**FORMATO DISEÑO Y DESARROLLO**

1. **INTRODUCCIÓN**
   1. **Propósito del sistema**

Implementar un flujo de procesamiento y disponibilización de datos, para el flujo de trabajo por lotes de múltiples fuentes.

* 1. **Alcance**

La solución propuesta está compuesta por varias historias de usuario, que comprenden el tratado de la información, el procesamiento y la depuración de los datos entregados de fuentes ARMDEF:

* **5521 HU - Configurar punto de servicio:** Historia por la cual se configura todo lo respectivo al punto de servicio.
* **5822 HU - Configurar dispositivo medidor eléctrico:** Historia por la cual se configura todo lo respectivo al medidor eléctrico.
* **10693 HU - Asociar/desasociar medidor eléctrico principal a punto de servicio:** Historia por la cual se realiza la asociación de un punto de servicio a un medidor eléctrico, esto por medio de interfaz.
* **13627 HU - Configuración de Sistemas externos:** Historia de usuario por la cual se realiza la configuración de la conexión a sistemas externos de forma que el sistema pueda hacer intercambios de información desde y hacia dichos sistemas.
* **5070 HU - Ajustes de Cortes de Energía**: Historia por la cual se realiza la configuración de llenado de datos en caso de apagón.
* **4731 HU - Completar datos faltantes**: Dependiendo de la configuración registrada en la aplicación, en caso de apagón y perdida de lecturas se debe rellenar aquellos vacíos generados, con nulos o ceros.
* **5191 HU - Configurar variables de perfil:** Historia por la cual se realiza la configuración de las variables de perfil de carga.
* **5193 HU - Configurar variables de registros:** Historia por la cual se realiza la configuración de las variables de registro.
* **5194 HU - Configurar variables de Eventos:** Historia por la cual se realiza la configuración de las variables de eventos.
* **4715 HU - Validar etiqueta de variable**: Dada la configuración de variables, se debe almacenar o no los valores de las lecturas que llegan en dichas variables, según lo marcado en la configuración.
* **5144 HU - Umbrales de validación de lecturas:** Configuración por la cual se define el rango de fechas en pasado y en futuro que se deben procesar.
* **4746 HU - Validar lecturas en pasado y futuro**: Dada la configuración de umbrales, se debe realizar una depuración de las lecturas cargadas que no cumplan el rango de fechas en pasado y futuro.
* **4752 HU - Completar información de lecturas**: Identificar la pertenencia de las lecturas y realizar todo lo requerido para poder completar las lecturas de forma de que puedan pasar por el procesamiento por lotes.
* **14177 HU - Versionar lecturas:** Proceso por el cual se almacena y versionan las lecturas.
* **6207 HU - Ejecutar validación de intervalos:** Historia que describe la forma en que se debe realizar la validación de los intervalos para lecturas de perfil de carga.
  1. **Información general**
* *Project name:*
* *Activity/Bug ID/Functionality:*
* *Source Repository:*
* *EA Path*
* *Toggle Feature Name*
* *Product Analyst:*

1. **DISEÑO DETALLADO**
   1. **Vista de Proceso**

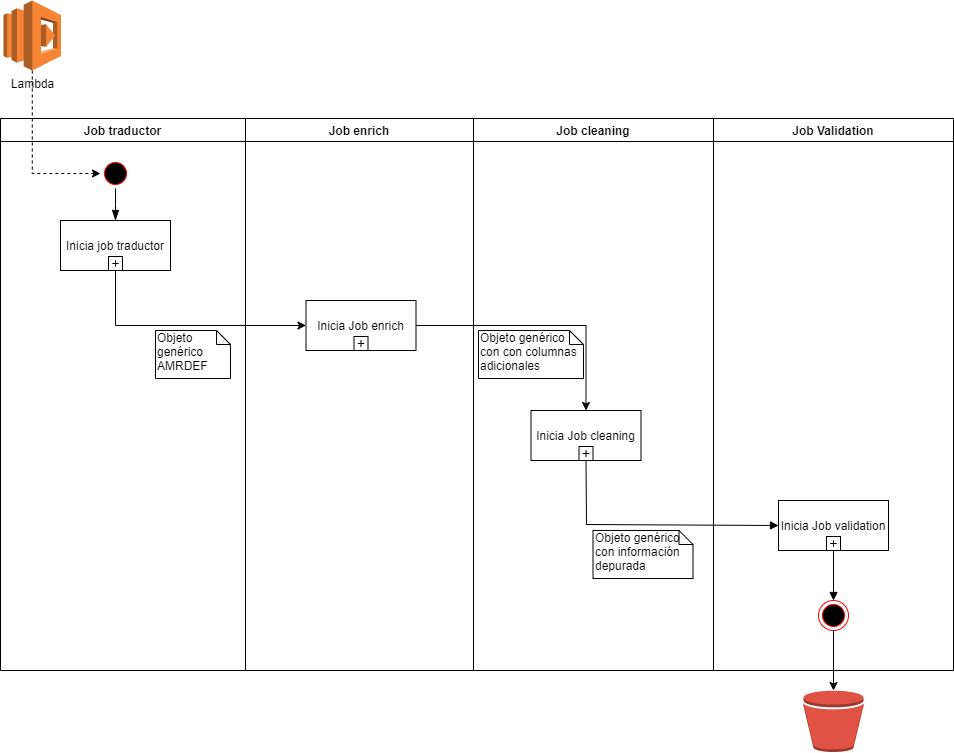


Diagrama 1- Vista general de proceso

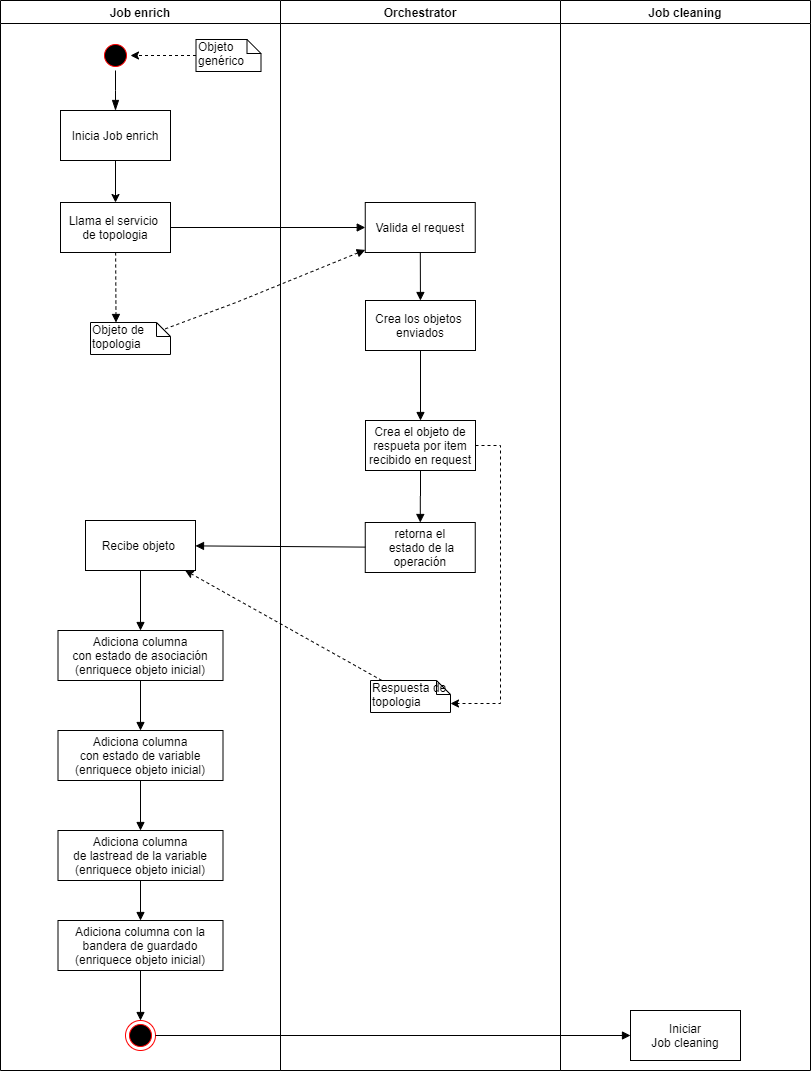
****

Diagrama 2 - Vista de Job Enrich

**Notas:**

* **Inicio Job Enrich:** Se aclara que para este proceso se usara nada más los objetos para la gestión de la topología del dispositivo.
* **Objeto Genérico:** Se debe tener en cuenta que el objeto genérico inicial, es decir el que inicia el proceso de enriquecimiento no cuenta con muchos de los campos mostrados en los ejemplos, sin embargo el proceso de enriquecimiento podrá trabajar con el mínimo posible.
* **Respuesta Topología:** Se debe tener en cuenta que la respuesta entregada por el servicio de topología será la misma que el objeto enviado originalmente, con la única diferencia que se agregan las propiedades bandera para indicar el estado de alguna transacción.
* **Orchestrator:** Se debe tener en cuenta que para hacer las peticiones a este servicio, se debe transformar la información de tabla a JSON, esto para poder hacer una petición valida. De igual forma la información de respuesta será entregada en dicho formato para ser agregada como columnas al objeto genérico inicial.
* **Estampa de tiempo UTC:** Las lecturas contaran con la estampa de tiempo en la que recolecto los datos la fuente. Sin embargo esta estampa es capturada en la hora local, por lo tanto la estampa de tiempo UTC debe ser calculada. Para esto se debe tener en cuenta la zona horaria en la que se encuentra ubicado el medidor, dicha configuración será retornada junto con el proceso de enriquecimiento. <https://earthsky.org/astronomy-essentials/universal-time>

**Medidor ubicado en Bogota – Colombia (UTC-05)**

**Hora de lectura local: 10:00**

**Hora UTC: 15:00 (03:00PM)**

**Calculo: 10:00 + 5**

**Medidor ubicado en Buenos aires – Argentina (UTC-03)**

**Hora de lectura local: 12:00**

**Hora UTC: 15:00 (03:00PM)**

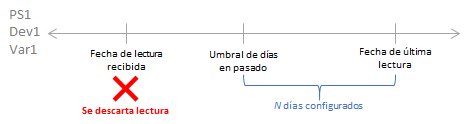
**Calculo: 12:00 + 3**



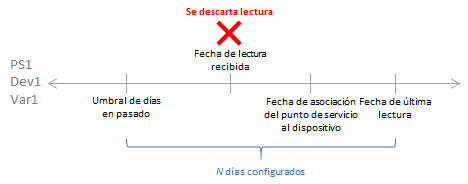
Diagrama 3 - Vista de JobClean

**Notas:**

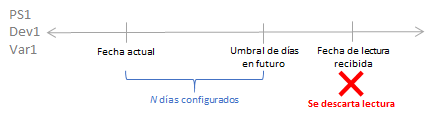
* **Descartar por pasado futuro:** Para descartar las lecturas se cuentan los días hacia atrás configurados. En caso de futuro se toman desde la fecha actual para adelante. La configuración de los días para realizar este descarte serán retornados desde el servicio de plataforma get\_configuration. Se debe tener en cuenta los siguientes caso para el descartado de lecturas por pasado futuro:
  + Se reciban lecturas con fecha menor a la fecha de última lectura menos los días configurados entonces el sistema deberá descartar dichas lecturas.



* + Se reciban lecturas con fecha menor a dicha fecha de asociación entonces el sistema no evaluará los umbrales de almacenamiento en pasado sino que las descartará automáticamente.



* + Se reciban lecturas con fecha mayor a la fecha actual más los días configurados entonces el sistema deberá descartar dichas lecturas.



**Nota:** La información de última lectura y fecha de asociación son retornadas durante el proceso de enriquecimiento en la sección de topología.

* **Descartar por duplicados en memoria:** Si en las lecturas se encuentras dos lecturas para la misma variable, estampa de tiempo y log pero con diferente valor, se debe descartar por inconsistencia.
* **Consulta para recuperar lecturas activas en TrustData:** Se debe buscar la menor fecha de lectura en el bloque de memoria.
* **Rellenar lecturas de LP:** Se debe rellenar con ceros o nulos dependiendo de la configuración después de encontrar la bandera de apagón.
* **Finalización de proceso:** Al finalizar el proceso se deben remover las columnas banderas creadas por el enriquecimiento, es decir se debe finalizar con el objeto genérico original de topología con lecturas depuradas.
* **Versionar LP:** Cuando la lectura es nueva se carga con versión 1, si la lectura coincide con algún registro de **TrustData** pero con diferente valor de lectura entonces es marcado con el valor del siguiente número consecutivo. Se copia los valores de estimado y editado en cada nueva versión. Para realizar el versionamiento de las lecturas se debe tener en cuenta alguna los siguientes criterios:
  + En caso de ser una lectura nueva se crea la versión "Original" que cuenta con las estampas de tiempo tal y como se obtienen de las fuentes de datos.
  + En caso de que existan las lecturas sin embargo las magnitudes sean diferentes, se crea la versión “Original n+1”, siendo “Original n”, la más reciente.
  + En caso que se encuentren lecturas nuevas y existentes para variables tipo perfil o registro, el sistema obtiene la versión "Usage" a partir de la versión "Original" más reciente.

Magnitud versión "Usage" = Magnitud versión "Original" \* Factores

* + En caso que se encuentren lecturas nuevas y existentes para variables tipo evento, el sistema obtiene la versión "Usage" igual a la "Original" más reciente.

Usage = Original

* **Relleno de ceros y nulos**: Se debe tener en cuenta que el servicio que obtiene la configuración de ceros y nulos, también retornara la configuración del rellenado correspondiente para varias banderas, que deben ser comparadas con el campo del objeto genérico llamado **validationFlags,** este campo contiene la información de los QualityCode en formato JSON. Se retorna toda la tabla de configuración de apagón (PowerOutage) para su comparación en cada uno de los registros, ya que los QualityCode pueden variar entre cada una de las lecturas, esta configuración será retornada por el servicio get\_configuration.
* **Logs:**
  + Cuando en la configuración de la variable a la que pertenecen las lecturas, no se encuentre marcada para "Almacenar" entonces el sistema descartará las lecturas asociadas. Las lecturas descartadas deben ser registradas en el log (ReadingsProcessing), donde se registrara el id de las variables que se descartaron.
  + En caso de ser descartadas por la configuración de pasado futuro la entrada en el log (ReadingsProcessing) debe cumplir la siguiente estructura:
    - Id de la variable que se descartó.
    - La cantidad de lecturas descartadas.
    - Motivo por el que se descartó.



* Diagrama 2 - Vista de Job validation

**Notas:**

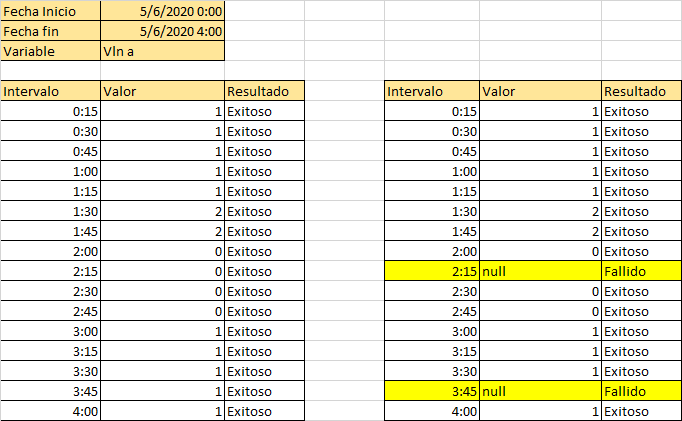
* **Valida datos cantidad de datos descargados contra la cantidad estimada:** Se debe tener en cuenta que en esta acción se agrupa únicamente por variables, es decir que la validación de estampas de tiempo faltantes se hace por cada variable que se encuentre en las lecturas. También se debe tener en cuenta que las operaciones siguientes se realizan por variable hasta que se envié el resultado de la operación. El cálculo de esta operación es de la siguiente forma:

𝑁° 𝑑𝑒 𝐷𝑎𝑡𝑜𝑠 esperados = 𝑁° 𝑑𝑒 lecturas 𝑝𝑜𝑟 ℎ𝑜𝑟𝑎 ∗ 𝑁° 𝑑𝑒 ℎ𝑜𝑟𝑎𝑠

𝑁° 𝑑𝑒 ℎ𝑜𝑟𝑎𝑠  = 31 horas

𝑁° 𝑑𝑒 𝐷𝑎𝑡𝑜𝑠 = 4 ∗ 31 = 124 datos

* En este proceso solo se usara el nodo de lecturas del objeto genérico.
* Es importante mencionar que un bloque de lecturas puede o no estar completo, en caso de no estarlo se debe identificar en donde se encuentra el vacío para marcarlo como lectura fallida.



En el ejemplo anterior la tabla de la izquierda representa un paquete de lecturas donde todas son correctas, es decir que se marcaran como exitosas. En la tabla de la derecha se ve que existen dos vacíos marcados de color amarillo, en este caso se crearan los intervalos faltantes con valores nulos y serán marcados con estado fallido, los demás se marcaran en estado exitoso.

* Las lecturas serán marcadas en el campo **usageValid**, con true o false dependiendo el resultado de la operación. En caso de ser fallido, adicionalmente se debe llenar el campo **ValidationDetail** especificando el fallo, con el siguiente formato JSON:

{

"Usage":

{

"IntervalsError":"Interval not exist"

}

}

* La información de lecturas será creada por defecto como no agrupadas, es decir que serán guardadas en el TrustData con el campo agrupado como false.
  1. **Vista de Lógica**

Se adjunta el diagrama general con navegación para entender el flujo base y los objetos usados dentro de cada una de las operaciones:



* 1. **Vista de despliegue**

N.A.

* 1. **Vista de Componentes**

N.A.

* 1. **Vista de Información**

Objeto genérico de topología en formato JSON, tendrá la siguiente estructura:



En caso de no tener los datos para llenar todas las propiedades del objeto de topología, se podrá especificar únicamente los campos obligatorios para el funcionamiento del proceso, dichos campos se encuentran en el siguiente ejemplo:



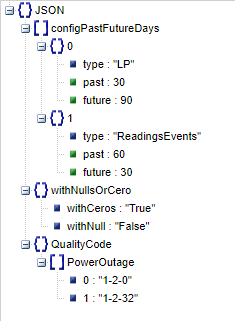
En caso de ser tabular cumplirá con la siguiente estructura, en el siguiente archivo encontrara la estructura tabular con formato enriquecido nombrado en el enrich:



La lista retornada desde los servicios de topología contara con una estructura similar a esta:



El objeto retornado por plataforma en la operación de Clean, es un objeto de configuración que contendrá todo lo respectivo ha pasado, futuro, nulos y ceros. La estructura de dicha respuesta será de la siguiente forma.



Se adjunta también los criterios de aceptación de la historia de usuario 4752 – Completar información de lecturas.



La definición de los logs junto con los campos que serán diligenciados para estos y los pasos en donde deben ser llenados, se especifica en el siguiente documento:}



1. **Decisiones de diseño**

* Se realiza una implementación en donde se separa cada proceso en Traductor, Enrich y Cleaning.
* En cada uno de los procesos se realizan las tareas pertinentes para cumplir su objetivo.
* La comunicación entre procesos se hará por medio de objetos genéricos organizados en columnas.
* La comunicación de las operaciones internas de cada proceso puede variar dependiendo de la necesidad, es decir que puede ser necesario consumir un endpoint que implicaría formatear la información en objetos JSON.