

Anonimizacion de MRI / protocolo unidad (MRI FACE)

Tenemos un protocolo nuevo para los pacientes de la Unidad de Memoria. Las imagenes no pueden anonimizarse por defecto pues deben constar los datos de los pacientes en el PACS. No obstante para el tratamiento en e pipeline, y el posterior uso de los datos estos si que deben estar anonimizados. Cada sujeto debe identificarse solo con el numero de historia clinica interno de ACE.

Como saber las MRI de este protocolo que faltan?

Para pedir las MRI que faltan a Corachan, lo primero es bajar [el listado de solicitudes de administracion](#). Aqui hay un monton de datos que no nos interesan asi que lo que hacemos es tomar las columnas de *NHC*, *Nombre* y *Fecha MRI*, copiarlas a otro sitio y salvar el documento como CSV.

Ahora, esto esta con el encoding mal asi que hay que hacer,

```
$ tr -d '\r' < mriface_faltan_raw.csv > mriface_faltan_tr.csv
```

y para acomodar un poco la lista y facilitar la conversion de *singovia*,

```
$ while read line; do simp=$(echo ${line} | sed 's/"//g; s/\([0-9]*\),\(.*\),[0-9]*\/[0-9]*\/[0-9]*\1;\2_NHC_\1/; s/,//g; s/ /_/g'); echo "${simp}"; done < mriface_faltan_tr.csv | sed 's/;/,/ ' > mri_face_faltan_encode.csv
$ sort -t, -k 1 -n mri_face_faltan_encode.csv > mri_face_faltan_encode_sorted.csv
$ sort -t, -k 1 -n mriface_faltan_tr.csv > mriface_faltan_sorted.csv
$ join -t, mri_face_faltan_encode_sorted.csv mriface_faltan_sorted.csv > mriface_faltan_20220707.csv
$ ./csv2xls.pl mriface_faltan_20220707.csv
```

El archivo resultante, *mriface_faltan_20220707.xls* es el que se envia a Corachan para reclamar las imagenes.

Esquema del protocolo

1. La peticion de MRI se hace desde la unidad
2. Corachan realiza y envia al servidor el MRI hecho
3. Los archivos se anonimizan y suben a XNAT guardando *numero de historia clinica interno*, *fecha de MRI* y *codigo interno de corachan* (por trazabilidad)

Preguntas

- Como identificar el numero de historia correspondiente a una MRI si los archivos originales vienen con nombre y apellidos?  ([como_localizar_el_codigo_interno](#))

Procedimiento

La imagen llega en un zip (en principio) y hay que eliminar los datos del sujeto tanto del dicom como de los nombres de archivo.

```
[osotolongo@brick03 anonym]$ ls -l /nas/corachan/MRI_FACE/*.zip
-rw-r--r-- 1 mtejero mtejero 128104758 Nov 10 13:38
/nas/corachan/MRI_FACE/REYES CASTELLANOS SEGUNDO.zip
-rw-r--r-- 1 mtejero mtejero 94390961 Nov 10 13:38
/nas/corachan/MRI_FACE/ROSA RUIZ MARIA CARMEN.zip
[osotolongo@brick03 anonym]$ ls -l /nas/corachan/MRI_FACE/REYES\
CASTELLANOS\ SEGUNDO | head
total 384224
-rwxr--r-- 1 root root 262746 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.100.2021.11.10.13.31.43.10.74508838
.dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.101.2021.11.10.13.31.43.10.74508849
.dcm
-rwxr--r-- 1 root root 262744 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.10.2021.11.10.13.31.43.10.74507848.
dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.102.2021.11.10.13.31.43.10.74508860
.dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.103.2021.11.10.13.31.43.10.74508871
.dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.104.2021.11.10.13.31.43.10.74508882
.dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.105.2021.11.10.13.31.43.10.74508893
.dcm
-rwxr--r-- 1 root root 262746 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.106.2021.11.10.13.31.43.10.74508904
.dcm
-rwxr--r-- 1 root root 262748 Nov 10 13:31
REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.107.2021.11.10.13.31.43.10.74508915
.dcm
[osotolongo@brick03 anonym]$ dcdump /nas/corachan/MRI_FACE/REYES\
CASTELLANOS\
SEGUNDO/REYES_CASTELLAN.MR.RM_CRANEO_PROTO.10001.100.2021.11.10.13.31.43.10.
74508838.dcm
(0x0021,0x105c) ? - Warning - Unrecognized tag - assuming explicit value
representation OK
```

```

(0x0021,0x1188) ? - Warning - Unrecognized tag - assuming explicit value
representation OK
(0x0021,0x118a) ? - Warning - Unrecognized tag - assuming explicit value
representation OK
(0x0021,0x1201) ? - Warning - Unrecognized tag - assuming explicit value
representation OK
(0x0021,0x1202) ? - Warning - Unrecognized tag - assuming explicit value
representation OK
(0x0033,0x101c) ? - Warning - Unrecognized tag - explicit value
representation is UN
(0x0002,0x0000) UL File Meta Information Group Length      VR=<UL>
VL=<0x0004> [0x000000c4]
(0x0002,0x0001) OB File Meta Information Version          VR=<OB>   VL=<0x0002>
[0x00,0x01]
(0x0002,0x0002) UI Media Storage SOP Class UID          VR=<UI>   VL=<0x001a>
<1.2.840.10008.5.1.4.1.1.4>
(0x0002,0x0003) UI Media Storage SOP Instance UID        VR=<UI>   VL=<0x003a>
<1.3.12.2.1107.5.2.50.175609.30000021102908500307500002189>
(0x0002,0x0010) UI Transfer Syntax UID                  VR=<UI>   VL=<0x0014>
<1.2.840.10008.1.2.1>
(0x0002,0x0012) UI Implementation Class UID            VR=<UI>   VL=<0x001e>
<1.3.12.2.1107.5.99.3.20080101>
(0x0002,0x0013) SH Implementation Version Name          VR=<SH>   VL=<0x0008>
<SIEMENS >
(0x0008,0x0005) CS Specific Character Set                VR=<CS>   VL=<0x000a>
<ISO_IR 100>
(0x0008,0x0008) CS Image Type                            VR=<CS>   VL=<0x002a>
<ORIGINAL\PRIMARY\M\SWI\NORM\DIS2D\MFSPLIT >
(0x0008,0x0012) DA Instance Creation Date                VR=<DA>   VL=<0x0008>
<20211029>
(0x0008,0x0013) TM Instance Creation Time                VR=<TM>   VL=<0x000e>
<083715.862500 >
(0x0008,0x0016) UI SOP Class UID                          VR=<UI>   VL=<0x001a>
<1.2.840.10008.5.1.4.1.1.4>
(0x0008,0x0018) UI SOP Instance UID                      VR=<UI>   VL=<0x003a>
<1.3.12.2.1107.5.2.50.175609.30000021102908500307500002189>
(0x0008,0x0020) DA Study Date                            VR=<DA>   VL=<0x0008> <20211029>
(0x0008,0x0021) DA Series Date                          VR=<DA>   VL=<0x0008> <20211029>
(0x0008,0x0022) DA Acquisition Date                     VR=<DA>   VL=<0x0008> <20211029>
(0x0008,0x0023) DA Content Date                         VR=<DA>   VL=<0x0008> <20211029>
(0x0008,0x002a) DT Acquisition Date Time                VR=<DT>   VL=<0x0016>
<20211029083715.862500 >
(0x0008,0x0030) TM Study Time                            VR=<TM>   VL=<0x000e> <082223.400000 >
(0x0008,0x0031) TM Series Time                          VR=<TM>   VL=<0x000e> <084238.800000 >
(0x0008,0x0032) TM Acquisition Time                     VR=<TM>   VL=<0x000e> <083715.862500
>
(0x0008,0x0033) TM Content Time                         VR=<TM>   VL=<0x000e> <084241.691000 >
(0x0008,0x0050) SH Accession Number                     VR=<SH>   VL=<0x000a> <4D21128519>
(0x0008,0x0060) CS Modality                             VR=<CS>   VL=<0x0002> <MR>
(0x0008,0x0070) LO Manufacturer                         VR=<LO>   VL=<0x0008> <Siemens >
(0x0008,0x0080) LO Institution Name                     VR=<LO>   VL=<0x0010> <CLINICA

```

```
CORACHAN>
(0x0008,0x0081) ST Institution Address      VR=<ST>    VL=<0x0024> <Calle
Buigas 19,Barcelona,,ES,08007 >
(0x0008,0x0090) PN Referring Physician's Name  VR=<PN>    VL=<0x0000> <>
(0x0008,0x1010) SH Station Name            VR=<SH>    VL=<0x000a> <AWP175609 >
(0x0008,0x1030) LO Study Description       VR=<LO>    VL=<0x0014> <RM CRANEO
PROTOCOLO >
(0x0008,0x1032) SQ Procedure Code Sequence  VR=<SQ>    VL=<0xffffffff>
----:
  > (0x0008,0x0100) SH Code Value          VR=<SH>    VL=<0x0006> <711431>
  > (0x0008,0x0102) SH Coding Scheme Designator  VR=<SH>    VL=<0x0004>
<agfa>
  > (0x0008,0x0104) LO Code Meaning        VR=<LO>    VL=<0x0014> <RM CRANEO
PROTOCOLO >

(0x0008,0x103e) LO Series Description       VR=<LO>    VL=<0x000a> <SWI_Images>
(0x0008,0x1040) LO Institutional Department Name  VR=<LO>    VL=<0x0008>
<DEFAULT >
(0x0008,0x1050) PN Performing Physician's Name  VR=<PN>    VL=<0x001a>
<VIVAS LARRUY, ASSUMPTA^^^ >
(0x0008,0x1070) PN Operators' Name         VR=<PN>    VL=<0x0006> <YELILE>
(0x0008,0x1090) LO Manufacturer's Model Name  VR=<LO>    VL=<0x000e>
<MAGNETOM Vida >
(0x0008,0x1110) SQ Referenced Study Sequence  VR=<SQ>    VL=<0xffffffff>
----:
  > (0x0008,0x1150) UI Referenced SOP Class UID      VR=<UI>    VL=<0x0018>
<1.2.840.10008.3.1.2.3.1>
  > (0x0008,0x1155) UI Referenced SOP Instance UID  VR=<UI>    VL=<0x0036>
<1.2.124.113532.80.22194.20519.20211029.81929.639296747>

(0x0008,0x1120) SQ Referenced Patient Sequence  VR=<SQ>    VL=<0xffffffff>
----:
  > (0x0008,0x1150) UI Referenced SOP Class UID      VR=<UI>    VL=<0x0018>
<1.2.840.10008.3.1.2.1.1>
  > (0x0008,0x1155) UI Referenced SOP Instance UID  VR=<UI>    VL=<0x0036>
<1.2.124.113532.80.22194.20519.20211029.74204.639210175>

(0x0008,0x1250) SQ Related Series Sequence     VR=<SQ>    VL=<0xffffffff>
----:
  > (0x0008,0x1140) SQ Referenced Image Sequence    VR=<SQ>
VL=<0xffffffff>
----:
  > (0x0008,0x1150) UI Referenced SOP Class UID      VR=<UI>    VL=<0x001c>
<1.2.840.10008.5.1.4.1.1.4.1>
  > (0x0008,0x1155) UI Referenced SOP Instance UID  VR=<UI>    VL=<0x0036>
<1.3.12.2.1107.5.2.50.175609.2021102908423878656614886>
  > (0x0008,0x1160) IS Referenced Frame Number     VR=<IS>    VL=<0x0004>
<100 >

  > (0x0020,0x000d) UI Study Instance UID          VR=<UI>    VL=<0x0036>
```

```

<1.2.124.113532.80.22194.20519.20211029.81929.639296747>
  > (0x0020,0x000e) UI Series Instance UID   VR=<UI>   VL=<0x003c>
<1.3.12.2.1107.5.2.50.175609.2021102908371871629714765.0.0.0>
  > (0x0040,0xa170) SQ Purpose of Reference Code Sequence   VR=<SQ>
VL=<0xffffffff>
  ----:
    > (0x0008,0x0100) SH Code Value   VR=<SH>   VL=<0x0006> <121326>
    > (0x0008,0x0102) SH Coding Scheme Designator   VR=<SH>   VL=<0x0004>
<DCM >
  > (0x0008,0x0104) LO Code Meaning   VR=<LO>   VL=<0x001c> <Alternate
SOP Class instance>

(0x0010,0x0010) PN Patient's Name   VR=<PN>   VL=<0x001c> <REYES
CASTELLANOS^SEGUNDO^^ >
(0x0010,0x0020) LO Patient ID   VR=<LO>   VL=<0x000a> <D21515256 >
(0x0010,0x0021) LO Issuer of Patient ID   VR=<LO>   VL=<0x0008> <UNKNOWN
>
(0x0010,0x0030) DA Patient's Birth Date   VR=<DA>   VL=<0x0008> <19451117>
(0x0010,0x0032) TM Patient's Birth Time   VR=<TM>   VL=<0x0006> <000000>
(0x0010,0x0040) CS Patient's Sex   VR=<CS>   VL=<0x0002> <M >
(0x0010,0x1010) AS Patient's Age   VR=<AS>   VL=<0x0004> <075Y>
(0x0010,0x1020) DS Patient's Size   VR=<DS>   VL=<0x0004> <1.8 >
(0x0010,0x1030) DS Patient's Weight   VR=<DS>   VL=<0x0004> <100 >
(0x0012,0x0062) CS Patient Identity Removed   VR=<CS>   VL=<0x0002> <NO>
(0x0018,0x0015) CS Body Part Examined   VR=<CS>   VL=<0x0004> <HEAD>
(0x0018,0x0020) CS Scanning Sequence   VR=<CS>   VL=<0x0002> <GR>
(0x0018,0x0021) CS Sequence Variant   VR=<CS>   VL=<0x0002> <SK>
(0x0018,0x0022) CS Scan Options   VR=<CS>   VL=<0x000a> <CG\RGR\PER >
(0x0018,0x0023) CS MR Acquisition Type   VR=<CS>   VL=<0x0002> <3D>
(0x0018,0x0024) SH Sequence Name   VR=<SH>   VL=<0x0008> <*swi3d1r>
(0x0018,0x0025) CS Angio Flag   VR=<CS>   VL=<0x0002> <N >
(0x0018,0x0050) DS Slice Thickness   VR=<DS>   VL=<0x0004> <1.2 >
(0x0018,0x0080) DS Repetition Time   VR=<DS>   VL=<0x0002> <35>
(0x0018,0x0081) DS Echo Time   VR=<DS>   VL=<0x0002> <20>
(0x0018,0x0083) DS Number of Averages   VR=<DS>   VL=<0x0002> <1 >
(0x0018,0x0084) DS Imaging Frequency   VR=<DS>   VL=<0x000a> <123.189283>
(0x0018,0x0085) SH Imaged Nucleus   VR=<SH>   VL=<0x0002> <1H>
(0x0018,0x0087) DS Magnetic Field Strength   VR=<DS>   VL=<0x0002> <3 >
(0x0018,0x0089) IS Number of Phase Encoding Steps   VR=<IS>   VL=<0x0004>
<223 >
(0x0018,0x0091) IS Echo Train Length   VR=<IS>   VL=<0x0002> <1 >
(0x0018,0x0093) DS Percent Sampling   VR=<DS>   VL=<0x0002> <90>
(0x0018,0x0094) DS Percent Phase Field of View   VR=<DS>   VL=<0x0008>
<86.1111 >
(0x0018,0x0095) DS Pixel Bandwidth   VR=<DS>   VL=<0x0004> <181 >
(0x0018,0x1000) LO Device Serial Number   VR=<LO>   VL=<0x0006> <175609>
(0x0018,0x1020) LO Software Version(s)   VR=<LO>   VL=<0x000e> <syngo MR
XA20 >
(0x0018,0x1030) LO Protocol Name   VR=<LO>   VL=<0x0016>
<t2_fl3d_tra_p2_swi_1.2>

```

.....

Hay varias alternativas para anonimizar los DICOM. Lo mas sencillo parece usar [dcanon](#) de [dicom3tools](#).

El call es mas o menos,

```
$ dcanon srcdir dstdir nodesc,nomove newpatientname newpatientid
```

El *newpatientname* deberia ser el codigo interno e idealmente vendra dentro del DICOM. Esto Corachan lo envia dentro de los comentarios,

```
(0x0010,0x4000) LT Patient Comments          VR=<LT>    VL=<0x000c> <NHC  
20211169>
```

Por otra parte, el *Patient ID* y el *Acquisition Date* nos aportan el resto de datos que necesitamos,

[preanon.sh](#)

```
#!/bin/bash

src=$1
shift

tdir=$(mktemp -t -d dcm.XXXXXXXXXX)
outdir=$(mktemp -t -d anon.XXXXXXX)
unzip "${src}" -d ${tdir}
hfile=$(find ${tdir} -type f | head -n 1)
patid=$(dckey -k "PatientID" "${hfile}" 2>&1 | sed 's/[[:space:]]//g')
sdate=$(dckey -k "AcquisitionDate" "${hfile}" 2>&1 | sed
's/[[:space:]]//g')
nhc=$(dckey -k "(0x0010,0x4000)" "${hfile}" 2>&1 | awk -F"NHC " '{print
$2}')
dcanon ${tdir} ${outdir}/${nhc}/${sdate} nomove ${nhc} ${patid}

rm -rf ${tdir}
rm -rf ${outdir}
```

y cuando hacemos,

```
[osotolongo@brick03 anonym]$ ./preanon.sh /nas/corachan/MRI_FACE/REYES\  
CASTELLANOS\ SEGUNDO.zip /old_nas/MRIFACE/
```

Tenemos entonces,

```
/old_nas/MRIFACE/  
└─ 20211475  
   └─ 20211029  
      └─ 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1000.dcm
```

```
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1001.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1002.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1003.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1004.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1005.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1006.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1007.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1008.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.1009.dcm
— 1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.7728.447490818.100.dcm
...
...
...
```

Donde 20211475 es el interno que viene en el DICOM y 20211029 la fecha en que se hace la MRI. De esta manera, cada MRI de este sujeto quedara guardada bajo la fecha correspondiente.

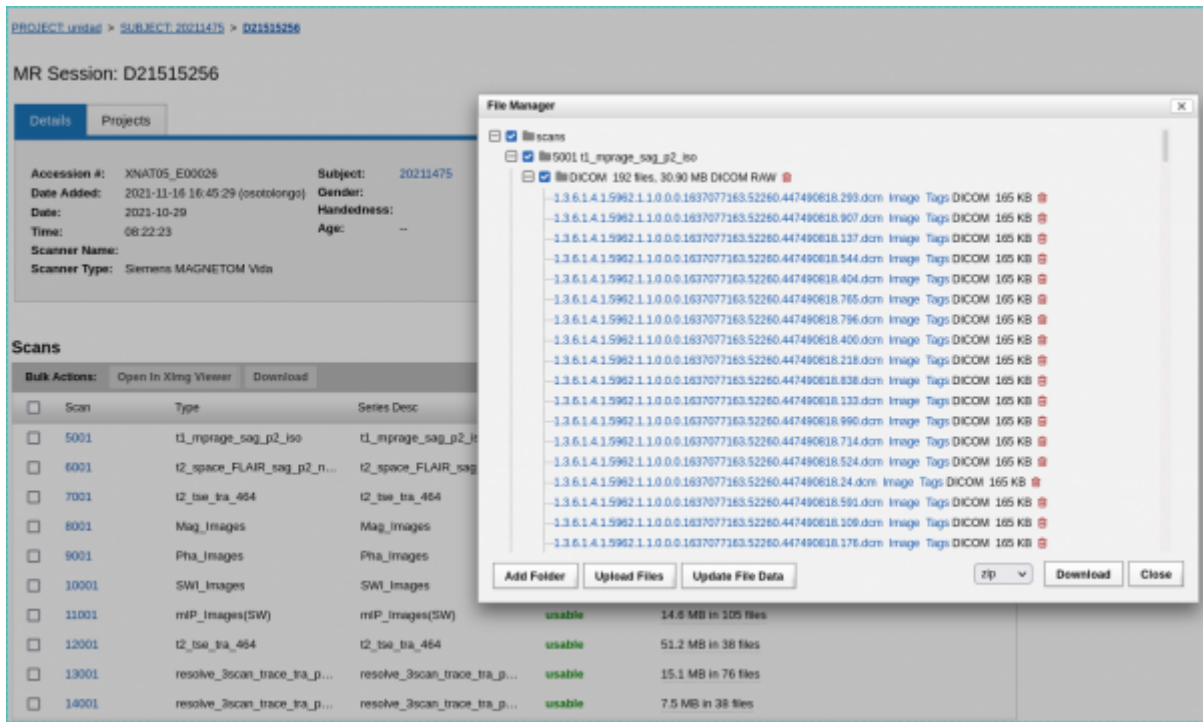
La info correspondiente queda tambien asignada en el DICOM, donde se puede ver que se respeta el *Patient ID* que viene de Corachan,

```
[osotolongo@brick03 anonym]$ dckey -k "PatientID"
/old_nas/MRIFACE/20211475/20211029/1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.772
8.447490818.1000.dcm
D21515256
[osotolongo@brick03 anonym]$ dckey -k "PatientName"
/old_nas/MRIFACE/20211475/20211029/1.3.6.1.4.1.5962.1.1.0.0.0.1637074134.772
8.447490818.1000.dcm
20211475
```

Ahora,

```
[osotolongo@brick03 anonym]$ xnatapic upload_dicom --project_id unidad --
subject_id 20211475 /old_nas/MRIFACE/20211475/20211029
```

Entonces el sujeto queda almacenado adecuadamente en XNAT



y ahora para procesar usamos el *PatientID* que venia desde Corachan,

```
[osotolongo@brick03 anonym]$ xnatapic run_pipeline --project_id unidad --  
pipeline RunFreesurfer --experiment_id D21515256  
[osotolongo@brick03 anonym]$ queue | grep xnat  
214521 fast RunFreesurfer.XNAT05_E00026 xnat R  
2:19 1 brick01
```

Asi que podemos unificar todos estos procesos en el mismo script y borrar los archivos anonimizados ya que quedan en XNAT,

[preanon.sh](#)

```
#!/bin/bash  
  
src=$1  
shift  
  
tdir=$(mktemp -t -d dcm.XXXXXXXX)  
outdir=$(mktemp -t -d anon.XXXXXXX)  
unzip "${src}" -d ${tdir}  
hfile=$(find ${tdir} -type f | head -n 1)  
patid=$(dckey -k "PatientID" "${hfile}" 2>&1 | sed 's/[[:space:]]//g')  
sdate=$(dckey -k "AcquisitionDate" "${hfile}" 2>&1 | sed  
's/[[:space:]]//g')  
nhc=$(dckey -k "(0x0010,0x4000)" "${hfile}" 2>&1 | awk -F"NHC " '{print  
$2}')  
dcanon ${tdir} ${outdir}/${nhc}/${sdate} nomove ${nhc} ${patid}  
xnatapic upload_dicom --project_id unidad --subject_id ${nhc} --  
pipelines ${outdir}/${nhc}/${sdate}  
rm -rf ${tdir}
```

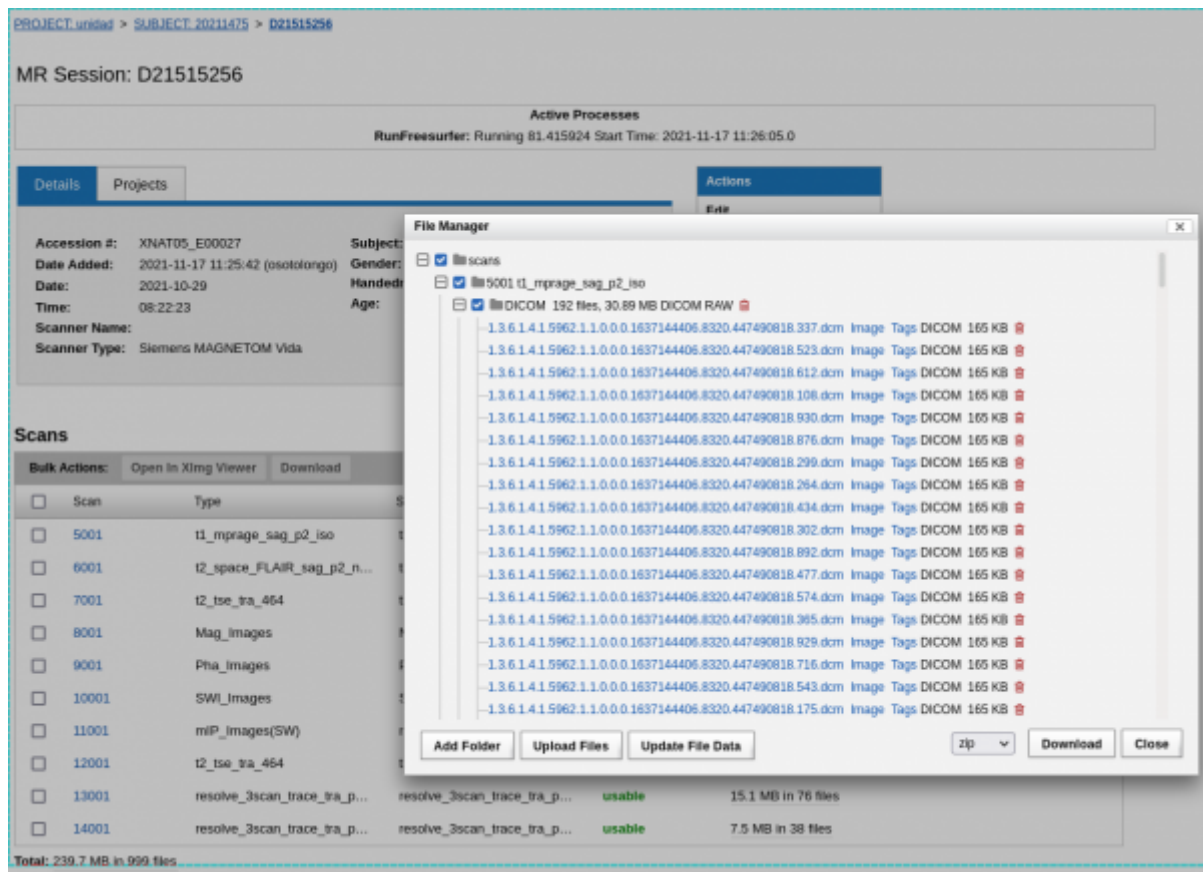


```
rm -rf ${outdir}
```

y solo habria que hacer,

```
[osotolongo@brick03 anonym]$ ./preanon.sh /nas/corachan/MRI_FACE/REYES\
CASTELLANOS\ SEGUNDO.zip
```

para que todo quede almacenado y enviado a procesar automaticamente.



Como localizar el codigo interno

Hay que lanzar en el servidor la query,

```
SELECT
  xapellido1,
  xapellido2,
  xnombre,
  his_interno

FROM [UNIT4_DATA].[imp].[vh_pac_gral]

WHERE (xapellido1 LIKE 'ALARCON'
AND xapellido2 LIKE 'MARTIN'
AND xnombre LIKE 'EMILIO')
```

que utilizando el [cliente de MS SQL](#) seria parecido a esto,

```
sqlcmd -U osotolongo -P XXXXXXXX -S 172.26.0.61 -s , -W -Q "SELECT xapellido1, xapellido2, xnombre, his_interno FROM [UNIT4_DATA].[imp].[vh_pac_gral] WHERE (xapellido1 LIKE 'ALARCON' AND xapellido2 LIKE 'MARTIN' AND xnombre LIKE 'EMILIO')"
```

pero tras probarlo veo que no puedo automatizarlo. Los nombres se escriben de casi cualquier manera y para una query puede **resultados disimiles**.

Lo que si puedo hacer es añadir una query de comprobacion al final del script,

```
sqlcmd -U osotolongo -P XXXXXXXX -S 172.26.2.161 -s "," -W -Q "SELECT xapellido1, xapellido2, xnombre, his_interno FROM [UNIT4_DATA].[imp].[vh_pac_gral] WHERE his_interno = '${nhc}';"
```

Y puedo comprobar visualmente si coinciden los nombres en la imagen y en la DB.

Para hacer un *bulk check* de los codigos podemos hacer algo como,

```
[osotolongo@brick03 mri_face]$ for x in /nas/corachan/mri_face_mayo/PAC\MAYO\ 2022/*; do n0=$(echo ${x} | awk -F'NHC_' '$2 ~ /^[0-9]+$/ {print $2}'); if [ ${n0} != '' ]; then ./query.sh ${n0} | grep ${n0}; fi; done 2>/dev/null
```

que, quitando los errores, devuelve una lista simple de los nombres y los NHC reales en la base de datos (**si todo esta OK**).

Nota: si hay algun NHC mal, no aparecera en esta lista, por lo que despues hay que revisar el directorio a ver si algo falta por subir. Lo mejor es ir consultando el listado del directorio y este output y los que falten se pueden mirar haciendo algo como,

```
[osotolongo@brick03 mri_face]$ sqlcmd -U XXXXXXXX -P XXXXXXXX -S 172.26.2.161 -s "," -W -Q "SELECT xapellido1, xapellido2, xnombre, his_interno FROM [UNIT4_DATA].[imp].[vh_pac_gral] WHERE xapellido1 like 'ALARCON' and xapellido2 like 'MARTIN';"
```

tl;dr

siempre deberia existir una copia del script en

<https://github.com/asqwerty666/acenip/blob/main/tools/preanon.sh>

y entonces hacemos,

```
~$ ./preanon.sh mi_mri.zip
```

y queda almacenado y procesado en el proyecto *unidad* de XNAT. Además el script devolvera nombre y apellidos del sujeto, segun el NHC almacenado en el DICOM para una doble comprobación.

From:
<http://imagen.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:
<http://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:preanonym&rev=1660640578>

Last update: **2022/08/16 09:02**

