

Automatizar 10 minutos al día

objetivo

a partir de las imágenes en local (descartado el sistema a través de FEEDLY, que da muchos «image not found»), automatizar el proceso de subida, importación, salvaguarda en la plataforma, sincronización con copia local, búsqueda de duplicados y eliminación de los mismos

método

el proceso está dividido en dos partes:

1. subida imágenes, importación a la BDD, eliminado imágenes subidas (en caso de no mostrar errores)
2. proceso de sincronización, búsqueda duplicados y mantenimiento de los mismos

```
alias 10minutosProcesoCompleto=«~/myscripts/10minutos/10minutos-upload.sh &&  
~/myscripts/10minutos/10minutos-dupes.sh»
```

1

```
#!/bin/bash

#
#       Sincroniza fotos candidatas con plataforma 10m, realiza importación, deja
#       directorios de importación apunto para próximo uso
#

#
#       variables
#

L1_PATH="/home/mate/Baixades/10minutos"
R1_PATH="/home/diezminutos/import"
LOG_FILE="$L1_PATH/10minutos.log"

#
#       MAIN
#

#       sincroniza ficheros plataforma 10 minutos (/home/mate/myscripts/10minutos-
sync.sh)
echo -e "Copiando imágenes en diezminutos...\n"
scp -r -i $HOME/.ssh/diezminutos@bungalow.dreamhost.com $L1_PATH/*
diezminutos@bungalow.dreamhost.com:$R1_PATH

#       importar fotos
echo -e "Lanzando petición para importación...\n"
```

```
/usr/bin/wget --delete-after http://10minutosaldia.com/maintenance/import_fotos  
STATUS=$?
```

```
#      limpiando directorios remotos para próximo uso  
#      comprobar petición OK  
if [ $STATUS -eq 0 ]; then  
    echo -e "Lanzando petición de limpieza del directorio...\n"  
    ssh -v -i /home/mate/.ssh/diezminutos@bungalow.dreamhost.com  
diezminutos@bungalow.dreamhost.com "~/10m-upload-and-update.sh"  
    STATUS=$?  
else  
    echo -e "ERROR wget!\n"  
    exit 1  
fi  
  
#      limpiar directorios locales  
#      comprobar petición OK  
if [ $STATUS -eq 0 ]; then  
    echo -e "Limpiando directorios de imágenes locales...\n"  
    find $L1_PATH/ -type f -delete;  
    STATUS=$?  
else  
    echo -e "ERROR SSH!\n"  
    exit 1  
fi  
  
if [ $STATUS -eq 0 ]; then  
    echo -e "proceso finalizado felizmente\n"  
else  
    echo -e "Algún error final detectado\n"  
    exit 1  
fi
```

2

```
#!/bin/bash  
  
#  
# Sincroniza 10m-Local, busca duplicados, genera el TXT y el subsecuente YML para  
# generar los HTML a través de mustache  
#  
#  
# variables  
#  
  
PATH1="/media/mate/WD2-3TB-EXT4/10minutos"  
MAX_IMAGENES=100 # número de líneas de cada fichero YML (y por lo tanto HTML).  
Entero (1500 duplicados) no carga!  
OTHER_BIN="/home/mate/myscripts/10minutos"
```

```
#
# variables rsync
#

R_PATH=~/uploads/"
L_PATH="/media/mate/WD2-3TB-EXT4/10minutos"
L2_PATH="$L_PATH/uploads/"
LOG_FILE="$L_PATH/10minutos.log"
BROWSER_PATH="/usr/bin/firefox"

#
# MAIN
#

# sincroniza ficheros plataforma 10 minutos (/home/mate/myscripts/10minutos-
sync.sh)
echo -e "Sincronizando...\n"
rsync -vrt --delete --exclude-from "$L_PATH/exclude.txt" -e "ssh -i
$HOME/.ssh/diezminutos@bungalow.dreamhost.com"
diezminutos@bungalow.dreamhost.com:$R_PATH $L2_PATH >> $LOG_FILE

# busca duplicados
echo -e "Dupes Searching...\n"
findimagedupes -t=95% --prune -f=$L_PATH/findimagedupes-10minutos.db
$L_PATH/uploads > $L_PATH/dupes.txt 2> /dev/null

# generar fichero .yuml a partir de duplicados
echo -e "Generando YUML...\n"
LINEAS=$(cat $L_PATH/dupes.txt | wc -l)
FICHEROS=$(( $LINEAS / $MAX_IMAGENES ))

if (( $FICHEROS > 0 )); then
    echo -e "Existen varios ficheros para procesar...\n"
    split -d --lines=$MAX_IMAGENES $L_PATH/dupes.txt $L_PATH/dupes_ --additional-
suffix=.txt
    for i in `seq 0 $FICHEROS` ;
    do
        if (( $i <= 9 )); then
            SUFIJO="0$i"
        else
            SUFIJO="$i"
        fi
        $OTHER_BIN/findimagedupes-YML-generator.sh $L_PATH/dupes_$SUFIJO.txt
    done
else
    $OTHER_BIN/findimagedupes-YML-generator.sh $L_PATH/dupes.txt
fi

# generar HTML con .yuml y .mustache
echo -e "Generando HTML con mustache...\n"

if (( $FICHEROS > 0 )); then
    split -d --lines=100 $L_PATH/dupes.txt $L_PATH/dupes_ --additional-suffix=.txt
    for i in `seq 0 $FICHEROS` ;
```

```
do
    if (( $i <= 9 )); then
        SUFIJO="0$i"
    else
        SUFIJO="$i"
    fi
    mustache $L_PATH/dupes_${SUFIJO}.yml
$L_PATH/findimagedupes-10minutos.mustache > $L_PATH/10minutos-
findimagedupes_${SUFIJO}.html
done
else
    mustache $L_PATH/dupes.yml $L_PATH/findimagedupes-10minutos.mustache >
$L_PATH/10minutos-findimagedupes.html
fi

# ejecutar Chrome con resultados
$BROWSER_PATH --private-window file:///media/mate/WD2-3TB-EXT4/10minutos/10minutos-
findimagedupes.html 2>/dev/null

# eliminar ficheros temporales
if (( $FICHEROS > 0 )); then
    rm $L_PATH/dupes_*. *
fi
```

1. !/bin/bash
2. Genera fichero YML a partir del TXT generado por findimagedupes
3. Necesita un argumento: nombre_fichero.txt

2. FUNCIONES

```
cabecera() { cat > $OUTPUT « EOF — images: [ EOF
```

```
}
```

```
footer() { cat » $OUTPUT « EOF ] — EOF }
```

2. MAIN

```
filename=«$1» base_filename=${filename%%.*} OUTPUT=«$base_filename.yml» LINEAS=0 IMAGENES=0
```

1. llamada función cabecera

```
cabecera
```

```
while read -r linea do
```

```
LINEAS=$(( $LINEAS+1 ))
```

```
# deseamos: part: [ [ url: «1.jpg», name: «name1» ], [url: «2.jpg», name: «name2» ] ],
```

1. inicio cadena

```
echo «part: [ » » $OUTPUT
```

```
#genera un item por cada imagen, tantos como sean necesarios, en la cadena
```

```
for i in ${linea[@]}
do
    CRC=${i##*/}          # name ha de ser solamente el CRC, sin path!
    CRC=${CRC:0:-4}       # elimina la extensión
    echo "[ url: '$i', name: '$CRC' ]," >> $OUTPUT
    IMAGENES=$((IMAGENES+1))
done

# final cadena
echo " ]," >> $OUTPUT
```

done < «\$filename»

1. llamada función footer

footer

echo -e «LINEAS: \$LINEAS» echo -e «IMAGENES: \$IMAGENES»

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/misproyectos:automatizacion10minutos?rev=1512690686>

Last update: **07/12/2017 15:51**

