Prueba técnica – Ingeniero de Datos

Presentado por Carlos Dubón Cornejo.

[Generalidades: 1](#_Toc196443110)

[Parte 1 – Trabajo de Consultas 1](#_Toc196443111)

[Enunciado 1](#_Toc196443112)

[Consideraciones, hallazgos y supuestos. 2](#_Toc196443113)

[Solución 2](#_Toc196443114)

[Parte 2 - Extracción de datos haciendo uso de APIs 2](#_Toc196443115)

[Enunciado 2](#_Toc196443116)

[Solución 3](#_Toc196443117)

[Parte 3 - Preguntas adicionales 3](#_Toc196443118)

# Generalidades:

* Plantea tus respuestas en base a tu experiencia / conocimientos.
* Puedes establecer los supuestos que consideres para completar la prueba.
* Se presume que, en tu rol, tienes la autonomía para tomar decisiones sobre que herramientas utilizar.
* Prepara una presentación / archivos que contenga las respuestas de las secciones detalladas.

# Parte 1 – Trabajo de Consultas

## Enunciado

Encontrarás adjunto 2 archivos csv correspondientes a una las aplicaciones de créditos y el historial crediticio de dichos usuarios respectivamente. Se le ha solicitado su apoyo como Ingeniero de Datos para la creación de un dataset que facilite algunas variables relevantes para algunos análisis y entrenamiento de modelos predictivos.

Se necesita que, a la tabla de aplicaciones, se le añadan las siguientes columnas (ver Supuesto 1):

* El Estatus que el usuario tenía 3 meses antes de hacer la aplicación. En caso de no tener estatus disponible, colocar “No disponible”
* El saldo promedio de los 6 meses previos a la fecha de la aplicación (ver Supuesto 2).

Para esta tarea se necesita que cree un proceso que tome las 2 bases como recursos y haciendo uso ya sea de Python o SQL, pueda realizar las uniones y transformaciones necesarias. Finalmente se espera que este proceso funcione para futuros clientes que apliquen.

**Notas:**

* La tabla de aplicaciones está a nivel de usuario, es decir que cada registro corresponde a un usuario único
* La tabla de récord crediticio muestra el estatus y saldos de los clientes a en ciertos rangos de fechas específicas, en este caso son cortes mensuales, pero su proceso debe de funcionar, aunque esta tabla tuviese otros niveles de granularidades (diarias, semanales, etc)

## Consideraciones, hallazgos y supuestos.

### Supuesto 1

Considerando que los requerimientos de los campos a añadir se basan en criterios mensuales, y dado que se necesita que el proceso sea adaptable a niveles semanales o diarios, además de contemplar que en la base de datos existen casos en los que un mismo mes presenta más de un registro (como se ilustra en la imagen adjunta).Tabla

El contenido generado por IA puede ser incorrecto.

Se tomará como supuesto, que se establece como regla de negocio:

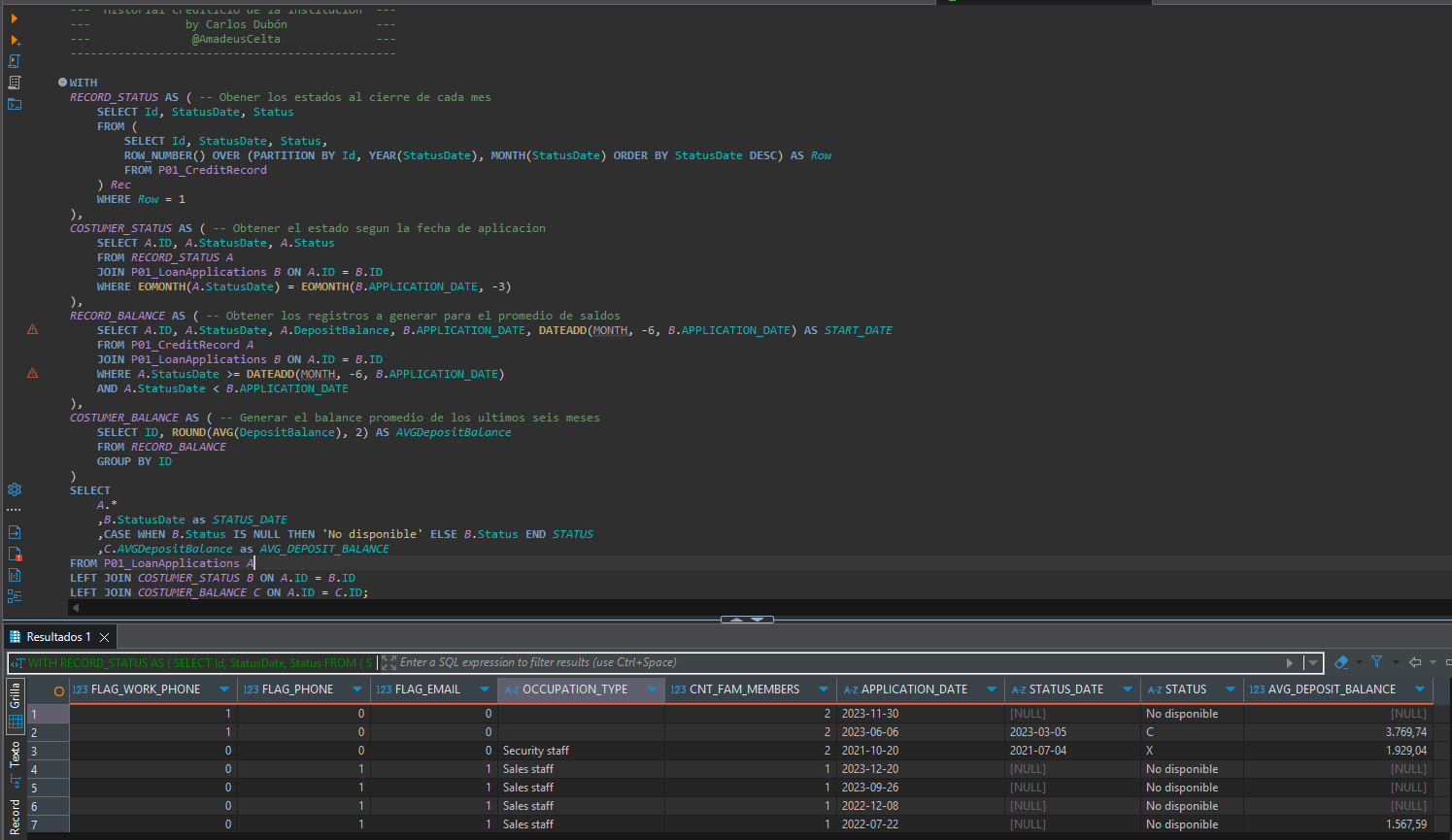
* Para el campo "Status": Se seleccionará el valor correspondiente a la fecha máxima dentro de cada mes. Esto asegurará que, incluso si los datos son tomados a nivel diario o semanal, siempre se utilice el último dato del mes como cierre para los Status.
* Para el cálculo de saldos: Se utilizará el promedio de los registros de los últimos seis meses, sin importar si los datos están registrados a nivel mensual, quincenal, semanal o diario.

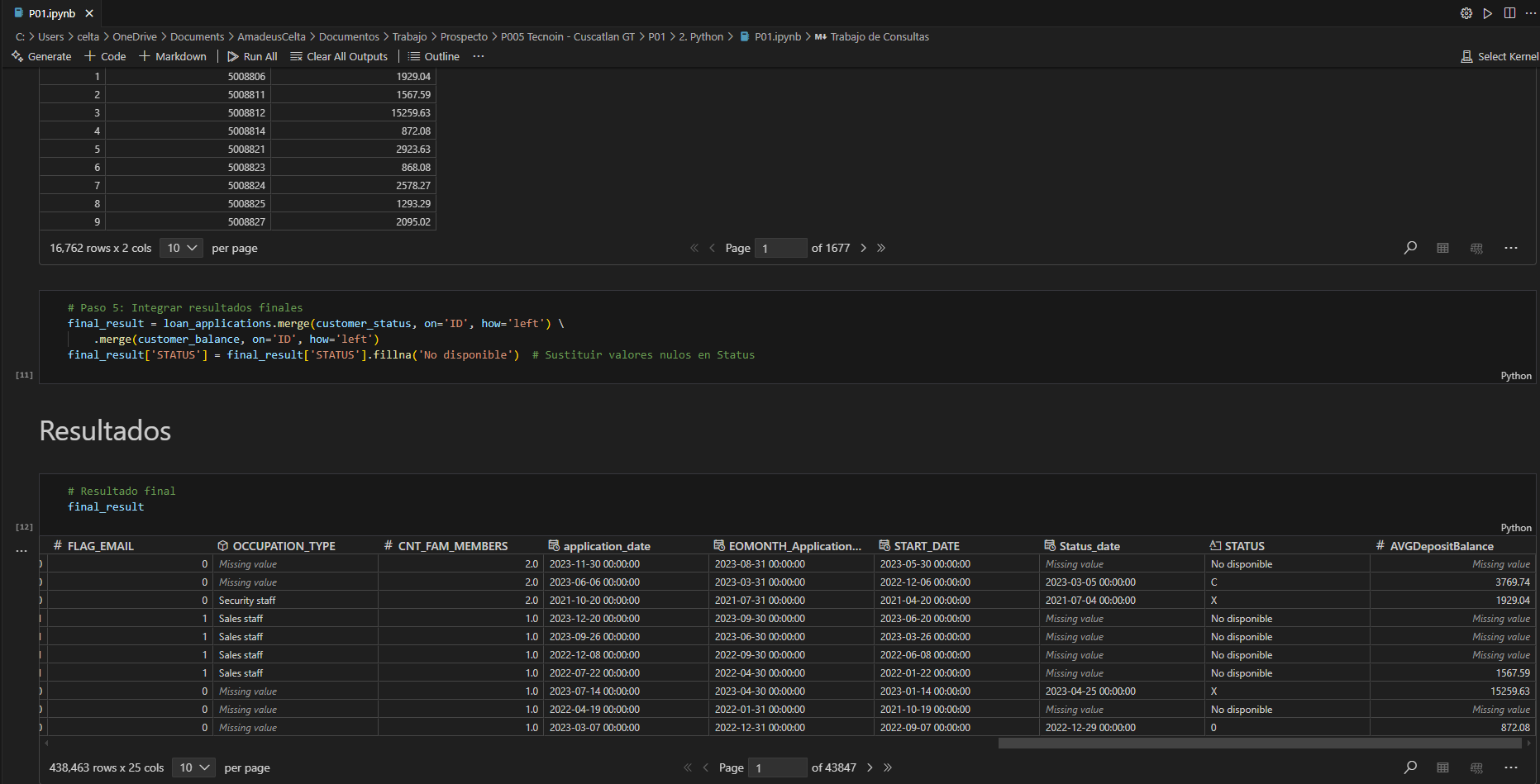
### Supuesto 2

Durante una evaluación, se identificó que, de un total de 438k registros en LoanApplication, solo 16,762 se encuentran en CreditRecord. Para aquellos registros que no cuentan con datos suficientes para calcular el promedio, se establecerá el valor como null.

## Solución

Se realizan dos soluciones:

1. Script en SQL: Se seleccionó SQL Server 2022 como base de datos para el desarrollo. Los datos fueron importados a las tablas correspondientes y se generó un script encargado de construir el dataset solicitado. Debido a la falta de contexto sobre la cantidad de registros que se podrían procesar en un entorno de producción, el diseño se centró en la creación del dataset utilizando CTE's (Common Table Expressions). Como parte de la solución, se entrega un backup de la base de datos junto con el script de consulta SQL.
2. Script en Python: Se desarrolló un script en **Python** que procesa la información directamente desde los archivos CSV. Para este propósito, se utilizó la librería **Pandas**, permitiendo manejar las diferentes consideraciones de manera eficiente. Como parte de la entrega, se incluye un **cuaderno de Jupyter** con el código utilizado y un **PDF** que contiene su contenido documentado.



# Parte 2 - Extracción de datos haciendo uso de APIs

## Enunciado

Usando <https://pokeapi.co/>extraer la información relacionada (especie, base\_experience, abilidades, ataques (moves), entre otros) a los siguientes Pokemons:

1. Bulbasaur
2. Charmander
3. Pidgeotto

Como resultado, cree los siguientes Datasets / Tablas / Dataframes

1. Tabla “pokemon” con los siguientes campos:
   1. Nombre
   2. Id
   3. Base\_experience
   4. Especie
   5. Height
2. Tabla “ataques” (moves) con los siguientes campos:
   1. Pokemon\_id
   2. Nombre del movimiento
3. Tabla “habilidades” con los siguientes campos:
   1. Pokemon\_id
   2. Nombre de habilidad
   3. Esta oculta o no

**Nota:** Su proceso debe de ser flexible para la incorporación de nuevos Pokemons.

## Solución

Para abordar esta situación, se optó por implementar un servidor **MySQL 8.2** y utilizar **Pentaho Data Integration** para generar un job que consuma el servicio correspondiente y aplique las consideraciones necesarias para poblar las tablas destinadas al almacenamiento de información.

**Características de la solución:**

1. **Consumo de servicios:**
   * Se utiliza el servicio **pokeapi.co** para extraer la mayoría de la información.
   * Del servicio **pokeapi.co/v2**, se obtiene la categoría de cada Pokémon.
2. **Control de los Pokémon seleccionados:**
   * La tabla **P02\_PokemonSelected** controla cuáles Pokémon serán consultados.
   * Los Pokémon registrados en esta tabla con estado **1** (Activo) serán consultados y su información almacenada en las respectivas tablas.
   * Si un Pokémon es eliminado de la tabla o se cambia su estado a **0** (Inactivo), se eliminan todos los registros relacionados de las demás tablas.
3. **Tablas de información almacenadas:**
   * **P02\_PokemonsSelected:** Registra los Pokémon seleccionados de los que se obtendrá información.
   * **P02\_Pokemons:** Contiene la información principal de los Pokémon seleccionados (Tabla 1).
   * **P02\_PokemonsMoves:** Detalla los movimientos de cada Pokémon (Tabla 2).
   * **P02\_PokemonsAbilities:** Describe las habilidades de los Pokémon (Tabla 3).
4. **Tablas para requerimientos no funcionales:**
   * **P02\_Parameters: Diseñada para almacenar parámetros del proceso, como el URL del servicio utilizado.**
   * **P02\_PokemonsETLProcess: Registra cada ejecución del job, incluyendo hora de inicio, hora de finalización y el estado (Iniciado, Finalizado, Error).**
   * **P02\_PokemonsETLLog: Documenta la ejecución de cada paso del proceso ETL. En casos de error, solo se registran los pasos completados hasta el momento de la falla.**

# Parte 3 - Preguntas adicionales

Se te pide contestar los puntos detallados a continuación:

1. Que buenas prácticas / consideraciones tomas en los siguientes puntos:
   1. Seguridad y privacidad de la información al manejar datos sensibles

En la medida de lo posible, evitar el uso de estos campos. Al ser utilizado, asegurarse de compartir datos que no puedas ser identificativos del cliente. Asegurar los medios de entrega y que los receptores de los datos/informes posean los privilegios necesarios para obtener la información. Evitar dejar copias de los datos sensibles en archivos en equipos o en tablas temporales.

* 1. cómo implementas gobernanza de datos en tus proyectos.

Identificar a los dueños de los datos. Identificar los usuarios y los niveles de acceso que posee cada interesado y asegurar que cada interesado, de acuerdo con sus niveles de acceso, puedan acceder a la información correspondiente.

* 1. que aplicas para optimizar el tiempo de procesamiento y rendimiento en las consultas a las bases de datos / SQL

Dependerá del contexto. Pueden implementarse, CTE, Subconsultas o uso de tablas temporales y Stagging. Revisar los planes de ejecución. Además de las mejores practicas en la construcción de consultas como lo son, requerir solo los campos necesarios, filtrar de manera de manera efectiva las consultas entre otros.

* 1. Qué utilizarías para manejar datos faltantes en un conjunto de datos.

Siempre es necesario identificar el contexto y en que consisten los datos que se esta trabajando. Iniciaría con un análisis exploratorio para identificar el impacto de los datos nulos, si su impacto, por ejemplo si es un porcentaje mínimo de registros, pueden eliminarse los datos del set utilizado. Si en los campos hay una considerable correlación con el análisis que se esta trabajando, se puede hacer uso de técnicas de sustitución del promedio o mediana. O en el caos de ser datos categóricos, hacer uso de valores por defecto o banderas de datos no disponibles.

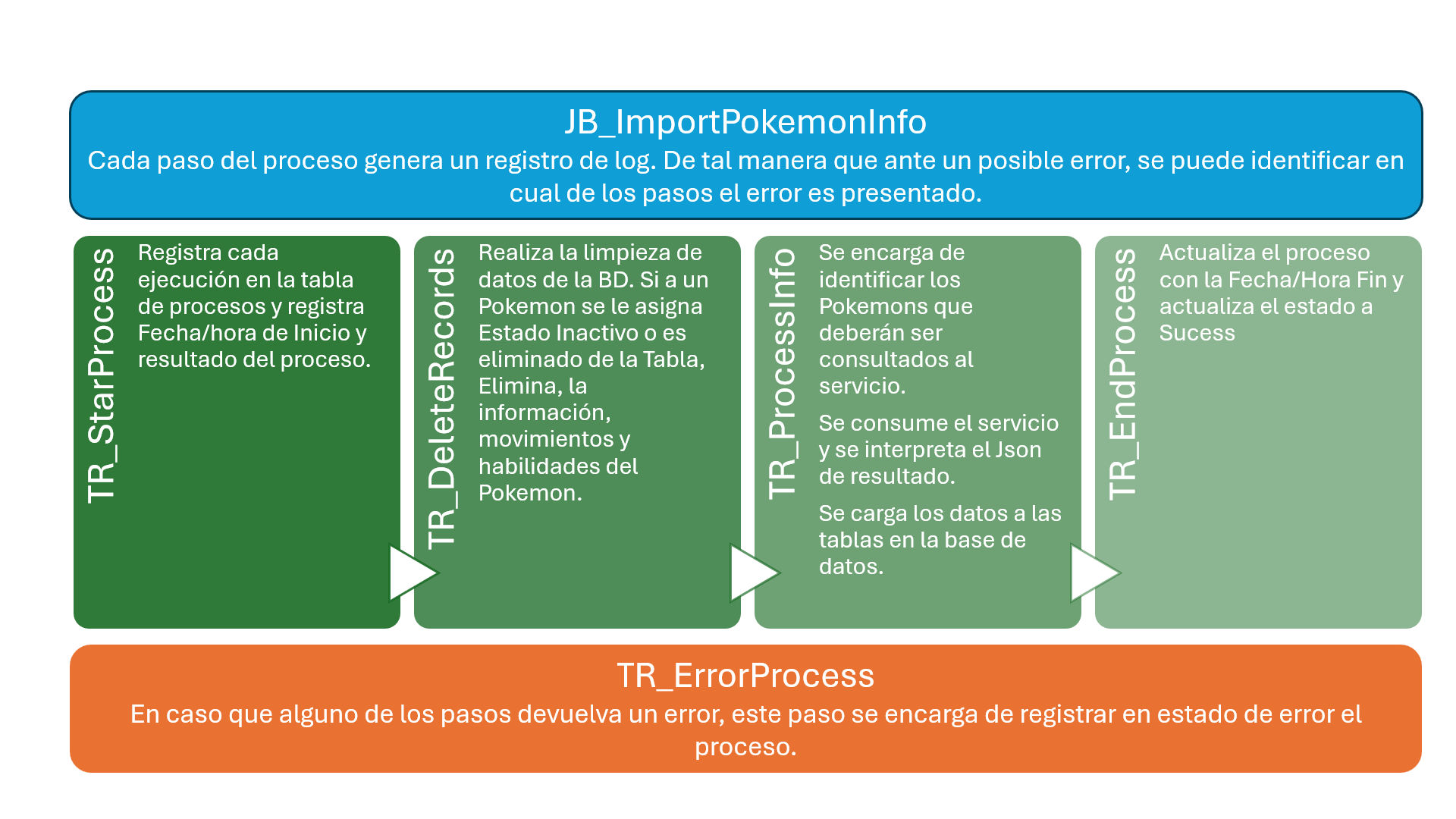
1. ¿Cuál es el flujo de trabajo de ETL que has implementado? Describir o realizar diagrama

En el entendido que se refiere al ETL realizado en la Parte 2. Anexo el flujo general del Job.

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

En el siguiente diagrama se describe el proceso.



El proceso más importante con relación a la recopilación de datos para el proceso es TR\_ProcessInfo. El diagrama general de este paso es el siguiente:

Diagrama

El contenido generado por IA puede ser incorrecto.

El proceso se encarga de:

* + Obtiene la lista de Pokemons Activos y seleccionados para el proceso. Además, obtiene el URL de la API que proporcionará la información.
  + Consume el servicio y obtiene el Json respuesta.
  + Genera tres procesos de lectura de datos.
    - Lectura de información general. Este consume un segundo servicio para obtener la categoría del Pokemon y registra en la tabla correspondiente.
    - Lectura de Moves, Obtiene la lista de todos los moves o ataques de cada Pokemon. Al registrar en la tabla de Moves, se genera todos los registros necesarios para almacenar cada move de cada Pokemon.
    - Lectura de Abilities. Obtiene la lista de todos las Abilities o habilidades de cada Pokemon. Al registrar en la tabla de Abilities, se genera todos los registros necesarios para almacenar cada Ability de cada Pokemon.
  + Espera que los procesos de insert/update sean ejecutados y procede a completar el log del proceso.

1. Se te indica que apoyarás al equipo de Inteligencia artificial, por lo que te consultan también:
   1. ¿Qué consideraciones tomarías cómo ingeniero de datos para armar los conjuntos de datos en los proyectos de IA?
      * Como un punto inicial, considero que se requiere participar activamente en comprender las necesidades del negocio para que la implementación de los modelos y la generación de los datos sea lo más adecuada y efectiva posible.
      * Procedería a evaluar la calidad de los datos, que sean consistentes y evaluar los valores faltantes.
      * Podría ejecutar la limpieza de datos, además de balancear las clases en caso de presentarnos ante problemas de clasificación.
      * Documentación de los procesos, entre otros.
   2. ¿Cómo definirías la calidad de un conjunto de datos para un proyecto de inteligencia artificial?

Principalmente por medio de la integridad, precisión, relevancia y consistencia de los datos. Si se considera que el set de datos cumple por lo menos con estas características, la calidad de los datos será la adecuada.

* 1. ¿Qué herramientas y tecnologías consideras esenciales para el trabajo con datos en proyectos de IA?
     + Herramientas para integración de Datos y procesos ETL, como Pentaho Data Integration o SSIS
     + Bases de Datos, las que estén alineadas con las políticas y estrategias de tecnología de la organización.
     + En cuanto al análisis y procesamiento de datos, considero indispensable Python.
     + librerías como TensorFlow.