Aragón Open Data

Generación de dataset festivos de Aragón

Control del Documento

Información del Documento

|  |  |
| --- | --- |
| Identificación del Documento | **AOD\_GFA\_1.0.1** |
| Nombre del Documento | **[AOD]\_Generación\_de\_festivos\_Aragón\_v1.0.1.docx** |
| Nombre del Proyecto | Aragón Open Data |
| Cliente |  |
| Autor del Documento | Juan Manuel Ortiz |
| Versión del Documento | 1.0 |
| Estado del Documento |  |
| Fecha de Publicación | 04.05.2017 |

Registro de Cambio

| Versión | Fecha | Actualizaciones | Elaborado / Revisado por |
| --- | --- | --- | --- |
| 1.0.0 | 04.05.2017 | Versión inicial | Juan Manuel Ortiz |
| 1.0.1 | 16.08.2017 | Ampliación de festivos de Comunidad | Juan Manuel Ortiz |
|  |  |  |  |
|  |  |  |  |

Historial de Revisión / Aprobación del Documento

| Fecha | Nombre | Organización/Puesto | Comentarios |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribución del Documento Final

Las siguientes personas son designadas como destinatarios de la versión final de este documento:

| Nombre | Organización/Puesto |
| --- | --- |
| <Nombre> | <Organización/ Posición que ocupe dentro de la estructura organizacional > |
|  |  |
|  |  |
|  |  |

Índice

[1 Alcance del documento 4](#_Toc481670039)

[2 Problemática y contexto actual 5](#_Toc481670040)

[2.1 Método de actualización actual 5](#_Toc481670041)

[2.2 Generación de documentos finales 5](#_Toc481670042)

[3 Solución adoptada 6](#_Toc481670043)

[4 Proceso de generación 7](#_Toc481670044)

[4.1 Estructura del proyecto 7](#_Toc481670045)

[4.1.1 Directorio conf 7](#_Toc481670046)

[4.1.2 Directorio docs 7](#_Toc481670047)

[4.1.3 Directorio etl 8](#_Toc481670048)

[4.1.4 Directorio info 8](#_Toc481670049)

[4.1.5 Directorio log 8](#_Toc481670050)

[4.1.6 Otros directorios 8](#_Toc481670051)

[4.2 Configuración inicial 8](#_Toc481670052)

[4.3 Ejecución del proceso 9](#_Toc481670053)

[4.4 Consideraciones importantes 9](#_Toc481670054)

# Alcance del documento

Este documento explica los pasos a seguir para generar los archivos que se incluyen en el dataset de festivos de la comunidad de Aragón en la plataforma Aragón Open Data.

Para realizar la generación de los distintos ficheros solicitados a incluir en el dataset, se ha utilizado un proceso automático mediante la herramienta Talend Open Studio.

# Problemática y contexto actual

El proceso de generación de los archivos de festivos solventa la problemática de la generación actual del calendario de festivos, pues debe realizarse de una manera manual.

El dataset de festivos de Aragón contiene información susceptible de ser actualizada varias veces durante el año.

## Método de actualización actual

A la persona responsable le llega un correo con una hoja *Excel* que contiene la información sobre los festivos de las tres provincias, cada una en un documento distinto.

A su vez, se le envían varios documentos según provincia que contienen actualizaciones del documento *Excel* inicial, según se van generando los festivos en las distintas localidades.

Es necesario unir las actualizaciones al documento *Excel* final conforme van llegando.

## Generación de documentos finales

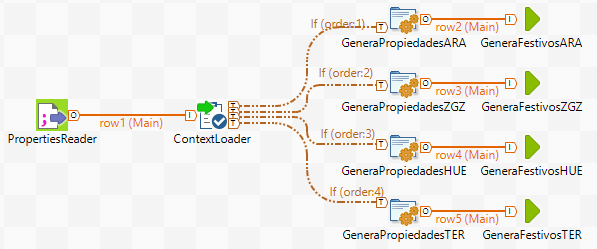
Una vez que cada documento *Excel* haya sido actualizado, existen modos, unos automáticos y otros manuales, de generar cada fichero distinto que se sube al dataset, esto es, ficheros *XLS*, *CSV*, *JSON* y *XML*. Esto es un problema porque cada proceso realiza las conversiones de manera distinta y sin ningún control de calidad del contenido final, por lo que es difícilmente explotable por entidades ajenas que quieran hacer uso de esta información. Además se quiere añadir un nuevo formato para poder importar los festivos en cualquier aplicación de calendario, ques el formato *ICS*.

# Solución adoptada

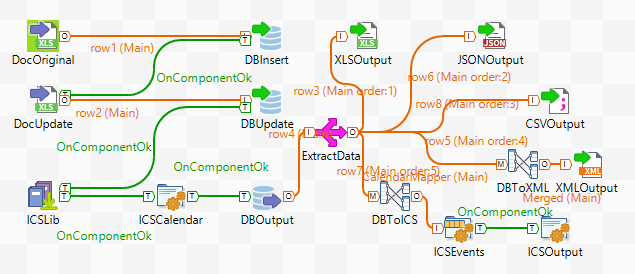
Se ha generado un proceso automático que genera la documentación requerida en base a los documentos *Excel* que llegan relacionados con los festivos, combinando las actualizaciones, con el documento actualizado la vez anterior, y extrayendo esta información a formatos *XLS, CSV, JSON, XML* e *ICS*.

Para ello se ha utilizado *Talend Open Studio 6.1.3*, que provee las herramientas para generar procesos complejos (ETLs) de una manera simple, mediante objetos que realizan las funciones que el programador necesita, diviéndolas en subtareas y añadiendo la capacidad de poder configurar todo como se necesite. Como tarea principal, se ha configurado un proceso que, a partir de un fichero de propiedades editable, configura los datos de la provincia sobre la que va a generar los archivos finales y lanza el proceso de generación. Esto da la potencia de poder generar los ficheros de todas las provincias, o de las que se necesiten. En el apartado de ejecución se explica con más detalle cómo realizar el proceso de generación.

El esquema básico de esta tarea es el siguiente:



Cada subtarea se encarga de combinar los ficheros Excel iniciales que contienen los festivos de cada provincia y sus actualizaciones y extaerlos a cada formato como se ha indicado anteriormente. Es un proceso genérico para todas las provincias y su esquema básico es este:



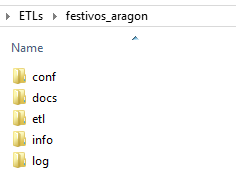
Como proceso de combinación, se ha utilizado una base de datos ligera, basada en archivos, llamada *HSQLDb*. Sobre ella se vuelcan los datos del fichero de festivos original, se actualizan esos datos con los datos del fichero de actualización realizando una combinación en la propia tabla de base de datos, y después se leen para generar los archivos de salida.

# Proceso de generación

Talend, a partir de las tareas descritas anteriormente, genera como proyecto final unos ficheros que nos permiten ejecutar el proceso de generación desde sistemas *Windows* o *Linux*, mediante ficheros *.bat* o *.sh*, lo que le da la potencia de poder ser ejecutado en cualquier sistema o entorno. Además es una aplicación portátil, la cual puede copiarse en cualquier entorno y, con una pequeña configuración, funciona y genera los ficheros deseados.

## Estructura del proyecto

El proyecto final, el cual está preparado para poder ser ejecutado tiene la siguiente estructura básica e inicial (recordemos que es configurable, por lo que cada usuario puede organizar el proyecto como desee):



### Directorio conf

Contiene el fichero de configuración ***etl-config.properties*** donde se encuentran las rutas de los directorios de entrada y salida, nombres de los ficheros, conexión a base de datos, etc.

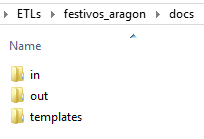
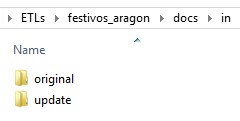
Es el fichero que debe configurarse al montar el proyecto por primera vez y poder ejecutarlo sin errores. En la versión básica, todos los directorios necesarios se encuentran dentro de la propia ETL, los cuales se pueden ver en la imagen anterior. En la sección de configuración del proceso se indica cómo configurar este archivo para el correcto funcionamiento.

Contiene además el fichero de configuración del log del proceso ***log4j.xml***, útil para seguirlo y detectar errores que puedan producirse.

Este directorio no debería moverse, pues se hace referencia (relativa) desde el fichero de ejecución.

### Directorio docs

Este directorio se compone de los directorios que guardan los ficheros de entrada, tanto el completo como la actualización, así como el directorio de salida de los ficheros:

El directorio de entrada guarda dos directorios, uno que recoge los documentos originales y otro las actualizaciones. Además, existe una carpeta templates que guarda las plantillas de salida del fichero XLS, pues debido a que el formato del documento contiene cabeceras, se deben guardar en el documento de salida para que dicho formato se mantenga. Este directorio puede moverse donde sea y configurarse donde se quiera, pues toma las rutas de la configuración. Así mismo ocurre con los ficheros de entrada y de salida, que pueden tener el nombre que más convenga.

### Directorio etl

Contiene la tarea ETL como tal, extraída directamente de Talend. En ella se encuentran las clases y archivos necesarios para que la aplicación funcione correctamente, así como los ficheros ejecutables antes comentados. Si estuviera vacía, copiar el archivo comprimido de la carpeta Project de Github y extraer el contenido del directorio interno en esta carpeta. De esta manera nos aseguraremos de mantener la misma estructura de directorios que se menciona en este manual.

**Este directorio no debe modificarse pues puede ocasionar que el proceso deje de funcionar.**

### Directorio info

Contiene, información que puede ser de utilidad para lanzar el proceso, como este documento. No es relevante en el proceso e incluso podría eliminarse si fuera necesario.

### Directorio log

Contiene el fichero de traza del proceso. Además de dar información sobre cada tarea, nos da información sobre errores que puedan producirse en algún punto, de manera que se puedan detectar y resolver con mayor rapidez.

### Otros directorios

Existe un directorio, dentro de etl, que guarda la configuración de la base de datos. Está referenciado dentro del archivo de propiedades de manera relativa, aunque gracias a la versatilidad del gestor HSQLDb, podríamos llegar a generarlo donde quisíeramos. No se recomienda moverlo con el fin de mantener la integridad de los datos y del proyecto.

## Configuración inicial

Una vez instalado el proceso en la máquina, debemos establecer una configuración básica inicial para que funcione correctamente. Para ello, editaremos el fichero ***etl-config.properties***, situado en el directorio ***/conf***. Estas son las propiedades más importantes a tener en cuenta. Todos ellos incluye una propiedad por cada provincia, acabada en \_ZGZ para Zaragoza, \_HUE para Huesca y \_TER para Teruel:

* **inXlsOriginalDocPath**: ruta de los ficheros originales. Si quisiéramos poner una ruta relativa, ésta debe ser relativa desde el fichero .bat o .sh ejecutable.
* **inXlsUpdateDocPath**: ruta de los ficheros de actualización. Si quisiéramos poner una ruta relativa, ésta debe ser relativa desde el fichero .bat o .sh ejecutable.
* **inXlsOriginalDocName**: nombre de los ficheros originales.
* **inXlsUpdateDocName**: nombre de los ficheros de actualización.
* **inOutXlsSheetName**: nombre de la hoja que contiene los datos, dentro del fichero Excel. Tomará el nombre para leer en el fichero de entrada, en el de actualización y creará una hoja con este mismo nombre en el fichero de salida.
* **outDocsPath**: ruta de los ficheros de salida.
* **outXlsDocName**: nombre del fichero con formato XLS.
* **outCsvDocName**: nombre del fichero con formato CSV.
* **outJsonDocName**: nombre del fichero con formato JSON.
* **outXmlDocName**: nombre del fichero con formato XML.
* **outIcsDocName**: nombre del fichero con formato ICS.

El resto de propiedades no es necesario configurarlas, aunque se puede hacer si se desea. En el propio fichero se explica qué hace cada propiedad.

## Ejecución del proceso

Para poder ejecutar el proceso, es necesario lanzar el proceso por lotes **.*bat*** o .***sh*** llamado ***AOD\_Festivos\_run***, situado en el directorio ***/etl/AOD\_Festivos***. Se recomienda lanzarlo desde una consola de comandos (Windows) o un terminal (Linux), de la siguiente manera:

* Nos situaremos en el directorio ***/etl/AOD\_Festivos.***
* Lanzaremos el proceso con el siguiente comando:

Windows

-------

AOD\_Festivos\_run.bat --context=Server --context\_param provincesToLoad="ALL"

Linux

-------

AOD\_Festivos\_run.sh --context=Server --context\_param provincesToLoad="ALL"

Este proceso tiene dos parámetros:

* **--context**: Indica el contexto a lanzar. En nuestro caso siempre será server.
* **--context\_param provincesToLoad**: Indica las provincias para las que se van a generar los ficheros de salida y permite los siguientes valores según necesidad:
  + Cargar todas las provincias: valor ***nulo***, ***vacío***, ***"ALL"*** o ***"ARA,ZGZ,HUE,TER"***.
  + Cargar solo la comunidad de Aragón: ***"ARA”***.
  + Cargar solo la provincia de Zaragoza: ***"ZGZ"***.
  + Cargar solo la provincia de Huesca: ***"HUE"***.
  + Cargar solo la provincia de Teruel: ***"TER"***.
  + Cargar dos provincias: poner valores de cada provincia separados por coma: ***"TER,ZGZ"***, ***"HUE,TER"*** o ***"ZGZ,HUE"***. ***No importa el orden***.

## Consideraciones importantes

Como es un proceso que toma como referencia varios ficheros *Excel*, éstos deben estar presentes en el directorio que se haya configurado como directorio de entrada, y deberán existir con el nombre que se haya indicado en el fichero de configuración. De igual modo ocurre con el directorio de salida. Una mala configuración en los archivos provocará que el proceso no se ejecute correctamente.

Debería tomarse como referencia de fichero original el anterior documento *Excel* actualizado, pues de este modo nos aseguramos que los datos siempre estén actualizados y no se pierda información. Una buena forma de tener siempre la última versión previa a la siguiente actualización sería descargar el anterior fichero Excel subido al dataset.

Al ser un documento Excel, es importante el nombre de la hoja, fila y columna donde hay que empezar a leer los datos, etc. Estos campos, excepto el nombre de la hoja, están configurados de manera fija, por lo que el documento que envíen debería tener siempre el mismo formato. Si esto no fuera así, habría que modifcar el proceso y regenerarlo, por lo que si en un futuro este problema existiera, podría llegar a poderse configurar de manera dinámica.

La manera de subir los ficheros, actualmente es manual. En un futuro podría llegar a automatizarse el proceso, para que tomara automáticamente los ficheros de entrada y cargara los de salida en el dataset.

Intentar reducir las tildes y otros caracteres no imprimibles a la hora de nombrar documentos, hojas Excel, etc., pues al ser un proceso multiplataforma, pueden darse errores en la codificación de los ficheros y ocasionar que el proceso no funcione correctamente.