Islas Canarias
Del 15 al 19 de noviembre de 2021

Automatización de hilos de computación

Esther Torres Medina
Instituto Canario de Estadística, ISTAC
etormed@gobiernodecanarias.org

José Carlos Gil Carreras
Ecosistemas Virtuales y Modulares S.L., EVM
josecarlos.gil@evm.net

Miguel Núñez Pereyra
Ecosistemas Virtuales y Modulares S.L., EVM
miguel.nunez@evm.net

Idafen Santana-Perez
Universidad de Las Palmas de Gran Canaria, ULPGC
idafen.santana@ulpgc.es

Rita Díaz Adán
Arte Consultores Tecnológico S.L., ARTE
rdiaada@arte-consultores.com

Introducción

Con el fin de ofrecer al ciudadano un mejor servicio abarcando nuevos retos en la difusión de datos, hace unos años decidimos descargar al personal del Instituto Canario de Estadística (ISTAC) de tareas que pudiesen llevarse a cabo de forma autónoma. Analizando esas cargas de trabajo, se puede observar que en muchas ocasiones se trata de actividades que se repiten periódicamente y que además se ejecutan de la misma forma, lo que las hace muy adecuadas para su automatización.

La industrialización de los procesos computacionales asociados a los datos tratados por el ISTAC permite incorporar nuevos proyectos, asegurar la repetibilidad de los procesos estadísticos, y aumentar la fiabilidad y calidad de los resultados obtenidos. Esto permitirá al personal técnico centrarse en abordar esos nuevos retos que exige la sociedad.

Objetivos

Partíamos de una aproximación en las que la mayoría de cargas eran realizadas de forma semi-automatizada, en las que varias personas tenían que coordinarse para ejecutarlas y validarlas. Esta manera de llevarlas a cabo consumía una cantidad considerable de tiempo y tenía que ser llevada a cabo por personal técnico que pudiese resolver las incidencias. La nueva estrategia tiene como finalidad la automatización de dichas cargas, realizando simultáneamente las recodificaciones y validaciones necesarias para que, desde el momento de su llegada, los datos estén a disposición de los técnicos estadísticos en el repositorio.

El objetivo final de esta industrialización es ser capaces de llegar a publicar con la menor intervención del técnico posible. De esta forma, su labor se centrará en diseñar el proceso y todo el control de calidad necesario, liberándose de repetir manualmente el mismo en cada publicación.

Metodología

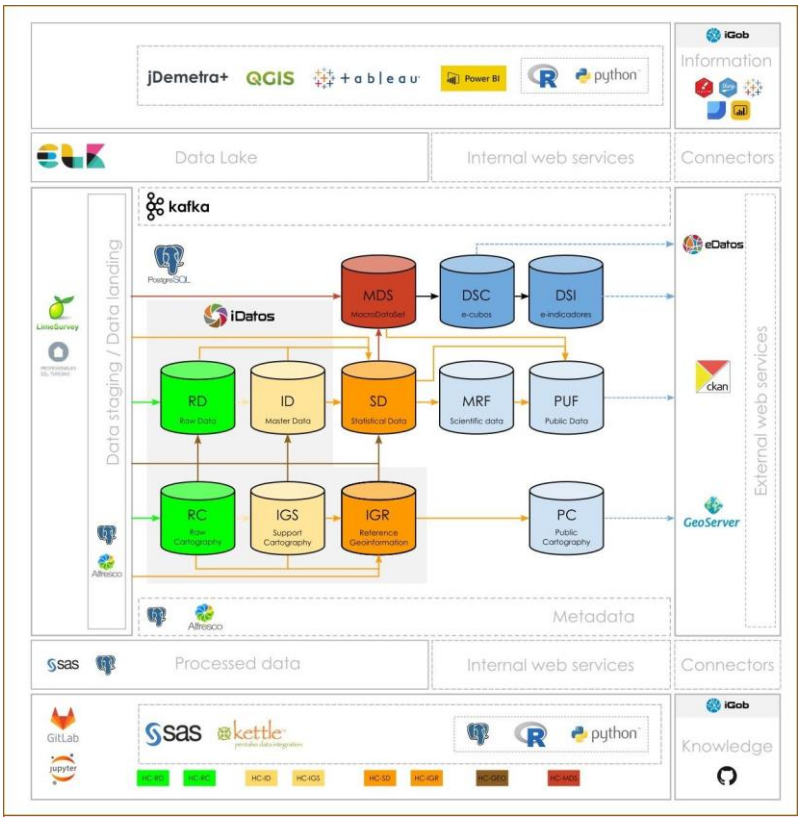
De cara a implementar esta estrategia, la herramienta elegida por el Instituto ha sido PDI (Pentaho Data Integration), herramienta de Hitachi que cuenta con versiones gratuitas, tanto del servidor como de las herramientas de desarrollo. Los desarrolladores pueden programar un flujo de datos, de forma modular, que transformará los datos de entrada en tablas en el repositorio. El desarrollo de los flujos y transformación de datos lo hacemos con la interfaz visual Spoon (Kettle) y la planificación y ejecución automática con la herramienta Pentaho Server. Además, nos apoyamos en todo un proceso de estandarización del procedimiento para llevarlas a cabo: definición de diseños de registros con un formato estándar propio; uso de clasificaciones estandarizadas; normalización en la nomenclatura de las variables, los tipos de datos a usar, el formato, etc.

En la actualidad también nos encontramos analizando Apache Hop (**H**op **O**rchestration **P**latform), herramienta creada a partir de PDI y que bajo el paraguas de la Apache Software Foundation pretende facilitar un entorno para la orquestación de datos y metadatos.

Las herramientas que utilizamos y versiones se pueden ver en la siguiente tabla, aunque nos encontramos en un entorno que está en evolución.

Funcionalidad	Herramienta	Versión
Desarrollo de procesos	Kettle	8.3
Automatización	Pentaho Server	8.3
Almacén de datos	PostgreSQL	12
Desarrollo de procesos (en estudio)	Hop	1.0
Gestión de logs	Stack ELK	

Con este fin decidimos comenzar por abordar la industrialización de la tarea de alimentar nuestro banco de datos cuya estructura se puede observar en la imagen:



Esquema del entorno de datos del ISTAC

Por una parte, tenemos una zona privada marcada en verde y tonos marrones, que comprende el almacén de los datos tal y como se reciben y sus transformaciones para un uso interno. Por otra parte, tenemos en azul los datos que se harán públicos aunque con

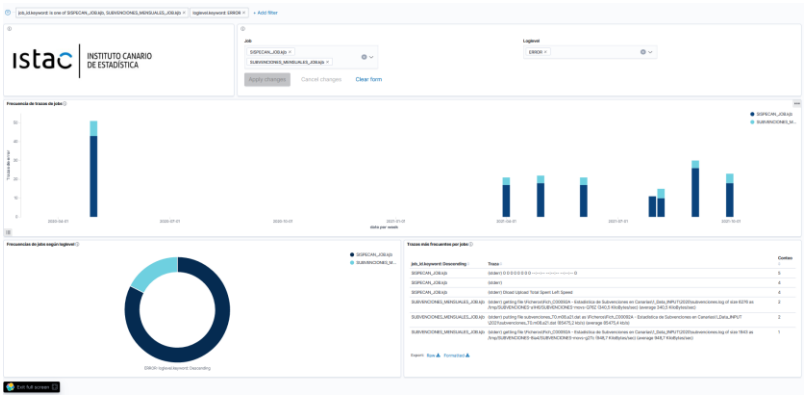
Comentado [1]: Contenido de los esquemas de cartografía (me lo pasó Mariano)
Los esquemas-tipo del Sistema de Datos Integrados en cuanto al nivel cartográfico son:
Raw Cartography (RC)
Esquemas de cartografía en crudo, con sólo tratamiento de normalización.
Support Cartography (IGS)
Esquemas de cartografía de soporte. En estos esquemas se almacenan cartografías normalizadas y depuradas, utilizadas para procesamiento y análisis espacial.
Geographic Information Reference (IGR)
Esquemas de cartografía de referencia. En estos esquemas se almacenan cartografías normalizadas, transformadas y depuradas, para su uso como entidades geográficas de difusión estadística. En este caso se almacenan distintas geometrías e información auxiliar.
Public Cartography (PC)
Esquemas de cartografía estadísticas de uso público. En estos esquemas se almacenan las cartografías públicas para su uso como entidades geográficas de difusión estadística (IGR) y otras cartografías almacenadas en IGS que puedan ser de interés analítico.

diferentes niveles de acceso según el tipo de dato y usuario al que va destinado. Las líneas que van de unos cilindros a otros son los hilos de computación, que realizan las transformaciones en los datos para pasar de un estado a otro. Actualmente, estamos trabajando en la elaboración de hilos de computación para la carga de los datos en crudo (RD), datos estadísticos (SD) e información cartográfica en diferentes niveles de madurez principalmente.

Dentro de este esquema, el proceso de trabajo puede definirse a grandes rasgos de la siguiente manera:

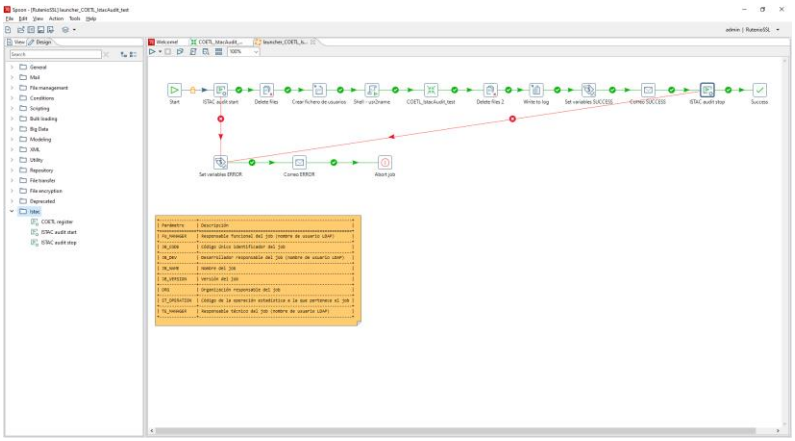
- Normalizar los diseños de registro (DSD) ajustándose a un formato que hemos definido previamente, de acuerdo a las necesidades de publicación de los datos procesados.
- Definición, y creación si es necesario, de las clasificaciones normalizadas a usar y de las tablas de recodificaciones necesarias.
- Análisis y definición de validaciones iniciales sobre los datos, de cara a asegurar la calidad de los resultados.
- Diseñar, programar y probar los procedimientos de forma local mediante una o varias ETLs en PDI, haciendo uso de los recursos generados en los puntos anteriores, y de las normas de buenas prácticas para el diseño de ETLs definidas en el ISTAC. El conjunto de normas de buenas prácticas las hemos recogido en un documento que estará accesible a todos los desarrolladores y se basa, en parte, en la guía *Naming Standards for PDI*[1].
- Automatizar los hilos de computación, una vez programados y probados. Este proceso se lleva a cabo en un principio en el servidor de cómputo de PDI, que será el sistema encargado de ejecutarlas en el momento que se especifique. Actualmente está en fase de desarrollo COETL, una herramienta diseñada para inventariar, meta-documentar y automatizar los hilos de computación de la que se habla más extensamente en esta ponencia. Con COETL se trata de aislar también el acceso al servidor ya que el grupo responsable de la automatización de hilos de computación accederá a COETL y será esta consola la que se comunique con el servidor de Pentaho o Hop para lanzar la ejecución.
- Además estamos inmersos en otro cambio con el fin de evitar accesos al servidor de producción. El seguimiento de errores se realizará con el stack ELK, de forma que el desarrollador, en caso de error en la ejecución se conectará a un dashboard de Kibana y desde ahí podrá consultar el log para ver qué ha ocurrido. De esta forma, el entorno de ejecución estará protegido y solamente tendrán acceso a él los administradores.

1 Matthew Casper, Megan Brown. 2020. Pentaho Data Integration - Naming Standards for PDI. https://support.pentaho.com/hc/article_attachments/360051574552/Naming_Standards_for_PDI.pdf



Cuadro de mando del log en Kibana

Para facilitar el procesamiento de los datos por parte de ELK y la elaboración de cuadros de mando útiles para el análisis de errores de cada ejecución, se inyectan una serie de metadatos en la ETL. Éstos serán enviados de forma automática por COETL y a través de unos pasos desarrollados por el ISTAC e introducidos en la ETL se volcarán al log.



Uso de steps para el registro de metainformación en el log

Con esta forma de trabajar, el técnico estadístico recibe una notificación cuando los datos están disponibles para su uso y no tiene que estar pendiente de realizar todo el proceso de adquisición y carga de la información.

Si bien esta estrategia aporta beneficios, cuyo impacto ya se ha visto reflejado en la gestión de datos de ISTAC, también introduce nuevos retos. Adaptar esta forma de gestión y procesamiento de datos a los diferentes casos de uso, con distintas formas de recepción y almacenamiento, análisis y depuración de errores, así como la gestión de roles, entre otros casos, requiere de un diseño y planificación coherentes.

En un futuro, la idea es no sólo automatizar las cargas sino poder llegar a crear hilos computacionales genéricos (validaciones, imputaciones, análisis de calidad,...) que nos permitan la publicación de datos de una forma sistematizada siguiendo unos criterios de calidad sólidos y bien establecidos, pudiendo llegar, en algún momento, a tener publicaciones que se realicen de forma autónoma.

Resultados

A continuación veremos dos ejemplos de procesos en los que estamos trabajando y que presentan diferente casuística.

ETL de transformación y carga de ficheros de microdatos de empleo

De las muchas ETL que se han desarrollado en el Instituto para la carga de datos en Base de Datos Postgresql vamos a describir la programada para el procesamiento, carga y posterior automatización del fichero estadístico de demandantes de empleo de SISPECAN.

Descripción de la ETL:

El desarrollo de la ETL se hace según el diseño de registro (DSD) definido para la normalización del fichero.

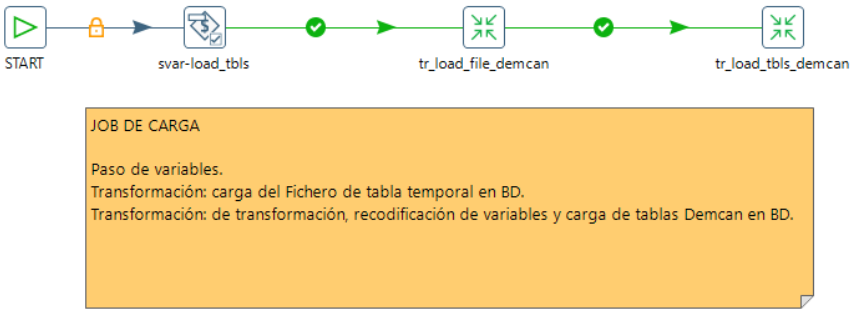
El proceso consta de dos partes, una llamada Job, donde hay un flujo de paso de parámetros y variables, a las transformaciones que realizan un proceso concreto y aglutinado. Otra, llamada transformaciones, donde se encuentran una serie de pasos, donde se realiza la computación con mapeos, cálculos y recodificaciones.

- Descripción del Job:

  En el Job se concentran los pasos para ejecutar el proceso completo de la ETL.

  En el siguiente ejemplo, se puede ver cómo siguiendo el flujo natural de trabajo, en un primer lugar, le asignamos valores a una serie de variables con un paso de parámetros, que se van utilizar en las siguientes transformaciones: ruta al fichero de carga, nombre de tablas y valores calculados de ciertas variables.

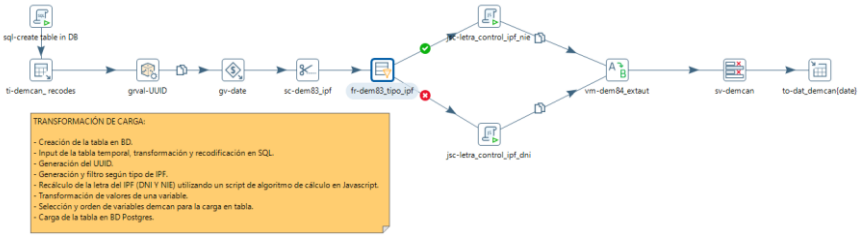
  Posteriormente, encontramos la transformación de carga del fichero demandantes de empleo, en una tabla temporal en la base de datos para su posterior transformación, recodificación y carga por parte de la siguiente transformación.



Job de carga de demandantes de empleo

- Descripción de la transformación:

  En la transformación, encontramos un serie de pasos (steps) que nos crearán la tabla para su posterior carga en el esquema de PostgreSQL indicado por el paso de variables y parámetros. Ejecutando una serie de transformaciones, filtros y recodificaciones según la normalización que consta en los DSD.



Transformación de carga de demandantes de empleo

Según el flujo natural de los pasos en la ETL, encontramos las fases del siguiente proceso:

- Creación de la tabla de carga según esquema y nombre en un script de SQL.

- Input de datos de la tabla temporal, cargada previamente, para la recodificación de variables a través de joins en proceso SQL.
  Es importante resaltar este paso como destacado en muchas de las ETL's desarrolladas. El uso de SQL en este paso es fundamental para las transformaciones y recodificaciones de las variables.
  Sería más difícil realizar éstas con otros pasos de Kettle o se incrementaría la complejidad de su elaboración y ejecución.
  Además con SQL, conseguimos obtener una mejor velocidad de proceso de computación al utilizar la potencia del motor de BBDD.

- Generación del UUID único del registro identificativo mediante random.

- Obtención de los valores de las variables a asignar como el STID y otros datos de temporalidad.

- Disección de la variable dem83 para generar el tipo, identificador y letra control del IPF.
- Filtro según tipo de IPF, para su posterior trato con el algoritmo de recálculo de la letra de control.
- Script en Javascript del algoritmo de letra de control del identificador según sea DNI o NIE.
- Mapeo de valores de entrada y salida según lo definido en el DSD de la variable dem84.
- Selección, orden y formato de las variables para la carga en tabla.
- Carga de tabla en esquema y asignación de nombre de la tabla según lo asignado en el paso de variables.

Resaltar la gran velocidad en que se gestiona la transformación y carga del flujo de datos en paquetes batch de Kettle.

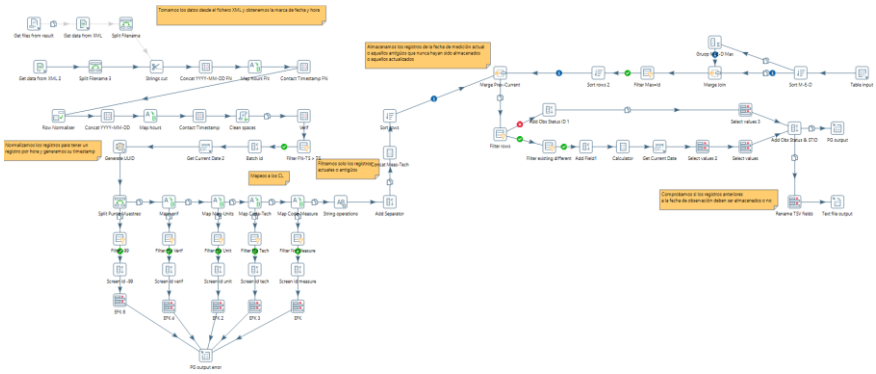
Calidad del Aire

La automatización del procesamiento y publicación de datos toma suma importancia cuando tratamos con datos de alta granularidad temporal. Un ejemplo de este tipo de datos son los datos de Calidad del Aire, gestionados por la Consejería de Política Territorial, Sostenibilidad y Seguridad del Gobierno de Canarias, así como por el Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente. Estos datos son capturados de forma horaria, con lo cual es preceptivo poder definir un proceso automático para su tratamiento.

Tomando como base los datos de entrada, en XML, se ha desarrollado una ETL, mediante un trabajo en PDI que permite la transformación de los registros de forma automática. Para la ingesta de estos datos se ha definido un trabajo en PDI encargado de recuperar los ficheros XML, publicados de forma horaria en un servidor FTP, para su recolección en local, su posterior normalización, y persistencia en una base de datos Postgres.

Trabajo de recuperación y proceso de datos

Este proceso lleva a cabo la invocación de una tarea de PDI desarrollada para llevar a cabo la parte del procesamiento de los datos, tal y como se muestra en la figura siguiente.



Procesamiento de datos horarios

A nivel general, el proceso se compone de las siguientes fases: **carga de datos,** a partir de los datos XML, **normalización:** pasando a un registro por cada hora de acuerdo al DSD, **filtrado de datos,** para descartar datos con fecha posterior al periodo actual de medición, **mapeos** para la traducción de los códigos del XML a códigos definidos por el DSD, **almacenamiento de datos actuales,** con el UUID y valor de estado de la observación para cada registro, **almacenamiento de datos pasados,** para actualizar datos ya existentes, y **gestión de errores,** para la comprobación de posibles datos erróneos y su almacenamiento en la tabla de errores.

Este proceso se ejecuta cada vez que un nuevo fichero de datos es recibido. Esto debe ocurrir cada hora, a lo largo de todos los días de la semana. No obstante, puede darse que los datos se reciban en grupos tras varias horas de medidas. Es por esto que el proceso de automatización es imprescindible para poder gestionar estos datos.

Gestión de errores

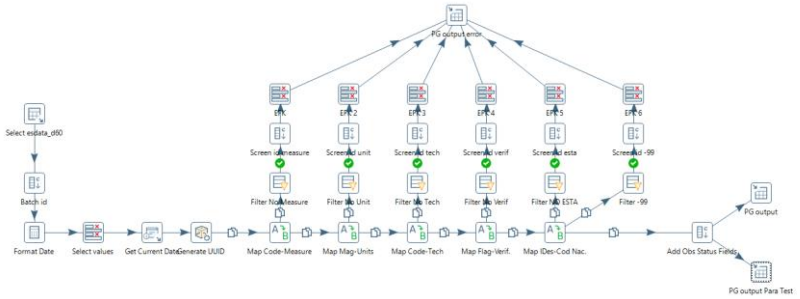
Dentro del proceso de transformación de datos es importante auditar correctamente la cantidad y naturaleza de los posibles errores que ocurran en los datos. Estos errores pueden ocurrir por diferentes razones, incluyendo fallos en el origen de los datos, fallos en la transformación aplicada, o fallo de ejecución a nivel técnico. Asimismo podemos identificar comprobaciones de error tanto a nivel de registro individual, como a niveles agregados, abarcando parcial o totalmente el conjunto de datos transformados.

La arquitectura de procesamiento de errores implementada se basa en los preceptos establecidos en *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data*[2], libro de referencia en procesamiento de datos en bases de datos relacionales

De acuerdo con los principios establecidos en el capítulo 4, se ha desarrollado un *screen* para cada comprobación, dentro de cada proceso *batch*. Como ejemplo, en la figura siguiente se muestra una ETL de carga de datos con seis *screen* de comprobación

2 Ralph Kimball and Joe Caserta. 2004. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data. John Wiley & Sons, Inc., , USA.

añadidas. Todas ellas convergen en un paso de escritura en base de datos, encargado de almacenar el identificador del registro que ha generado el error (UUID), así como el identificador del *screen* que ha generado el error y del proceso en que se está ejecutando.



Procesamiento de datos históricos

La tabla siguiente muestra un resúmen de los errores encontrados en la ejecución de la carga de datos históricos. Estos datos permiten separar el proceso de análisis de errores del procesamiento de los datos en sí.

UUID	SCREEN ID	ETL ID	FECHA
312deea9-aeb8-11e9-b6d8-a1305aeed962	ESTA_MISS_ID	T-ETL-HIST-CA-005	2019-07-25 09:42:53.382
312e3cca-aeb8-11e9-b6d8-a1305aeed962	TECH_MISS_ID	T-ETL-HIST-CA-005	2019-07-25 09:42:53.384
312fea7c-aeb8-11e9-b6d8-a1305aeed962	ESTA_MISS_ID	T-ETL-HIST-CA-005	2019-07-25 09:42:53.393
312fea7d-aeb8-11e9-b6d8-a1305aeed962	VERIF_MISS_ID	T-ETL-HIST-CA-005	2019-07-25 09:42:53.393
312fea80-aeb8-11e9-b6d8-a1305aeed962	ESTA_MISS_ID	T-ETL-HIST-CA-005	2019-07-25 09:42:53.393

Es importante resaltar que este proceso de registro de errores no interfiere con el proceso de publicación de los datos, que siguen siendo publicados, aún conteniendo potenciales errores. De esta forma el dato es entregado, y se mantiene en paralelo una auditoría de su calidad.

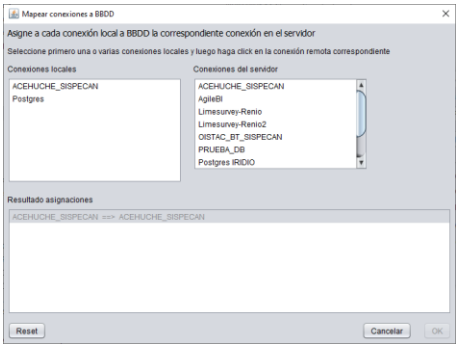
Proceso de Automatización

Una vez se ha programado y probado en local el procedimiento, se pasa a la fase de automatización. Como ya se ha mencionado, en estos momentos lo hacemos directamente en Pentaho Server y en un futuro próximo utilizaremos COETL.

Para ello, hay que realizar algunas modificaciones en el diseño para adaptarlo al entorno de ejecución del servidor. Algunos de estos pasos son realizados manualmente, pero para otros se han desarrollado herramientas internas, que facilitan las tareas de edición de los

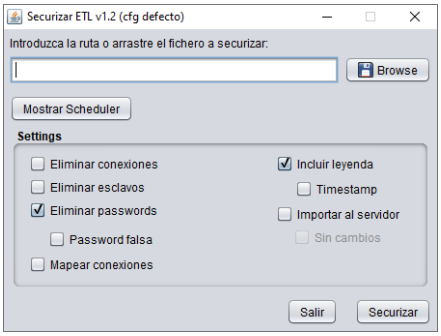
XML de jobs y transformaciones ayudando a controlar los errores que podría suponer su edición manual. Los principales cambios que se llevan a cabo son:

- Mapeo de las conexiones a bases de datos entre las que utiliza la ETL y vienen configuradas por el desarrollador, a las que tiene definidas el servidor.



Herramienta de ayuda al mapeo de conexiones

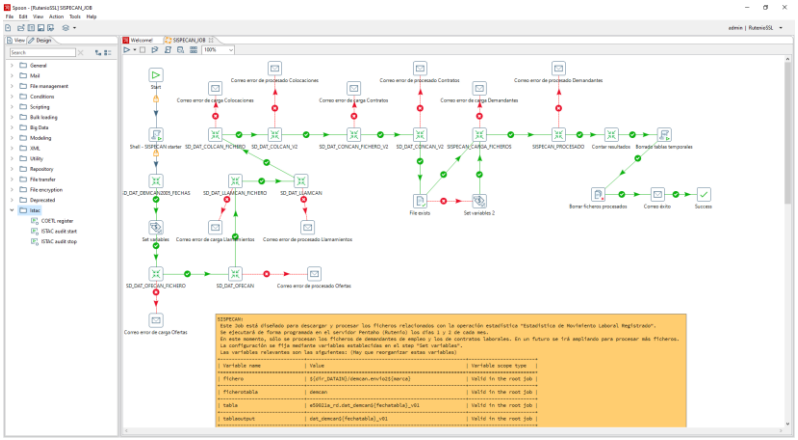
- Cambios de servidores, usuarios y contraseñas. Securización de los ficheros eliminando las credenciales de los usuarios desarrolladores ya que el servidor utilizará sus propias credenciales.



Herramienta para la securización de credenciales

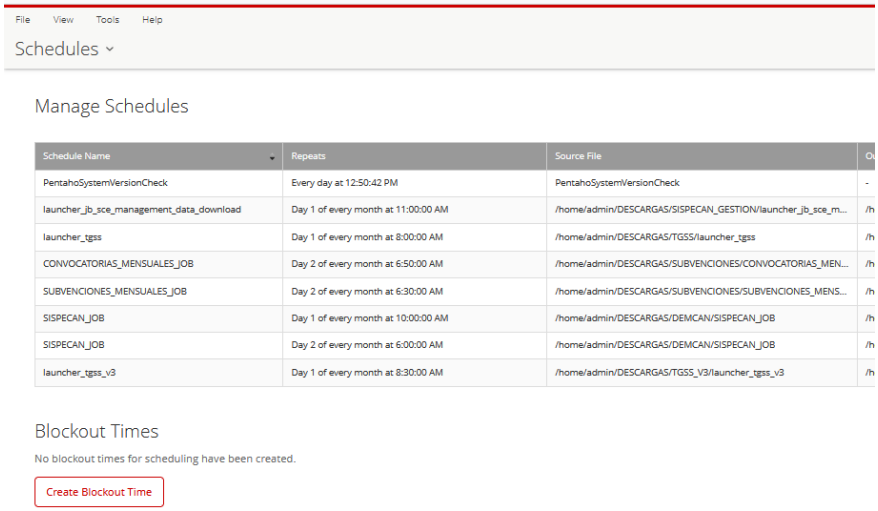
- Modificación de rutas de origen y destino de los ficheros a ser procesados. En el entorno de desarrollo, los datos de origen pueden estar en ubicaciones no accesibles desde el servidor o que ralenticen mucho el acceso a los datos. Es por esto por lo que, normalmente, los ficheros a procesar son volcados en una ruta interna de la máquina en la que está instalado el servidor y de la que se eliminan una vez realizada la carga.

- Introducción de control de errores para informar a los usuarios involucrados en el control del procedimiento e introducción de pasos de notificación de resultados a los usuarios finales de los datos dentro del ISTAC. Actualmente, se informa a los usuarios mediante el envío de un correo electrónico sobre el resultado de la ejecución. Esta parte de control se hace durante la integración de la ETL en el servidor de Pentaho y no por parte del desarrollador original.



Hilo de computación: carga de los ficheros estadísticos del SEPE

Teniendo en cuenta estos cambios, se suben al servidor los diferentes ficheros que forman parte del hilo de computación, se revisa la configuración, se realizan los cambios necesarios y se prueba. Una vez funciona correctamente se pasa a planificar de forma que se ejecute en los momentos previstos de forma autónoma.

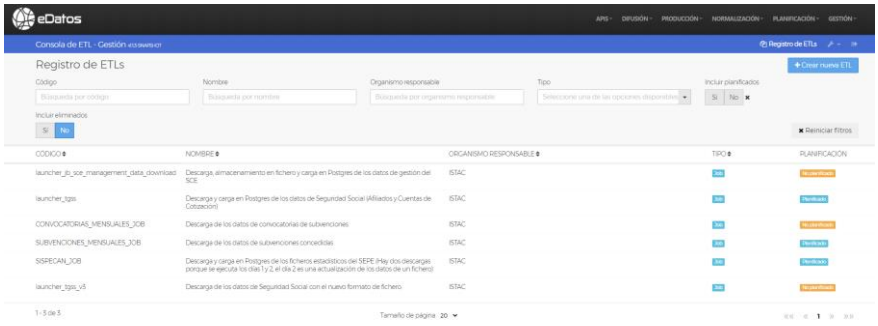


Planificación de hilos de computación

Consola de ETL

En el apartado anterior se ha descrito cómo es el procedimiento que se lleva a cabo actualmente para la automatización de los procesamientos estadísticos. Sin embargo, de cara a una industrialización del proceso, resulta fundamental contar con una herramienta que permita llevar a cabo las labores de gestión sobre las ETL. Al fin y al cabo, una ETL es un nuevo tipo de recurso que se crea dentro del organismo y, que como tal, debe ser

gestionado y meta-documentado correctamente. Con este fin nace la herramienta conocida como COETL (COnsola ETL).



Vista principal de COETL

Dentro de las funcionalidades de la herramienta COETL se encuentran las siguientes:

- Gestionar el inventario de las ETL existentes.
- Mostrar el inventario de todas las ETL existentes.
- Realizar búsquedas sobre el inventario de las ETL existentes.
- Asociar a cada ETL su correspondiente código.
- Permitir ejecutar una determinada ETL de forma manual (bajo petición del usuario).
- Permitir programar cuándo debe ejecutarse una determinada ETL de forma automatizada.
- Permitir desprogramar ETL que estaban planificadas para ejecutarse automáticamente.
- Registrar la información relativa a cada una de las ejecuciones de las ETL y si el resultado de dichas ejecuciones fue exitoso o fallido.
- Registrar la información relativa a cuándo será la próxima vez que se ejecute la ETL (en el caso de estar planificada).

Dentro de este sistema, uno de los activos más importantes es el código de cada una de las ETL. En versiones iniciales de la herramienta, el código era incluido dentro de la aplicación y asociado a su correspondiente ETL. Sin embargo, esta aproximación no resulta la más apropiada para la gestión de este tipo de activos y es por ello por lo que se opta por evolucionar el sistema de cara a la integración del mismo con un sistema de control de versiones. Esta integración aporta ventajas inmediatas como son:

- Examinar la evolución de las modificaciones que se realizan sobre los mismos.
- Consolidar versiones de los activos.
- Gestionar la seguridad y el acceso al código de forma independiente a la meta-documentación e histórico de ejecuciones.

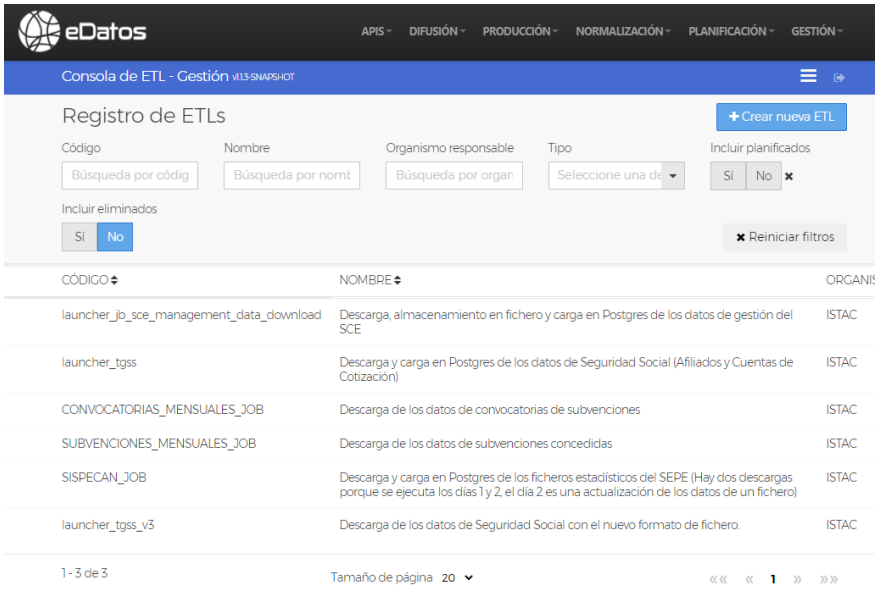
Además, es importante tener en cuenta que disponer de este tipo de piezas dentro de un sistema de cómputo resulta muy beneficioso. De hecho, dotarse de un sistema de control de versiones es el primer paso para alcanzar otros modelos más avanzados como el de una integración continua basada en datos y modelos para procesamientos estadísticos

(Smidek, 2020). Al fin y al cabo, lo que se busca es profesionalizar e industrializar la producción de datos al igual que con los años se ha hecho en el mundo del software.

En lo que a términos computacionales se refiere, COETL es capaz de gestionar ETL para dos tipos de plataformas de integración de datos diferentes: PDI (Pentaho Data Integration) y Apache Hop (Hop Orchestration Platform). En un primer momento, COETL contaba únicamente con capacidad para gestionar ETL de Pentaho. Sin embargo, dentro de las labores de vigilancia tecnológica que se llevan a cabo por parte del organismo, se detectó que la inclusión de Hop podría ser realmente beneficiosa. Si bien es cierto que en el corto plazo PDI responde a la mayor parte de las necesidades existentes, el hecho de que Apache Hop sea un proyecto que nace de PDI pero que está avalado por una comunidad activa lo convierte en una apuesta de cara al medio y largo plazo. Es por ello que, recientemente, se ha dotado al sistema de la posibilidad de ejecutar este tipo de ETL.

A nivel tecnológico cabe destacar que se trata de un sistema que se integra completamente con el resto de aplicaciones de la Infraestructura de Datos y Metadatos Estadísticos de Canarias (eDatos). Esta integración resulta fundamental para poder aprovechar el conocimiento que ya existe en otros sistemas. En concreto, el sistema se integra con:

- El inventario de operaciones estadísticas. Esta integración permite llevar a cabo una organización más eficaz puesto que permite relacionar las diferentes ETL con las operaciones estadísticas existentes.
- El sistema de envío de notificaciones. Esta integración permite llevar a cabo una comunicación efectiva sobre los resultados de las ejecuciones de las ETL. Esto se debe a que se puede notificar a todos los usuarios a los que les puede afectar una determinada ETL (por ejemplo, todos los técnicos de producción de una determinada operación estadística) y, además, a cada uno de ellos se les notifica por la vía que prefieran.
- Los módulos encargados de las labores de autenticación y autorización. De esta forma se consigue una gestión centralizada de la seguridad que puede estar incluso vinculada a las operaciones estadísticas sobre las que tienen permisos los diferentes técnicos.



Detalle del inventario de ETLs

COETL cumple por tanto el objetivo de ser la piedra angular sobre la que se lleva a cabo un exhaustivo control sobre el inventariado, la meta-documentación, las planificaciones y las ejecuciones de los procesamientos estadísticos. Se consigue así lograr los siguientes objetivos:

- Todas las ETL se encuentran centralizadas e inventariadas.
- Las ETL se encuentran documentadas.
- Las ETL se ejecutan siempre desde la consola, de esta forma se evita que los técnicos ejecuten ETL localmente sin control de las ingestas que se están realizando.
- Se tiene un historial completo de las ejecuciones que se han realizado, quién las lanzó y en qué fecha.
- La ejecución de las ETL se puede programar para que se lancen de manera automatizada.

Conclusión

La automatización de procesos permite alimentar el banco de datos de forma desatendida ajustándose a procedimientos normalizados de diseño y calidad permitiendo dirigir los esfuerzos del personal estadístico a la publicación.

Las herramientas y metodologías, en constante evolución, hacen que el entorno de trabajo sea cambiante pero hasta en esta situación los beneficios son palpables y aunque el avance no sea a la velocidad deseada, sí intentamos que cada paso nos lleve hacia nuestro objetivo.

Principales referencias bibliográficas

- Ralph Kimball and Joe Caserta (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data.*
- Matthew Casper, Megan Brown (2020, 02 28). *Pentaho Data Integration - Naming Standards for PDI.* Retrieved 12 10, 2021, from https://support.pentaho.com/hc/article_attachments/360051574552/Naming_Standards_for_PDI.pdf
- Smidek, J. (2020, 12 03). *7 steps to elevate your BI reporting infrastructure to the next level*. Towards Data Science. Retrieved 10 10, 2021, from https://towardsdatascience.com/7-steps-to-elevate-your-bi-reporting-infrastructure-to-the-next-level-229513663885