**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.
  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**

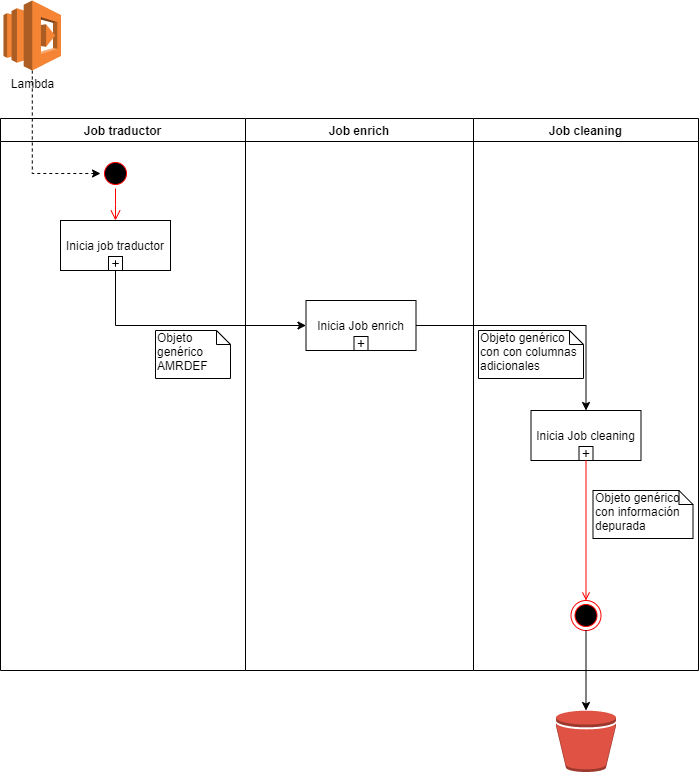
****

Diagrama 1- Vista general de proceso

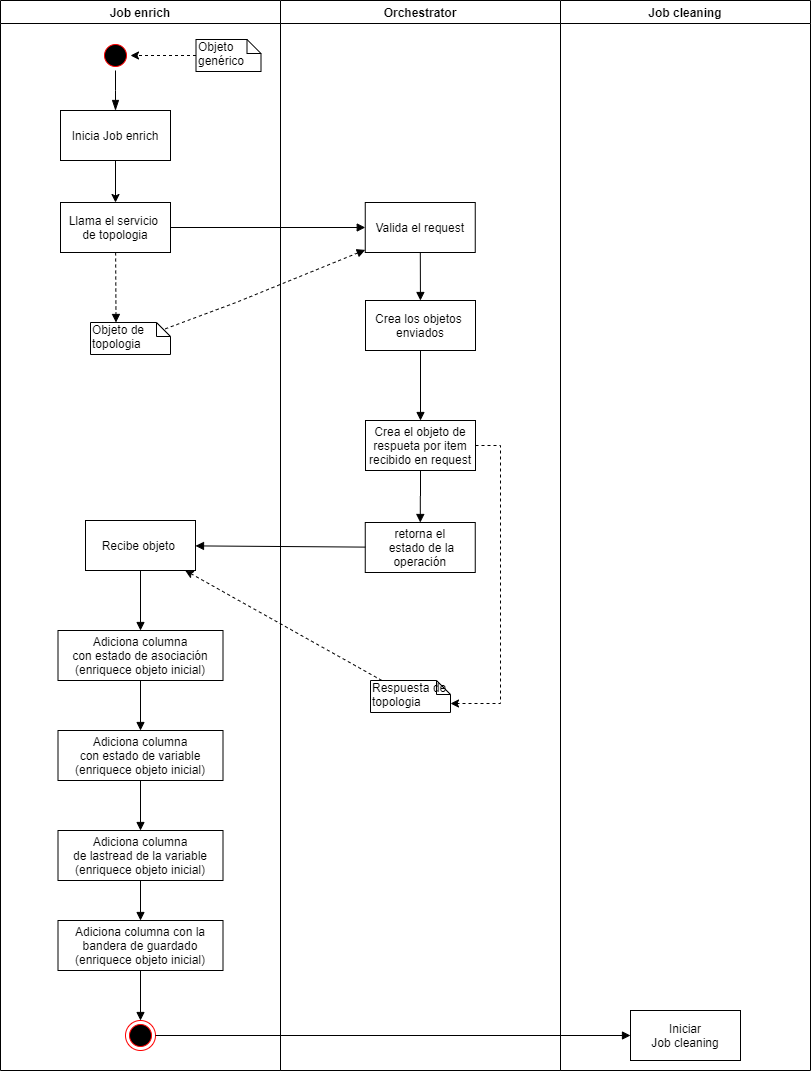
****

Diagrama 2 - Vista de Job Enrich

**Notas:**

* **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.

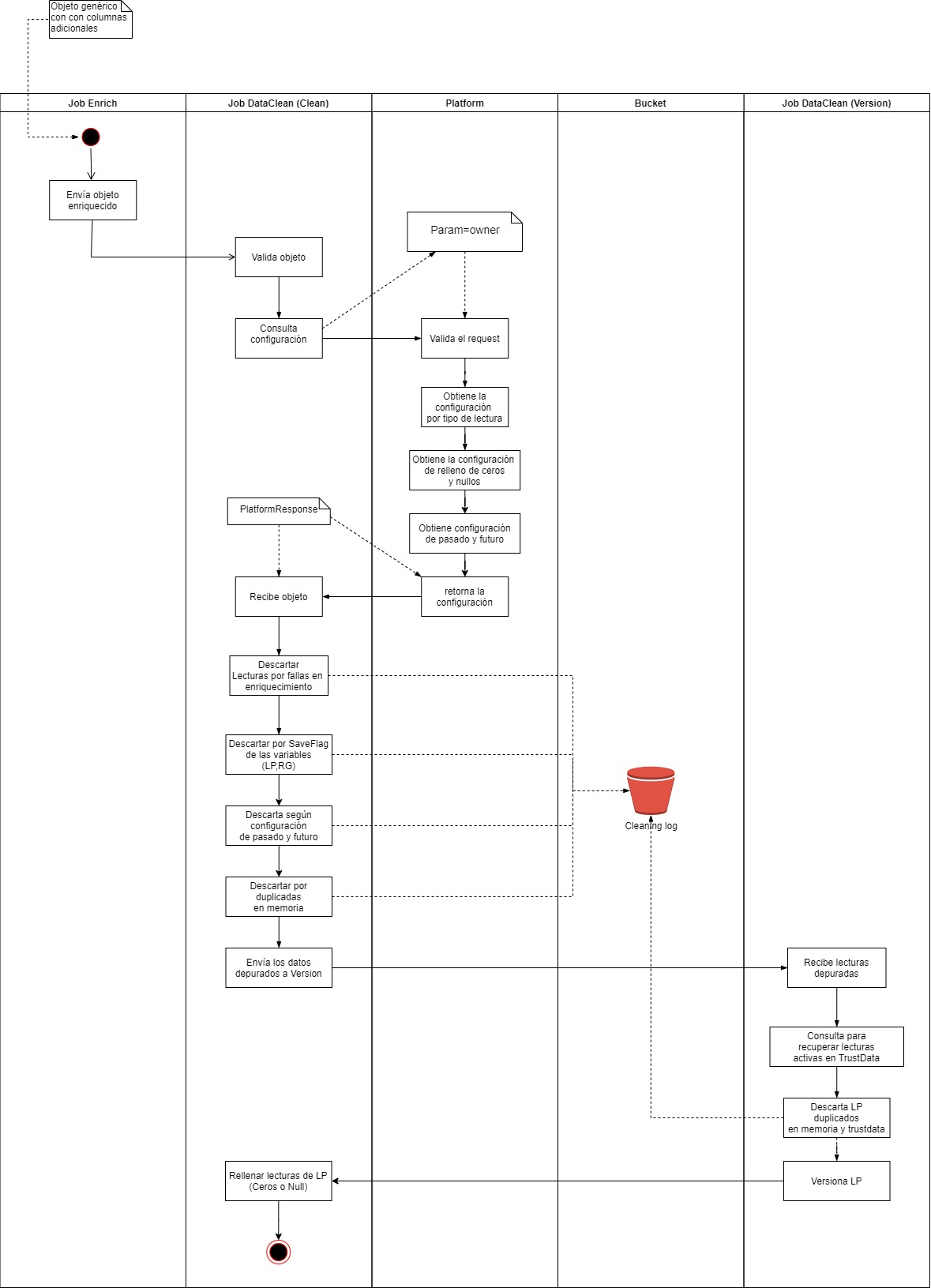


Diagrama 3 - Vista de JobClean

**Notas:**

* **Descartar por pasado futuro:** Para descartar se cuentan los días hacia atrás configurados a partir de la fecha de última lectura. En caso de futuro se toman desde la fecha actual para adelante.
* **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.
* **Versiona 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.
* **Rellenar lecturas de LP:** Se debe rellenar con ceros o nulos dependiendo de la configuración después de encontrar la bandera de apagón.
  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 en formato JSON, tendrá la siguiente estructura:



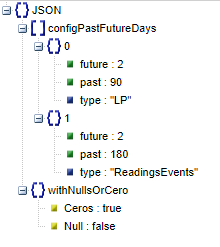
En caso de no tener los datos para llenar todas las propiedades del objeto, se podrá especificar únicamente los campos obligatorios para el funcionamiento del proceso, dichos campos se encuentran en el siguiente ejemplo:



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:



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.



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



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.