, se encuentra aquí el `pom.xml` y también 2 carpetas que dividen a los proyectos *Server* y *WebClient*.

data En esta carpeta hay una carpeta llamada `ff_images`, dentro de la cual están todas las imágenes archivadas en el sistema. Dentro de **data** también está la carpeta `hd1`, que contiene más de 150 archivos en formato `csv`, que contienen toda la información de las imágenes y en un formato útil para la creación de la red semántica.

def_files Esta carpeta contiene los archivos `.def` que dentro de la web semántica definen el modelo de datos (RDF) y las ontologías del sistema, es decir, que definen el vocabulario para describir las propiedades y clases, así como la relación entre ellas, la cardinalidad y caracterizaciones.

scripts Debido a que no existe un mecanismo de ingreso de nueva información al sistema, se ha utilizado una serie de archivos en *Excel* para el vaciado de información; sin embargo, este formato no es el idóneo para poder generar la web semántica, por eso se utilizan los scripts contenidos en esta carpeta, con el objeto de convertir los archivos *Excel* en `csv`.

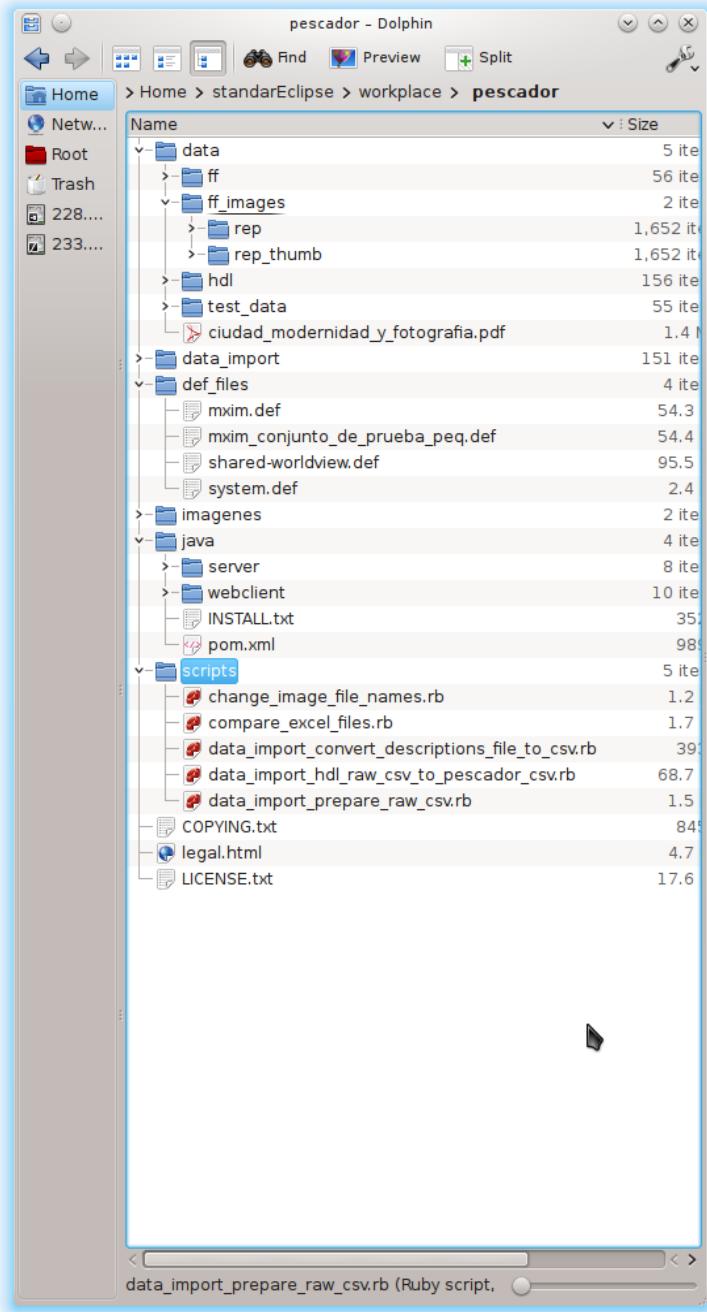


Figura 1: Carpeta con la estructura del sistema Pescador

Para entender la relación entre la gran cantidad de archivos y la relación entre ellos, se inició con la ingeniería inversa del proyecto. Se hicieron, a través de unas herramientas en Eclipse, diagramas de paquetes. Se generaron con la división hecha con Maven.

El diagrama de la figura 2 y de la figura 3 muestran la estructura de los proyectos *Server* y *Webclient*, respectivamente. No es posible ver con claridad las imágenes de estos diagramas, pero estarán disponibles junto a este manual en mayor calidad y resolución.

Para comenzar el análisis profundo de este proyecto, se recomienda iniciar con la clase `ServerController.java` que inicia con la lectura de los archivos `csv` para la creación de la web semántica. Ese archivo se encuentra en el proyecto `server` y en el paquete `mx.org.pescador.server`.

Se recomienda el estudio de las clases:

- `java.util.HashMap<K,V>`
- `java.util.Map<K,V>`

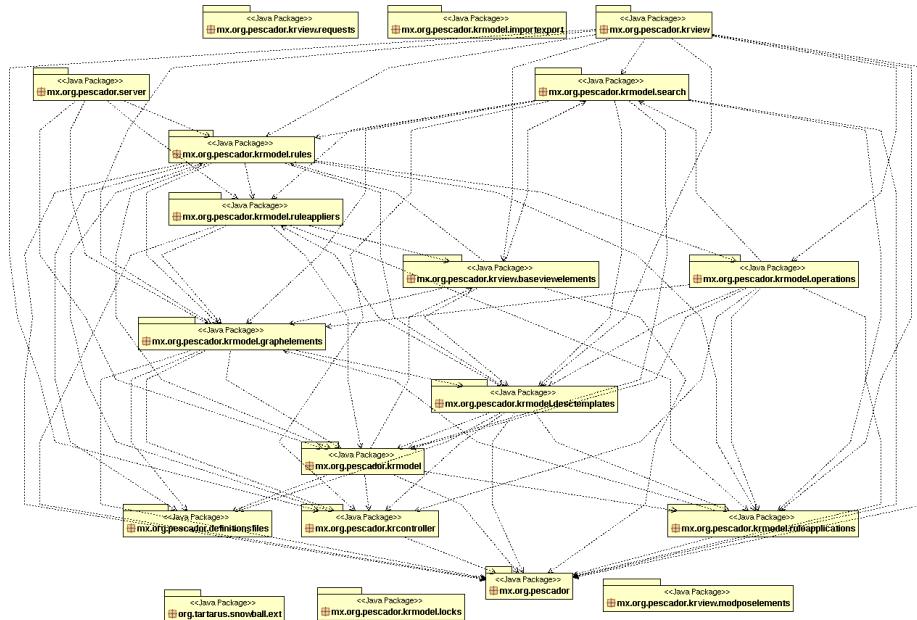


Figura 2: Diagrama de paquetes del proyecto *Server*

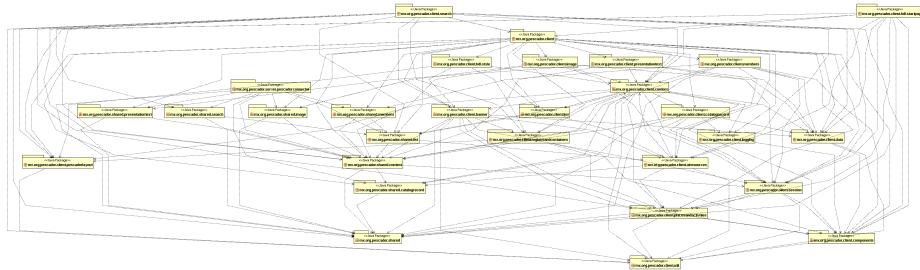


Figura 3: Diagrama de paquetes del proyecto *Webclient*

Red semántica

Una red semántica o esquema de representación en red es una forma de representación de conocimiento lingüístico en la que los conceptos y sus interrelaciones se representan mediante una gráfica. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles. Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales y mentales.

En una gráfica o red semántica los elementos semánticos se representan por nodos. Dos elementos semánticos entre los que se da una relación semántica estarán unidos mediante una línea, flecha, enlace o arista. Ciertos tipos de relaciones no simétricas requieren gráficas dirigidas que usan flechas en lugar de líneas.¹²

Para este proyecto se diseñó una red semántica para la representación de colecciones e imágenes.

Red Semántica de una Colección Archivística

A partir de los archivos csv contenidos en la carpeta /pescador/dat/hd1 se creó la representación gráfica de toda colección archivística. En esta sección se encuentra una explicación de los principales elementos que conforman el diagrama. Se abordará de manera seccionada por razones de simplicidad, ya que no es posible copiar en su totalidad el diagrama y que sea legible.

Se llamará *nodo* a los óvalos azules y *arista* a las líneas que conectan a los óvalos. Una línea puede representar varias aristas, si a ella están asociadas más de una leyenda, cada leyenda es un arista.

En esta red semántica el nodo principal es **Colección Archivística**, que está ligado con el arista **id** al nodo **id_value**, éste contiene el identificador único de la colección. El nodo **null** no contiene información, por lo que se descartarán todos aquellos nodos que así aparezcan, se mantienen en el diagrama porque está definido dentro de los archivos csv.

Todos los nodos que sean **string** tendrán texto, que es el valor de la característica que indique la arista que lo apunta.

¹²Red semántica. http://es.wikipedia.org/wiki/Red_sem%C3%A1ntica

La tercera flecha que apunta al nodo **string** representa a 13 aristas que apuntan a un dato de la Colección. Las 13 etiquetas dan información sobre el dato contenido, por ejemplo, las características físicas, el volumen o las condiciones de acceso.

Se hace la aclaración que volumen se refiere al número de imágenes contenidas en la colección.

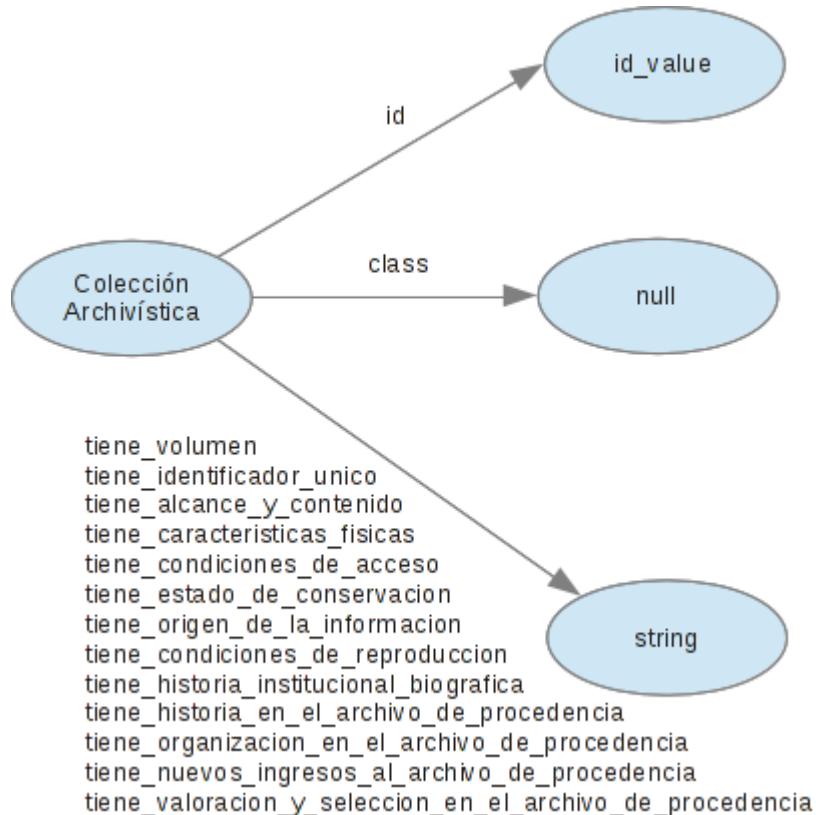


Figura 4: Colección Archivística

Toda colección cuenta con un título, el cual tiene un origen y este se detalla a través del tipo de título, esto se representa en el nodo **svw.title_types**. Los títulos pueden haber sido generados al momento de la catalogación o ya contar con un título. (Ver figura 5)

Los autores de una colección pueden ser instituciones (**svw.institutions**) o personas (**svw.people**). Como se ve en el diagrama, éstos están detallados con otros nodos que son unidos por aristas. En el caso de los editores y productores, pueden ser tanto personas como instituciones, acepta que sean ambos si la situación lo requiere. Cuando se menciona a uno o varios fotógrafos, éstos necesariamente son personas.

Las instituciones requieren de información variada como el nombre, el acrónimo y la ubicación, el nodo `swv.places` guarda toda la información posible de una dirección.

Una institución puede pertenecer a instituciones superiores, para ello se usa la arista `has_parent_institution` y `has_direct_parent_institution`.

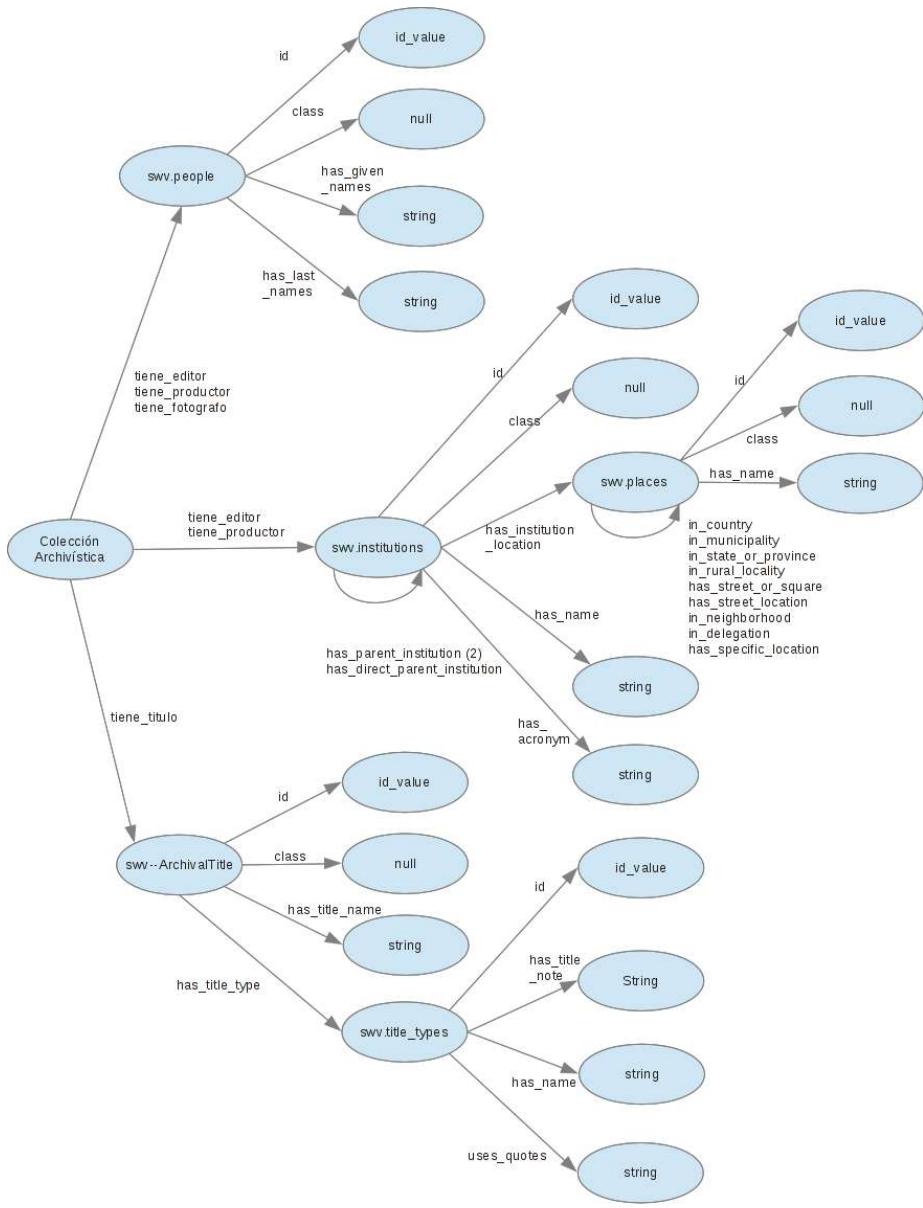


Figura 5: Colección Archivística

Toda Colección archivística tiene una fecha o un periodo que abarcan las imágenes que contiene, dicha información está ligada al nodo `swv-TimeRangeWSE`. Regularmente sólo se almacenan los años que comprende el periodo.

La colección tiene siempre una procedencia, en estos casos son ins-

tituciones o colecciones privadas, y todo está registrado en el nodo `mxim.referencias_de_procedencia`, nuevamente se encuentra el nodo de la institución que fue abordado anteriormente.

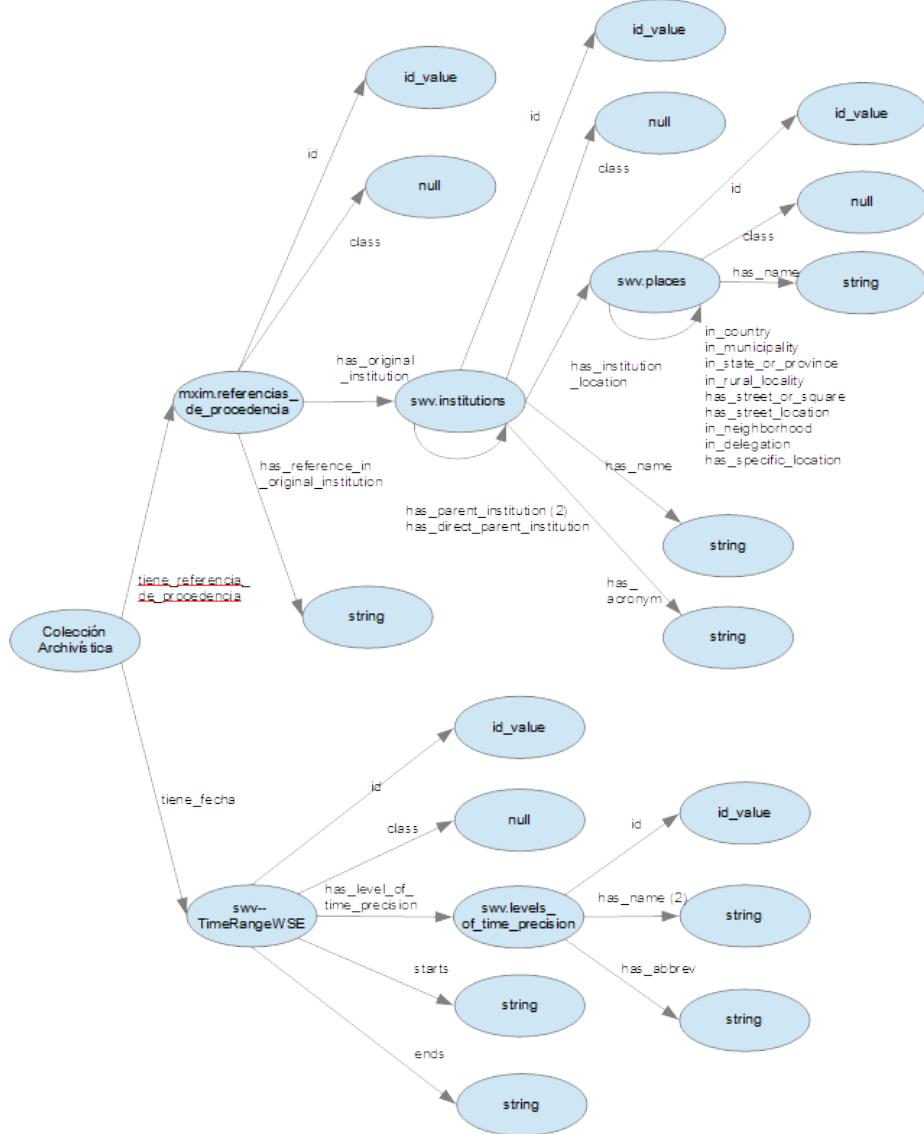


Figura 6: Colección Archivística

Dentro de una Colección Archivística existen componentes, éstos son principalmente publicaciones hechas por instituciones, y las imágenes aparecen en

alguna revista o en exposiciones. De igual forma, la información para la catalogación de una colección puede provenir de publicaciones y es importante tener los datos de la publicación y el lugar donde la información aparece.

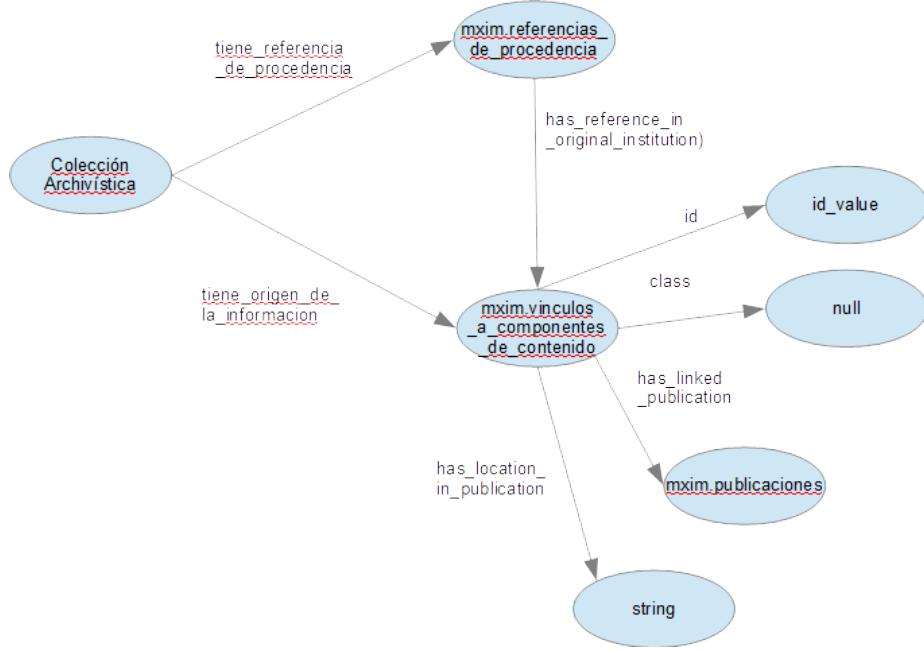


Figura 7: Colección Archivística

En el nodo **mxim.publicaciones** de la figura 7 salen varias aristas que especifican toda la información relacionada a dicho nodo. En la figura 8 se encuentran los detalles como el nombre, la periodicidad, el nombre del artículo donde aparecen las imágenes de la colección, la fecha y los autores.

Se suprimieron las aristas de todos los nodos que ya han sido expuestos, sin embargo, junto a este documento está una imagen en alta resolución que permite ver a detalle el diagrama en su conjunto y no seccionado como ha sido presentado en este documento.

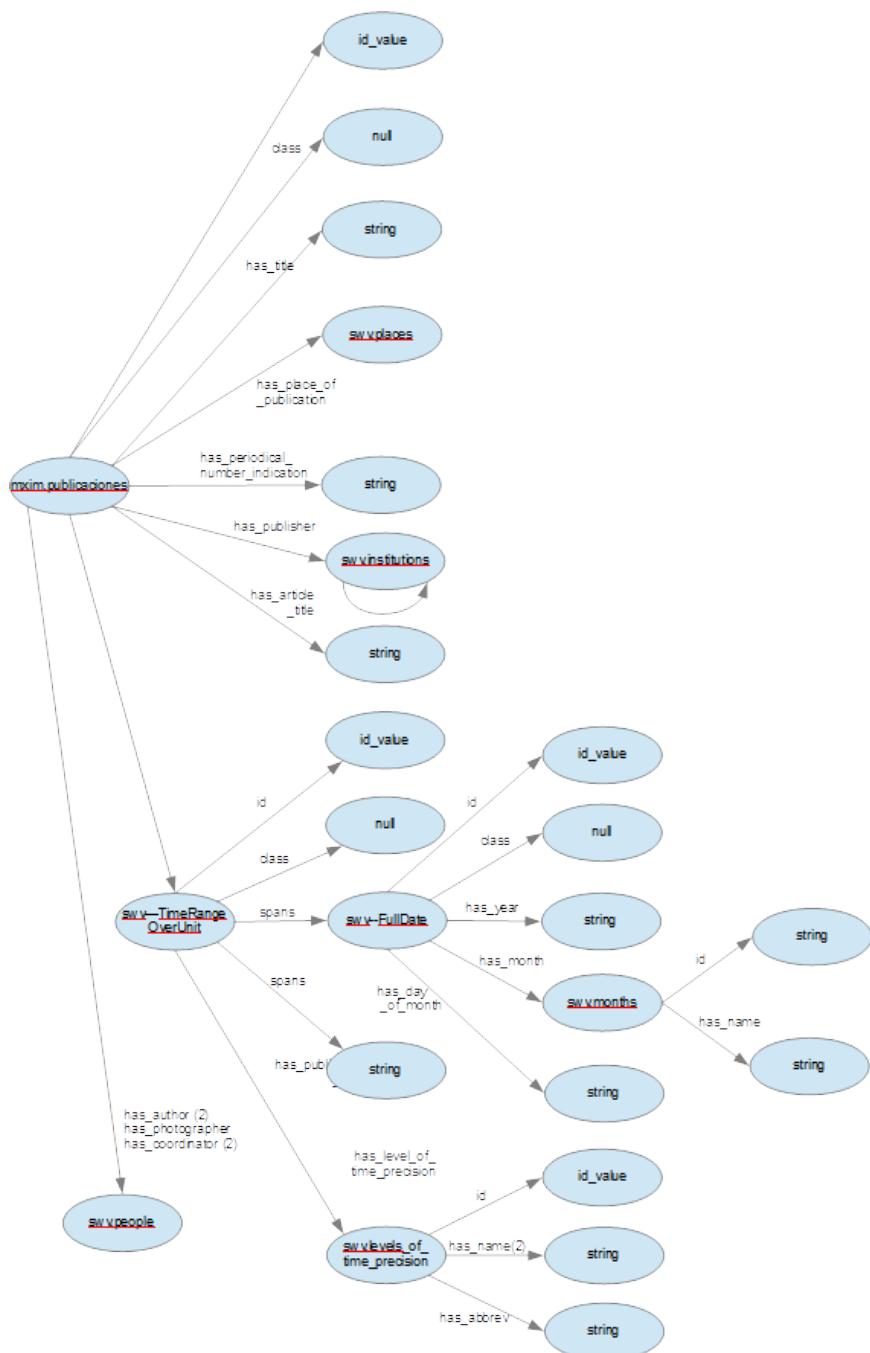


Figura 8: Colección Archivística

Red Semántica de una Imagen Archivística

Así como se creó el diagrama para las Colecciones Archivísticas, se hizo el correspondiente a la **Imagen Archivística**, el cual es aún más detallado y complejo. Para no repetir información, ya no se abordarán los nodos que fueron explicados en la sección anterior. Para la creación de este diagrama se usaron los mismos archivos **csv** contenidos en la carpeta **/pescador/dat/hdl** que ya fueron mencionados.

La Imagen Archivística tiene un identificador único y una clase (Esta siempre es con valor **null**, no se ha determinado la razón, se cree que es prescindible, probablemente en el diseño inicial tuvo alguna función).

El nodo **svw|FullDate** sirve para almacenar la fecha exacta, si es que se cuenta con ella, en que se tomó la fotografía o en que se publicó alguna caricatura. En el nodo **svw|months** existen 2 nodo más ligado a él, esto se debe a que se puede almacenar el nombre del mes en español o en inglés.

Para las fotografías se almacena la orientación con la que fue tomada, la altura, escala, entre otros.

Una imagen puede estar ligada a otras, y esto se representa con la flecha curva que sale del nodo **Imagen Archivística**, de esta forma se pueden ligar 2 imágenes si es que proceden del mismo negativo o una es reproducción de la otra o si tienen un encuadre similar.

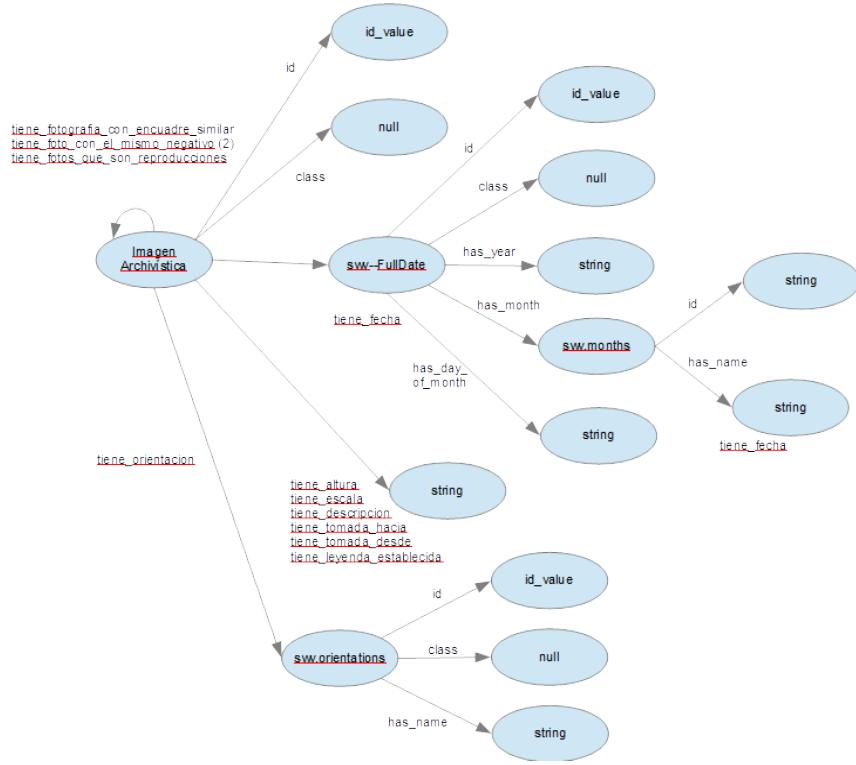


Figura 9: Imágen Archivística

En la figura 10 se muestra la estructura de los siguientes datos: la edición hecha, el ángulo en el que la cámara estuvo al tomar la fotografía, la estructura formal, el nivel de cámara y el lugar donde fue tomada la fotografía.

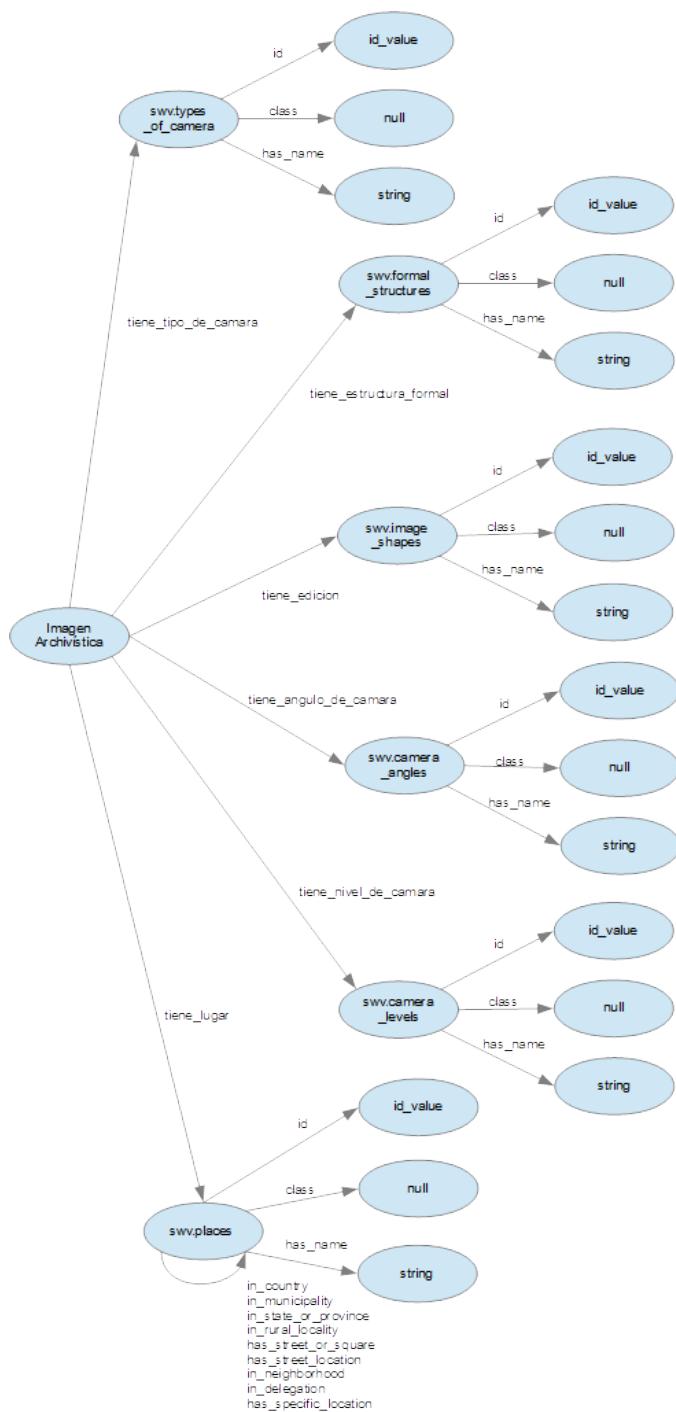


Figura 10: Imágen Archivística

Al igual que la Colección Archivística, la Imagen Archivística tiene autores que pueden ser tanto instituciones (`swv.institutions`) como personas (`swv.people`), este es el caso del editor, impresor, productor y fotógrafo fuente de grabado. Los que sólo pueden ser personas son el director, grabador, litógrafo, dibujante, responsable y quienes hacen el levantamiento.

Para registrar a los fotógrafos se hace un vínculo anterior, no se ha encontrado una razón para ello, podría estar ligado directamente al nodo `swv.people`.

Por otro lado, la imagen puede tener papel fotográfico y para registrarlo se usa el nodo `swv.roles_in_photographic_process`.

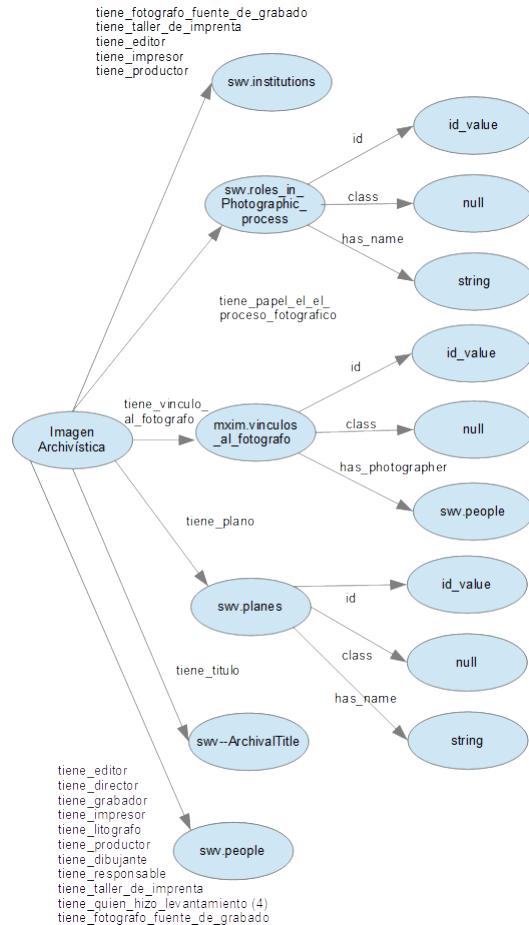


Figura 11: Imágen Archivística

La Imagen Archivística tiene más características que una Colección Archivística, pero se agrupan como una referencia de procedencia o como un componente de contenido. Los nodos `mxim.referencias_de_procedencia` y

`mxim.vinculos_a_componentes_de_contenido` tienen las mismas características ya mencionadas en la Colección Archivística.

El nodo `mxim.referencias_de_procedencia` sirve para representar fotos con encuadre similar, consecutivas, de la misma secuencia, del mismo negativo, que son reproducciones o que son base. Mientras que `mxim.vinculos_a_componentes_de_contenido` representa grabados relacionados, las notas de publicación y el origen de la infomación.

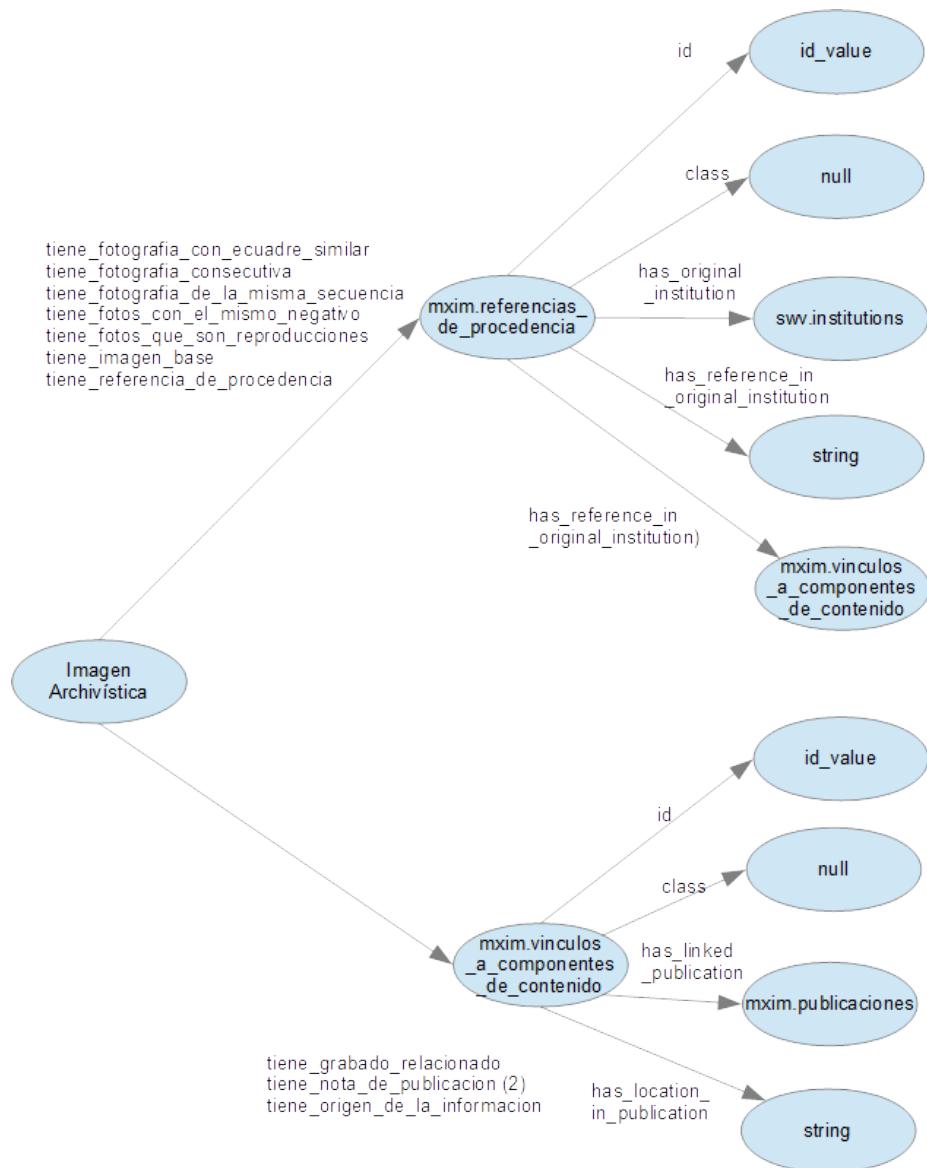


Figura 12: Imágen Archivística

El nodo `mxim.soportes_primerios` representa información sobre las características físicas de las imágenes como las dimensiones, los materiales, las leyendas con su ubicación.

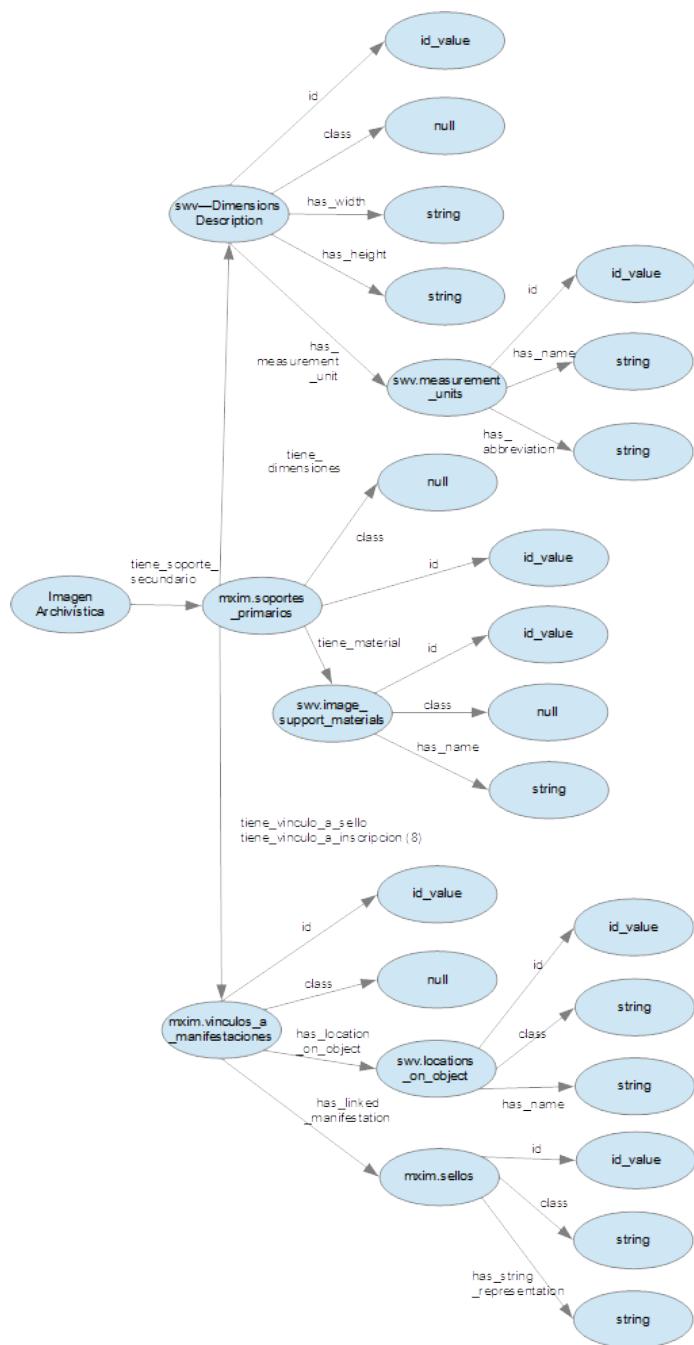


Figura 13: Imágen Archivística

El nodo `mxim.sopores_secundarios` contiene la información sobre los so-

portes de las imágenes que no componen a éstas. La estructura es idéntica al nodo `mxim.soportes_primerios`.

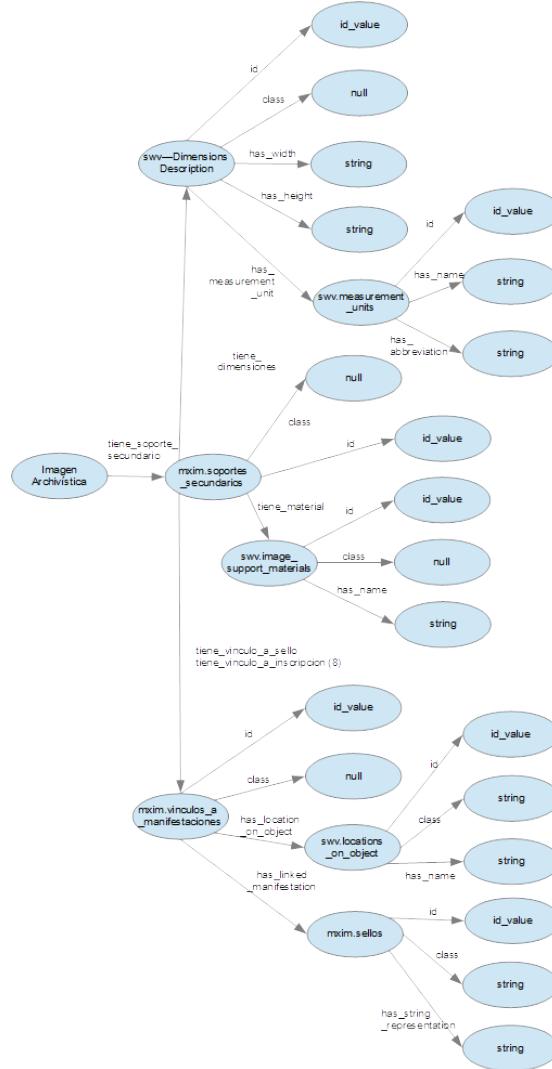


Figura 14: Imágen Archivística

Con este documento estará el diagrama completo y con una definición mejor que permita ver con más detalle la información y las relaciones entre las partes aquí expuestas.

Sitio Huellas de Luz

El sitio tiene una estructura de navegación bastante sencilla, sólo cuenta con 3 secciones: *Fototecas Digitales*, *Acerca del Sitio* y *Publicaciones y Vínculos*.

Mapa del sitio

Página de inicio: <http://lais.mora.edu.mx/huellasdeluz/>

- FOTOTECAS DIGITALES
 - El crucero de “El Caballito”
 - El crucero de ‘El Caballito’ desde el aire
 - ◊ Década de 1930
 - ◊ Década de 1940
 - ◊ Década de 1950
 - ◊ Década de 1960
 - ◊ Década de 1970
 - Flavio Guillén
 - Espacios Públicos
 - Retrato
 - ◊ Exteriores
 - ◊ Estudio
 - Quiromancia
 - Imágenes de los Ferrocarriles del Distrito Federal
 - Ferrocarriles del Distrito Federal 1857–1874
 - ◊ Caricatura
 - La Orquesta
 - El Ahuizote
 - ◊ Imágenes técnicas
 - Cartografía
 - ◊ Imágenes realistas
 - Fotografía
 - Grabado
 - Dibujo y pintura
 - Fotógrafos y Editores Franceses en México, Siglo XIX
 - IIE de la UNAM
 - Mapoteca Orozco y Berra
 - ◊ Panorámica General de México
 - ◊ Ciudades y ruinas americanas
 - Archivo General de la Nación

- Biblioteca Nacional de Francia
 - ◊ Album Fotográfico Mexicano
 - ◊ ”panorama général de mexico”
 - ◊ ”panorama général de puebla”
- I. Acevedo
- Getty Research Institute
- Unidades simples independientes
- Album Fotográfico Mexicano
- Imágenes de Orizaba
 - Antes de la llegada del ferrocarril
 - Edificios
 - ◊ Coleccionista Dante Hernández
 - ◊ Varios
 - ◊ Juan Vasallo
 - ◊ Winfield Scott
 - ◊ No conocidos
 - Eventos sociales
 - Ferrocarril
 - ◊ Alfred Briquet
 - ◊ Grabado
 - ◊ Gove y North
 - ◊ Julio Michaud
 - ◊ Varios
 - Vistas de la ciudad de Orizaba
 - ◊ Alfred Briquet
 - ◊ Julio Michaud
 - ◊ Gove y North
 - ◊ Winfield Scott
 - ◊ Varios
 - ◊ No conocidos
 - ◊ W. H. Jackson
 - Industria
 - ◊ Dante Hernández
 - ◊ Varios
 - ◊ C. B. Waite
 - Paisaje y producción rural
 - ◊ Alfred Briquet
 - ◊ Coleccionista Dante Hernández
 - ◊ W. H. Jackson

- Varios
- Plazas, calles y jardines
 - Alfred Briquet
 - Coleccionista Dante Hernández
 - No conocidos
 - Juan Vasallo
 - Gove y North
 - Winfield Scott
 - Varios
- Tipos
 - Winfield Scott
 - Varios
- Fotógrafos Alemanes en América Latina
 - Instituto de Investigaciones Estéticas, México
 - Fototeca Bernardo Graff, Argentina
 - Academia Nacional de Bellas Artes, Argentina
 - Centro de Documentación de Arquitectura Latinoamericana, Argentina
 - Museu Paulista, Brasil (Haberkorn)
 - Museu Paulista, Brasil (Achtschin y Preising)
 - Photographias Artísticas (No firmados)
 - Photographias Artísticas (Con firma O. A.)
 - Photographias Artísticas (Con firma Theodor Preising)
 - Museo Franz Mayer, México
 - Instituto Mora, México
 - Colección Guillermo Kahlo imágenes de época, Galindo y Villa.

● ACERCA DEL SITIO

- El Laboratorio Audiovisual de Investigación Social del Instituto Mora
- Proyecto Preservación de Imágenes, Sistemas de información, Acceso e Investigación
- Sistema de Información para Archivos de Imágenes, Pescador
- Uso de la página web y sus contenidos
- Créditos

● PUBLICACIONES Y VÍNCULOS

- “Briquet en México”, de Alejandra Padilla
- “Ciudad, Modernidad y Fotografía: Orizaba 1872–1910”, de Eulalia Ribera y Fernando Aguayo

- “Desiré Charnay”, de Julieta Martínez
- “La firma Julio Michaud, productores y preservadores de imágenes”, de Fernando Aguayo
- “La fotografía aérea en México para el estudio de la ciudad: el crucero de El Caballito” de Lourdes Roca
- “Fotógrafos Alemanes”, de Laura Castañeda
- “Imagen, fotografía y productores franceses en México”, de Fernando Aguayo
- “Los vehículos de la diversión: los trenes de mulitas en el Distrito Federal”, de Fernando Aguayo
- “Preservación de imágenes y sistemas de información, acceso e investigación: ¿un nuevo futuro para el patrimonio fotográfico?”, de Lourdes Roca