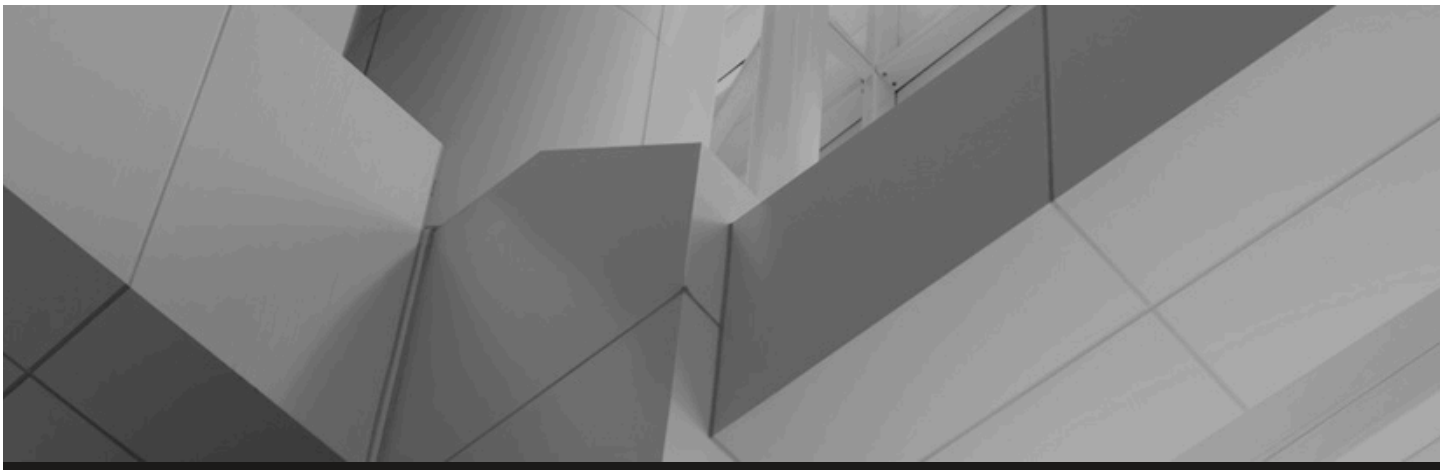

Data Ingestion

Índice

Introducción

Con el objetivo de brindar definiciones y claridad sobre ciertos conceptos y confusiones recurrentes, desde Data Ingestion Squad se elabora esta sección de la Wiki con información general y específica a tener en un cuenta al momento de realizar el desarrollo de una ingesta. La información brindada debe ser reforzada con lo elaborado por Data Engineering Chapter, ya que el uso y conocimiento de ciertas herramientas requiere de un paso a paso más detallado.

Conceptos básicos - Data Lake

Data Lake

Como repositorio centralizado para el almacenamiento de los datos se cuenta con Azure Data Lake Storage Gen2, que proporciona una solución basada en la nube para el almacenamiento en Data Lake de Microsoft Azure.

Azure Data Lake Storage

Azure Data Lake Storage combina un sistema de archivos con una plataforma de almacenamiento para grandes volúmenes de datos, haciendo uso de un espacio de nombres jerárquico que organiza mediante directorios y almacena metadatos sobre cada directorio y los archivos que contiene.

Zonas del Data Lake

El Data Lake contiene tres zonas distintas con el objetivo de diferenciar los criterios de almacenamiento y consumo de datos en cada una.

- **Prelanding Zone**: Almacena lo que se ingesta directamente desde cualquier sistema origen, como puede ser el resultado de una query a una base de datos o el response de una api. El propósito de esta zona es contener datos sin ninguna modificación o transformación, en lo posible, lo que la hace funcionar como un almacenamiento de replicas al sistema origen.

- **History Zone**: Almacena la totalidad de datos alguna vez ingestados en Prelanding Zone, pero agrega a nivel registro dos campos nuevos para brindar trazabilidad. El propósito de esta zona es contener cada uno de los

registros que se ingestaron y poder analizar los cambios que estos sufrieron a lo largo del tiempo.

- **Consume Zone**: Almacena con distintos criterios, según lo requieran los equipos, los datos alguna vez ingestados en Prelanding Zone con requisito de unicidad y la creación a nivel registro de dos campos para brindar trazabilidad. Esto quiere decir que se filtran aquellos registros duplicados por su propósito de disponibilizar los datos en su versión más reciente, para el consumo de los equipos que requieran trabajar con ellos.

Nomenclatura en las distintas zonas del Data Lake

Tanto para escribir como para navegar en el Data Lake es importante conocer la nomenclatura de cada zona, ya que cuentan con la concatenación de distintos elementos. El formato estándar es el siguiente:

- **Prelanding Zone**: `/mnt/prelandingzone/País/SistemaOrigen/Dataset/Año/Mes/Dia/Dataset`
- **History Zone**: `/mnt/historyzone/Pais/SistemaOrigen/Dataset`
- **Consume Zone**: `/mnt/consumezone/País/Contexto/SistemaOrigen/Dataset`

El formato a seguir es PascalCase, que une palabras utilizando solo mayúscula en la primera letra de cada una. Algunos ejemplos:

- **Prelanding Zone**:`/mnt/prelandingzone/Argentina/Truck/Planil/2024/10/01/Planil`
- **History Zone**: `/mnt/historyzone/Argentina/Truck/Planil`
- **Consume Zone**: `/mnt/consumezone/Argentina/Commercial/Truck/Planil`

Para el caso de Prelanding Zone, tanto el mes como el día conservan el formato con dos cifras. También pueden darse casos especiales que se mencionaran más adelante.

Acceso al Data Lake

La forma más directa de acceder y consultar el Data Lake es a través de Databricks, utilizando el comando mágico correspondiente en conjunción con la zona. Algunos ejemplos:

```
%fs ls /mnt/prelandingzone/
```

```
%fs ls /mnt/historyzone/
```

```
%fs ls /mnt/consumezone/
```

El resultado del comando es un listado de los objetos que contiene cada directorio, pudiendo extenderse la consulta al directorio completo de un dataset, de forma que listen los archivos que contiene.

Aspectos funcionales y técnicos de cada zona

Las zonas del Data Lake también se diferencian por su formato de escritura y las transformaciones que permiten.

- **Prelanding Zone**: La escritura en esta zona se realiza en formato Avro, por lo que ciertas ingestas pueden sufrir una transformación a formato tabular si lo requieren. Un ejemplo de esto último son los response obtenidos desde apis, que en su mayoría poseen el formato semiestructurado JSON, y requieren de una transformación para adaptarlos a formato tabular.

- **History Zone**: La escritura en esta zona se realiza en formato Parquet y esta particionada por el campo CREATED_DATE. Este campo es uno de los dos mencionados anteriormente como soporte para la trazabilidad de cada registro. Se genera junto al campo LAST_UPDATE, siendo CREATED_DATE la fecha en formato **YYYY-MM-DD** correspondiente a la primera vez que se escribió el registro en el Data Lake, y LAST_UPDATE siendo la fecha con horario en formato **YYYY-MM-DD hh:mm:ss** en la cual sufrió alguna modificación.

LAST_UPDATE	CREATED_DATE
2024-10-01 18:03:31	2024-10-01

> El campo CREATED_DATE no toma el horario debido a que funciona como campo partición, por lo que dentro de la carpeta correspondiente a un dataset se encontraran subdirectorios del estilo: **/CREATED_DATE=YYYY-**

MM-DD/, dando trazabilidad a lo que se escribió en Data Lake cada día, y luego permitiendo analizar los cambios a través de los días con LAST_UPDATE.

- **Consume Zone**: La escritura en esta zona también se realiza en formato Parquet, pero el particionado es opcional y puede ser a través de cualquier campo, mientras se justifique, o directamente no particionarse si no es necesario. También se cuenta con los campos CREATED_DATE y LAST_UPDATE con el mismo funcionamiento que en History Zone, aunque en formato **YYYY-MM-DD hh:mm:ss** ambos.

LAST_UPDATE	CREATED_DATE
2024-10-01 18:04:08	2024-10-01 18:04:08

Como la partición en esta zona no es necesariamente a través de CREATED_DATE, pueden encontrarse subdirectorios del estilo: **/part_anio_mes=YYYY-MM/**.

Debido a que History Zone y Consume Zone contienen datos que provienen de Prelanding Zone, para el traspaso de datos hacia ambas zonas se utiliza una función que realiza los siguientes pasos:

1. Se versiona mediante la creacion de un nuevo campo a nivel registro cada uno de los archivos Avro encontrados en Prelanding Zone, según su fecha y horario de escritura.

2. Se unen todos los archivos encontrados en un solo dataset temporal, cada uno conservando su versión a nivel registro.

3. En caso de encontrar distintas versiones, se utiliza una función ventana, que particiona por los campos clave del dataset y ordena por la versión, para obtener los registros mas recientes en caso de haber duplicados.

Esto se debe a que es muy importante preservar la unicidad de los registros a través de los campos clave del dataset. A lo que se le suma la condición de no tener valores nulos en campos declarados como clave, ya que por definición esto no es correcto.

Ambas condiciones se ponen a prueba luego del desarrollo de cada ingesta, con el objetivo de asegurar su funcionamiento una vez que los procesos estén productivos.

Traspaso de datos entre zonas - Notebooks

El traspaso de datos entre zonas del Data Lake ocurre a través de la ejecución de distintas notebooks desarrolladas en Databricks. Las últimas versiones, con los criterios correspondinetes a cada zona actualizados, son las siguientes:

- `Pasaje a History Zone: /LAS/Templates/Pz_Hz_Template_Db_V3`
- `Pasaje a Consume Zone: /LAS/Templates/Pz_Cz_Template_Db_V3`

Existe una confusión entre el pasaje de datos entre zonas y la forma de consultar el origen para ingestar los datos en primer lugar. Esto se debe a que antiguamente las notebooks se diferenciaban por maestras o incrementales. Hoy en día esa diferenciación no se utiliza debido a que no tiene relación con los criterios de cada zona.

Criterios de escritura en Consume Zone

Como se mencionó anteriormente, la escritura en Consume Zone puede variar dependiendo de lo relevado por los equipos. Se presentan las siguientes posibilidadesl, que se manejan a través del parámetro keepCz en la notebook Pz_Cz_Template_Db_V3:

- **Dataset particionado y keepCz = true**: Es el caso en el que se quiere acumular las novedades de cada ingesta a lo ya escrito en Consume Zone. Es decir que va a conservarse lo que ya estaba escrito en Consume Zone, se va a sumar lo que llegue nuevo desde Prelanding Zone, y a su vez si llegan registros con cambios que ya se encontraban en Consume Zone, estos simplemente se modifican con las novedades.

- **Dataset particionado y keepCz = false**: Este caso no es común ya que las novedades que lleguen desde Prelanding Zone van a sobrescribir lo que ya se encontraba en Consume Zone, pero solo en las particiones correspondientes.

- **Dataset sin partición y keepCz = true**: Funciona de la misma forma que el primer caso.

- **Dataset sin partición y keepCz = false**: En este caso todo lo que llegue desde Prelanding Zone simplemente sobrescribe todo lo que se encuentra en Consume Zone.

Debe tenerse en cuenta que esto es posible a través de los campos clave del dataset, ya que de otra forma no puede rastrearse la unicidad y darse la modificación en aquellos registros que lo requieran. Es decir, lo que se busca es una modificación en el resto de los campos que no están declarados como clave.

Funcionamiento de CREATED_DATE y LAST_UPDATE en Consume Zone

El primer caso del punto anterior es un buen ejemplo para entender el funcionamiento de los campos CREATED_DATE y LAST_UPDATE:

1. El día 2024-10-01 a las 00:00hs. se realizó por primera vez una ingesta, por lo que todos los registros en Consume Zone toman valor **2024-10-01 00:00:00** en los campos CREATED_DATE y LAST_UPDATE.

2. El día 2024-10-02 a las 00:00hs. se realizó una segunda ingesta. Lo que sucede es:

    - Todo lo escrito en Consume Zone del día anterior que no sufre modificaciones, conserva ambos campos con el mismo valor: **2024-10-01 00:00:00**.

    - Todo lo que llega nuevo desde Prelanding Zone respeta el mismo criterio de compartir valores en ambos campos: **2024-10-02 00:00:00**.

    - Los registros que sufren modificaciones, conserva el valor original (**2024-10-01 00:00:00**) en CREATED_DATE, toman el nuevo valor (**2024-10-02 00:00:00**) en LAST_UPDATE.

CLAVE	CAMBIO	NUEVO_EN_CZ	CREATED_DATE	LAST_UPDATE
1	No	No	2024-10-01 00:00:00	2024-10-01 00:00:00
2	Si	No	2024-10-01 00:00:00	2024-10-02 00:00:00

3	No	Si	2024-10-02 00:00:00	2024-10-02 00:00:00

Esto permite tener trazabilidad sobre los cambios a nivel registros, pudiéndose buscar lo ingestado en un día particular a partir de CREATED_DATE, o las novedades a través de LAST_UPDATE.

Casos especiales y un poco más

- Sobre vocabulario: es muy utilizado el termino path para referirse a los directorios; las abreviaciones Pz, Hz y Cz para las zonas del Data Lake; la abreviación DL para referirse al Data Lake; la abreviación NB para referirse a notebook; y suelen mencionarse los datasets escritos en Data Lake como tablas.

- Los paths en Pz pueden extenderse hasta nivel hora, e incluso minuto, de una ingesta cuando se realiza más de una en el mismo día. Un ejemplo: /mnt/prelandingzone/País/SistemaOrigen/Dataset/Año/Mes/Dia/h=Hora/m=Minutos

- Las ingestas de datasets que contengan datos de distintos países pueden almacenarse en Las. Es decir que, en lugar del nombre de un país en los paths se puede hacer referencia simplemente a Las.

- Las ingestas desde Snowflake suelen incluir el esquema del origen en los paths, entre el sistema origen y el dataset. Un ejemplo: /mnt/prelandingzone/País/SistemaOrigen/Esquema/Dataset/Año/Mes/Dataset

- Las ingestas desde Sharepoint poseen una organización diferente, ya que utilizan el nombre del mismo y el del proyecto por el cual se solicitó la misma. Un ejemplo donde Sharepoint funciona como contexto, AnalyticsDataLakeWs funciona como nombre del Sharepoint, SkyParesa como nombre de proyecto y ClientesSky como el nombre del dataset: /mnt/consumezone/Paraguay/Sharepoint/AnalyticsDataLakeWs/SkyParesa/ClientesSkyDonde Sharepoint

- Antiguamente las ingestas de archivos Excel o Csv se almacenaban con el término Manual como sistema origen. Un ejemplo: /mnt/consumezone/Bolivia/Commercial/Manual/LePeriodo

- Los contextos más utilizados son Commercial, Logistica y Sharepoint, quedando en desuso otras alternativas. Por lo que la mayoría de las ingestas

contienen Commercial como contexto.

- Pueden encontrarse muchos paths con distinta organización y uso de nomenclatura, lo que puede ser producto de procesos viejos que quedaron desactualizados o no estén relacionados a ingesta.

- Al momento de buscar el path de una ingesta puede tomarse como referencia:
  - Los campos file_dir_x de ConfigPivot.
  - Los parámetros del pipeline de ingesta.
  - Navegar a través del Data Lake con el comando correspondiente.
  - Revisar el código de las notebooks que se encargan del traspaso de datos entre zonas del Data Lake.

- Pueden encontrarse datasets que en Hz y Cz posean campos como LAST_UPDATE_PART, que quedaron en desuso y hoy se siguen generando por procesos de ingesta antiguos o desactualizados.

- Incluso pueden encontrase los campos CREATED_DATE y LAST_UPDATE con formatos distintos, por la razón anterior.

^ Conceptos básicos - Ambientes y herramientas de trabajo

Ambientes

La gestión de entornos controlados por separados es útil para gestionar desde la creación hasta la implementación de procesos.

El ambiente de desarrollo permite, valga la redundancia, el desarrollo de nuevos procesos y la realización pruebas sin comprometer los procesos ya implementados. Este ambiente resulta local para cada ingeniero a través de la utilización de ramas, donde el trabajo es en forma independiente para evitar errores y conflictos.

Por el otro lado, el ambiente de producción contiene el flujo de cada uno de los productos de datos que se entregan a los usuarios finales. Siempre debe tener una versión estable de cada proceso en ejecución y es DataOps quien se encarga

de mantenerlo estable, lo que lo aleja de los errores y conflictos que puedan generar nuevos desarrollos o etapas de prueba.

GitHub

GitHub es una plataforma de desarrollo colaborativo y alojamiento de código basada en el sistema de control de versiones Git. Permite almacenar, gestionar y compartir código en repositorios, donde se mantiene un historial de cambios y es posible colaborar con otros desarrolladores.

El repositorio las-datapipeline almacena todos archivos relacionados a los pipelines de datos correspondientes al equipo de LAS, y es gestionado por Iris. La rama master es el reflejo de lo que se encuentra en el ambiente productivo, y para cada nuevo desarrollo es necesario crear una nueva rama a partir de ella.

Flujo de trabajo

Azure Data Factory

Azure Data Factory es un servicio de integración de datos en la nube de Microsoft Azure, diseñado para automatizar y orquestar flujos de datos a gran escala. Permite mediante pipelines extraer datos de fuentes variadas (como bases de

datos SQL, archivos en Azure Blob Storage, API REST, etc), y realizar transformaciones (en nuestro caso a través de notebooks en Databricks).

Los distintos ambientes se identifican de la siguiente forma:

- **Desarrollo**: La suscripción ABI LAS Non-Prod contiene a abi-las-dev-uee-comercialdatalake-df y abi-las-dev-uee-integracion2-df.

Ambientes de desarrollo en Azure Data Factory

- **Produccion**: La suscripción Analytics contiene a adf-prod-iris-las.

Ambiente de producción en Azure Data Factory

Ambientes de desarrollo en Azure Data Factory

En Azure Data Factory desarrollo cuenta con dos entornos:

- **abi-las-dev-uee-comercialdatalake-df**: Aquí se encuentran los desarrollos más antiguos, ya que fue el primero en ser creado y debido a su saturación se generó otro entorno. Contiene procesos de ingesta importantes como los que se encargan de Truck, Snowflake y Chess, por ejemplo.

- **abi-las-dev-uee-integracion2-df**: En este entorno se destinan todos los nuevos desarrollos, con el objetivo de dejar atrás malas prácticas que se encuentran en el anterior. También la organización y el uso de nomenclaturas es importante, y se busca que este entorno lo respete lo máximo posible.

Organización en Azure Data Factory

Azure Data Factory ofrece una interfaz gráfica que permite organizar sus componentes en directorios específicos. Para procesos de ingesta se intenta respetar el siguiente esquema cuando se trata de pipelines:

- `País/SistemaOrigen/Template/Pipeline`
- `País/SistemaOrigen/Pipeline`

Algunos ejemplos:

Organización pipeline template de ingesta

Organización pipeline de ingesta

Cada nombre debe seguir el formato PascalCase.

Nomenclatura de componentes en Azure Data Factory

Cada componente creado en Azure Data Factory debe respetar la siguiente nomenclatura:

- **Pipelines**: `0_[SistemaOrigen]_[Dataset/Template]_PPL` o `[N]_[SistemaOrigen]_[DestinoDl]_[Dataset/Proyecto]_[Dataset/Proyecto/País]_PPL`

- **Datasets**: `[Source/Sink]_[SistemaOrigen/Destino]_[Formato]_DTS`
- **LinkedServices**: `LASCommercial_[SistemaOrigen]_[Conexion]`
- **Triggers**: `A_LASCommercial[País][SistemaOrigen][Dataset/Proyecto]`

Pueden darse casos en los que se requieran nombres más específicos, aunque estos siempre deben respetar lo máximo posible la nomenclatura. Algunos ejemplos:

- **Pipelines**: `0_Truck_Template_PPL`, `0_Truck_PedidosIncrementalAr_PPL`, `0_Sharepoint_Cz_ReportParameters_FileBo_PPL`
- **Dataset**: `Sink_Datalake_Avro_DTS`
- **LinkedService**: `LASCommercial_Databricks_Ingestions`
- **Trigger**: `A_LASCommercialLasSapMadrugada`

El segundo pipeline de ejemplo concatena 'Pedidos' y 'Ar' para hacer referencia a las ingestas de tablas con pedidos desde Truck Argentina. Es importante que los nombres no permitan confusiones, ya que por ejemplo, si el nombre del mismo pipeline no concatenara el país no se sabría a cual de todos se refiere.

Cada nombre debe seguir el formato PascalCase.

Databricks

Databricks es una plataforma para el trabajo con datos que utiliza una forma de procesamiento distribuido. Permite trabajar vía notebooks usando SQL y Python aprovechando toda la capacidad de cómputo que tiene detrás, y evita dedicar tiempo a cuestiones operativas para poder enfocarse en extraer valor de negocio a partir de los datos.

Los distintos ambientes se identifican de la siguiente forma:

- **Desarrollo**: abi-las-dev-uee-Analitycs-dbs.

Ambientes en Databricks

- **Produccion**: abi-las-prd-uee-Analitycs-dbs.

Organización en Databricks

La organización en Databricks es relativa a la función de cada notebook o al conjunto de varias. Sin embargo, siempre se parte de la misma forma:

- `País/SistemaOrigen/`

Algunos ejemplos:

- `/Paraguay/Sharepoint/AnalyticsDataLakeWs/TablasExternas/Cz_Cz_AnalitycsDataLakeWs_TablesPy`
- `/LAS/Templates/PIVOT/Pz_Pz_Config_InsertPivot`

Cada nombre debe seguir el formato PascalCase.

Nomenclatura de notebooks en Databricks

Así como en la organizacion, la nomenclatura es relativa a la función de la notebook y se intenta respetar la siguiente forma:

- `[Origen]_[DestinoDl]_[SistemaOrigen]_[Dataset/Proyecto]`

Cada nombre debe seguir el formato PascalCase.

Creación de ramas

La creación de un rama de trabajo puede realizarse desde cualquiera de las herramientas mencionadas:

- **Github**: Solo es necesario ir a la sección de ramas y presionar el botón para crear una nueva rama.

Crear una rama en Github

- **Databricks**: Desde la configuracion del repo es posible crear una rama en la misma seccion donde se ven los cambios.

Crear una rama desde Databricks

- **Data Factory**: La misma herramienta de integración con Github permite crear una nueva rama.

Crear una rama desde Data Factory

Nomenclatura de ramas

El nombre de cada rama indica su creador y el propósito del desarrollo. Para el caso de desarrollos de ingesta suele seguirse el siguiente formato:

- `usuario/[newdataset/update/bug]-ingesta-[sistemaorigen/proyecto/dataset]`

Esto permite diferenciar si el desarrollo implica la implementación de un nuevo dataset en producción (newdataset); la actualización de alguna configuración, código o documentación de un proceso (update); o la corrección de algún conflicto o inconveniente en un proceso ya implementado (bug).

La utilización del sistema origen, el proyecto o el nombre del dataset puede depender de lo especifico del desarrollo o la existencia de ramas similares en paralelo.

ConfigPivot

El Sharepoint LAS Analytics tiene una sección de Configs, donde se encuentran listas (de Sharepoint, que se ven como tablas) de configuración tanto para desarrollo como para producción.

Los procesos de ingesta requieren de crear un nuevo registro por cada dataset a ingestar, esto debido a que pueden usarse sus valores en los parámetros de los pipelines correspondientes. El caso mas utilizado y sencillo de explicar es el de las querys: los campos 'query' y 'query_reproceso' contienen las consultas que se realizan en los sistemas de origen.

Campos de ConfigPivot

La utilización de los campos en ConfigPivot puede depender del tipo de ingesta y de los pipelines a utilizar. Sin embargo, pueden tomarse como referencia dos casos sencillos que representan una mayoría:

- **Ingestas full**: El campo 'tipo_carga' con valor '1' indica que siempre será tomado el valor en 'query_reproceso', donde se suele almacenar la query correspondiente a la ingesta sin ningún tipo de filtro. Es decir, siempre que quiera ingestarse la totalidad de un dataset puede almacenarse la misma query en 'query' y 'query_reproceso', con el valor '1' en el campo 'tipo_carga'.

- **Ingestas incrementales**: Cuando el campo 'tipo_carga' toma el valor '0' se indica que el valor del campo 'query' es el que debe tomarse. En los casos donde quiera ingestarse un dataset de forma incremental, puede almacenarse una query sin filtros en 'query_reproceso' y otra con la condición necesaria en 'query'. De esta forma puede realizarse una primera ingesta con

el valor '1' en 'tipo_carga', para tener la totalidad del dataset, y luego modificarse a '0' para las demás ingestas.

En los campos 'tabla_nombre', 'schema', 'origen' y 'pais' se busca la coincidencia desde los pipelines de ingesta. Esto quiere decir que siempre, tanto pipeline como registro en ConfigPivot, deben tomar los mismos valores en estos campos/parámetros.

Los campos 'file_dir_cz', 'file_dir_hz' y 'file_dir_pz' pueden contener el path de cada zona en el Data Lake donde se realiza la ingesta. Se debe tener en cuenta que no es posible dinamizar los valores que toman estos campos, por lo que el path de Pz no puede expresarse completo.

Utilización de ConfigPivot

La utilización de los registros en ConfigPivot para el desarrollo de ingestas se da de la siguiente forma:

- Se crea o modifica un registro en ConfigPivotDev.

- Se ejecuta el pipeline 02_Sharepoint_GetConfigPivotList_PPL en abi-las-dev-uee-comercialdatalake-df con 'Dev' como parámetro. Este proceso toma la ultima versión de la lista en Sharepoint y escribe una copia en el Data Lake, por lo que es importante respetar este paso independientemente del ambiente de trabajo.

- La notebook Pz_Pz_Config_selectPivot busca coincidencias entre los parámetros recibidos desde el pipeline de ingesta y los campos del registro escrito en Data Lake, a través del nombre del dataset, el esquema, el sistema origen y el país. Una vez que encuentra el registro correspondiente, devuelve como output en el mismo pipeline de ingesta la totalidad de los campos.

- Cuando las pruebas de ingesta son satisfactorias y es necesaria la implementación, se debe replicar el registro de ConfigPivotDev en ConfigPivotProd.

- También debe ejecutarse el pipeline 02_Sharepoint_GetConfigPivotList_PPL en el ambiente productivo, aunque sumado al paso anterior es tarea de DataOps.

Más casos especiales e información extra

- Se suelen utilizar los nombres comercial e integracion para referirse a abi-las-dev-uee-comercialdatalake-df y abi-las-dev-uee-integracion2-df respectivamente. También son utilizados los términos 1 y 2, aunque en menor medida. ADF funciona como abreviacion para Azure Data Factory y, aunque menos intuitiva tal vez, Db para Databricks.

- Puede haber casos en los que deban realizarse nuevos desarrollos en comercial a pesar de respetar los nuevos criterios, y que incluso se manipulen componentes con vieja nomenclatura. Esto dependerá del costo de migrar el proceso a integracion o de la fragilidad y criticidad del proceso.

- También puede variar la organización en ADF, como por ejemplo para los pipelines de ingesta de archivos .xlsx o .csv que suelen contener el nombre del Sharepoint y algún identificador del proyecto en su directorio. Ejemplo: Bolivia/Sharepoint/AnalyticsDataLakeWs/InventoryDeployment/0_AnalyticsDataLakeWs_Cz_BoliviaInventoryDeployment_Files_PPL.

- En ConfigPivotDev suelen encontrarse registros repetidos para un mismo dataset, e incluso valores en sus campos que no se corresponden a cómo funciona el proceso actualmente. Es por eso que para entender un proceso ya implementado se debe mirar ConfigPivotProd.

- A pesar de existir un consenso sobre el uso del campo correspondiente al tipo de carga en ConfigPivot, hay demasiados registros que utilizan el contrario. No es problema aunque un reproceso implica modificar la única query que tienen, y se presta a confusión el criterio correcto.

- Algunos procesos de ingesta, como el de archivos .xlsx o .csv, utilizan los campos de ConfigPivot de formas específicas que tal vez no se relacionen o hagan sentido con el nombre de los mismos. Esto es un aprovechamiento de la herramienta para ahorrar parámetros en los pipelines de ingesta.

^ Lineamientos para ingestas de inputs manuales

Objetivo

El objetivo del presente documento es indicar los lineamientos necesarios y requeridos para poder solicitar la ingesta de un archivo manual en IRIS Data Lake.

Alcance

Las definiciones y responsabilidades descriptas en este documento aplican a todos los colaboradores de la compañía, propios o terceros, que requieran una ingesta de input manual.

Requisitos

Para la solicitud de ingestas el squad debe hacer una presentación inicial en el comité de A&DG con el fin de validar la propuesta y las necesidades de ingesta de los productos/iniciativas, siguiendo los lineamientos indicados en los documentos/videos de la siguiente carpeta: https://anheuserbuschinbev.sharepoint.com/sites/Analytics-DataLake/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FAnalytics%2DDataLake%2FShared%20Documents%2F0%20%2D%20EDL%20%2D%20A%26GD&viewid=592bba1d%2Dba14%2D4e3f%2D9a94%2D72dc3e475291

Aclaraciones importantes

- Para discovery se debe solicitar al origen una muestra de datos.
- Para que se desarrollen las ingestas de una nueva iniciativa se debe completar el Data Mapping con la metadata de la raw data.
- Para ingestas de inputs manuales se debe adjuntar una muestra de los archivos a ingestar en la tarjeta del comité.

Formato admitido

Los tipos de formatos admitido para ingestar Inputs Manuales son:

- **Formato tabular CSV**: archivo de texto con valores separados por punto y coma (;).

- **Excel (XLSX – XLS):** para este formato de archivo es necesario revisar que no contengan macros, funciones, fórmulas, tablas pivot, componentes gráficos, comentarios de texto en celadas, formatos, filas o columnas ocultas, es decir, únicamente se admite en la hoja de cálculo las columnas y filas que conforman el dataset a ingestar.

Formato de compresión de datos admitidos

Se pueden comprimir los archivos admitidos, usando el formato de compresión Zip. Se debe indicar la compresión anexando la extensión .zip al nombre del archivo. Por ejemplo:

- MyData.zip indica un o varios archivos con formato CSV o Excel, comprimido con ZIP.

Tamaño de archivo máximo

Sin límites.

Repositorio para la ingesta

Los archivos que se soliciten ingestar, deben residir en el siguiente repositorio:

- **Link de Desarrollo**: 02 - Desarrollo

- **Link de Producción**: 01 - Produccion

No se admitirán nuevas solicitudes de ingestas desde FileServer.

Nomenclatura

El nombre del archivo debe respetar el siguiente formato:

- Se admiten snake_case con caracteres alfanuméricos (guiones bajos "_").

- Se sugiere que el nombre del archivo contenga la fecha en caso de que corresponda. Por ejemplo:

    - `Sharepoint/[ambiente]/[país]/[producto-proyecto]/archivo202306.csv`

    - `Sharepoint/[ambiente]/[país]/[producto-proyecto]/archivo_202307.xlsx`

    - `Sharepoint/[ambiente]/[país]/[producto-proyecto]/archivo_202308.xls`

Estructura

A continuación, se detallan los requisitos de estructura que se deben garantizar antes de solicitar la ingesta.

- La primera fila del archivo debe corresponder al encabezado de columnas, es decir, debe contener los nombres de las columnas.

- Los encabezados de columnas no pueden incluir caracteres especiales, ni ser totalmente numéricos. Se admiten snake_case con caracteres alfanuméricos y la separación de palabras por medio de guiones bajos "_".

- Los valores de datos en cada columna deben respetar el tipo de dato que corresponda. Los tipos de datos admitidos son:

    - Texto.

    - Entero.

    - Decimal (punto como delimitador de decimales).

    - Booleano.

    - Fecha:
        - dd/mm/aaaa
        - dd/mm/aaaa hh:mm:ss

- Determinar las PKs del dataset. No se admiten campos PKs con valores nulos y tampoco se pueden repetir los mismos valores para más de una fila.

- El equipo solicitante de la ingesta debe garantizar la calidad de los datos (completitud, formato, consistencia), previo a la solicitud de ingesta.

Metadatos

Los metadatos del dataset deberán quedar registrados en el Data Mapping y debe actualizarse cada vez se requiera una modificación en la ingesta.

Frecuencia de ingesta

Se debe definir la frecuencia con la que se ingestará el archivo solicitado (diaria, semanal o mensual), indicando el horario en el que el archivo se encontrará disponible/ actualizado.

Descripción de dataset y campos

Se debe presentar la descripción funcional del dataset. La misma debe ser lo suficientemente completa y clara para permitir el entendimiento del contenido del dataset por parte de cualquier usuario, tenga o no conocimientos técnicos.

El mismo requerimiento debe considerarse para las descripciones funcionales de cada uno de los campos.

En cuanto a las siguientes columnas del archivo del Data Mapping, las mismas deben ser completadas de acuerdo con la obligatoriedad mencionada a continuación:

- Formato (obligatorio)

- Admite valores Nulos (S/N) (obligatorio)
- Admite vacíos (S/N) (obligatorio)
- Admite valores duplicados (S/N) (obligatorio)
- Valores permitidos (Min-Máx; o valores específicos separados por ";" punto y coma) (obligatorio)

Fuentes no permitidas

A partir del momento de creación y publicación de este documento, no se admitirán "nuevas" ingestas de datos provenientes de bajadas realizadas desde Cognos o BIPROD.

Requerimiento de ingesta

Para solicitar la ingesta se debe generar un ticket en el Board de A&DG **https://dev.azure.com/ab-inbev/LAS_Analytics_Datalake/_boards/board/t/Arquitectura/Stories** con el objetivo de revisar la iniciativa que requiera de la ingesta de un input manual. Es fundamental completar todos los requisitos especificados en este documento.

Para que el requerimiento sea incluido en el backlog del equipo de ingesta, para su posterior desarrollo, la tarjeta debe avanzarse hasta la columna "Ingestion Status".

Otras consideraciones

A continuación, algunas declaraciones respecto a los archivos generados manualmente e ingestados al DataLake:

- Deben ser catalogados por el Data Owner, con apoyo del Technical/Functional Data Referent.

- No serán alcanzados por controles de calidad de Data Governance.

- Deben pasar por el proceso OneTrust, en caso de que contener datos personales o sensibles detallados en el siguiente link: https://iris.ambev.com.br/pt-br/governance/one-trust

- Cualquier cambio de estructura y/o nomenclatura, impactará en los productos que lo consuman, en el catálogo unificado de Data Assets, y en cualquier otro caso de uso de negocio que consuma el dataset. Por lo tanto, no deben producirse tales cambios sin antes cumplir con todos los pasos para la productivización de los mismos.

- El Data Owner o el Technical/Functional Data Referent son los responsables de cualquier problema de calidad que se presente al momento de la ingesta, su solución y las demoras que ocasione este tipo de problemas de calidad en el proceso de ingestas.

- Aquellos requerimientos de ingesta que presenten un problema durante el período de desarrollo, a causa de inconvenientes que deba resolver el squad o el Data Owner, tendrán 5 días hábiles para solucionarlo sin impactar en la priorización del desarrollo de ingesta.

Directivas de retención

Los archivos en los repositorios de Sharepoint se depurarán una vez que estos sean procesados/ingestados en Data Lake.

Control de cambios de los liniamientos

Versión	Fecha de Vigencia	Modificación	Responsable de confección	Aprobador
00	01/09/2023	Creación del documento	Daniela Bertoia	
01	15/09/2023	Se actualiza documento	Daniela Bertoia	
02	09/10/2023	Revisión y ajustes de documentación	Santiago Pita	Gabriel Soler

Ingestas de archivos .csv y .xlsx desde Sharepoint - Inputs manuales

Lineamientos

Antes de comenzar con el desarrollo de cualquier ingesta es importante leer y aplicar los lineamientos definidos para este caso en particular.

Pipelines de ingesta

Para realizar la ingesta de archivos .xlsx y .csv debe utilizarse alguno de los siguientes pipelines según corresponda:

- 0_SharePointExcelFiles_Template_PPL
- 0_SharePointCsvFiles_Template_PPL

Ambos pipelines se encuentran en abi-las-dev-uee-integracion2-df y los desarrollos deben ser en el mismo, ya que es el criterio actual para este tipo de ingestas.

Documentación de 0_SharePointCsvFiles_Template_PPL

Documentación de 0_SharePointExcelFiles_Template_PPL