



Gestion Operativa

Challenge Técnico - funcional

Autor: **Mariano Javier Palumbo**

Fecha de entrega: **03/06/2022**

Tabla de contenido

1.	Consigna.....	3
2.	Solución Entregada	5
2.1	Consideraciones generales sobre las consignas	5
2.1.1	Definiciones adoptadas.....	5
2.1.2	Fuera de alcance	5
2.2	Secciones del script entregado	5
2.3	Mapeo de Datos.....	7
2.4	Diagrama de flujo de alto nivel.....	7
2.5	Funciones.....	8
2.5.1	Token	8
2.5.1.1	def get_token()	8
2.5.2	Path.....	8
2.5.2.1	def make_relative_path(relative_path, filename)	8
2.5.3	Lectura de Files	8
2.5.3.1	def read_file(relative_path_and_filename)	8
2.5.4	Orquestadores	8
2.5.5	Llamados a APIS.....	8
2.5.6	Parsers	8
2.5.7	Funciones de Fechas.....	9
2.5.7.1	def convert_string_to_date(str_fecha)	9
2.5.7.2	def check_lead_time(fecha_entrega, fecha_prometida)	9
2.5.7.3	def calculate_delay_time(fecha_entrega, fecha_prometida)	9
2.5.8	Generación de archivo .csv	9
2.6	Errores.....	9
2.7	Output.....	10
2.8	Casos de Prueba.....	10

1. Consigna

Amar un script en cualquier lenguaje (preferentemente python), que permita visualizar todas las condiciones de negocio partiendo de una orden de compra.

El resultado del script debe plasmarse en un archivo con extensión .csv

Partiendo de las siguientes órdenes, se necesita saber:

order_id

4114988927

4114988960

4114991684

4114996152

4114997794

4114999549

4115004253

1. Order:

a. ¿Qué productos se compraron?

i. Descripción del producto junto a su variación, si lo tiene.

1. Samsung Galaxy A20 - Color Blanco. - \$4.000
2. Funda Para Galaxy A20. - \$200
3. Cargador Inalámbrico - \$500

2. Payments:

a. Monto total de la orden

b. Moneda utilizada

3. Envío:

a. ¿A qué tipo de logística pertenece?

b. Estado y subestado del envío

c. ¿Cuál es el origen?

i. ¿Salió desde nuestros depósitos o del vendedor? (Si el tipo de Logística es "fulfillment" salió de nuestros depósitos).

d. ¿Cuál es el destino del envío? (analizar en base al campo "agency")

i. Agencia o Domicilio

1. Si fue a agencia, imprimir el ID de la agencia y del carrier
2. Si fue a domicilio, imprimir la dirección completa

e. ¿Qué proveedor lo transportó?

4. Lead Time

a. ¿Qué día se entregó?

b. ¿Cuál fue la promesa de entrega y la fecha límite de entrega?

c. ¿Se entregó en tiempo y forma?

ii. Envíos entregados fuera de la primera ventana estimada.

a. Registrar el delay de entrega, expresado en HH:mm:ss.

Documentación:

1. API's de Mercado Envíos
https://developers.mercadolibre.com.ar/es_ar/envio-de-productos#Introducci%C3%B3n-a-Mercado-Env%C3%ADos
2. Tiempos de Envío:
https://developers.mercadolibre.com.ar/es_ar/envios#Lead-Time
3. Órdenes:
https://developers.mercadolibre.com.ar/es_ar/gestiona-ventas
4. Pagos:
https://developers.mercadolibre.com.ar/es_ar/pagos
5. Tiempos de Envío:
https://developers.mercadolibre.com.ar/es_ar/envios#Lead-Time

Se debe utilizar estructuras de código lo más eficiente posible, entendiendo que todos los recursos consumidos para realizar el análisis mencionado puede afectar los servicios productivos. Con tu cuenta de Mercado Libre, debés generar el access token para poder utilizar los recursos. Las respuestas de cada api call debe ser mockeada. Para ello, deben revisar la documentación de cada API y generar un response simulado para cada caso. Lo ideal sería que el script tenga una variable de entorno que permita ejecutar dev o prod.

La variable de entorno con valor dev devuelve todas las respuestas mock y prod consumiría los recursos de MercadoLibre.

Entregable

Una vez finalizados los ejercicios, lo debes subir a tu repositorio GitHub de manera pública, para que podamos acceder. Compartir el link al finalizar la práctica junto a todo lo que consideres necesario.

2. Solución Entregada

2.1 Consideraciones generales sobre las consignas

2.1.1 Definiciones adoptadas

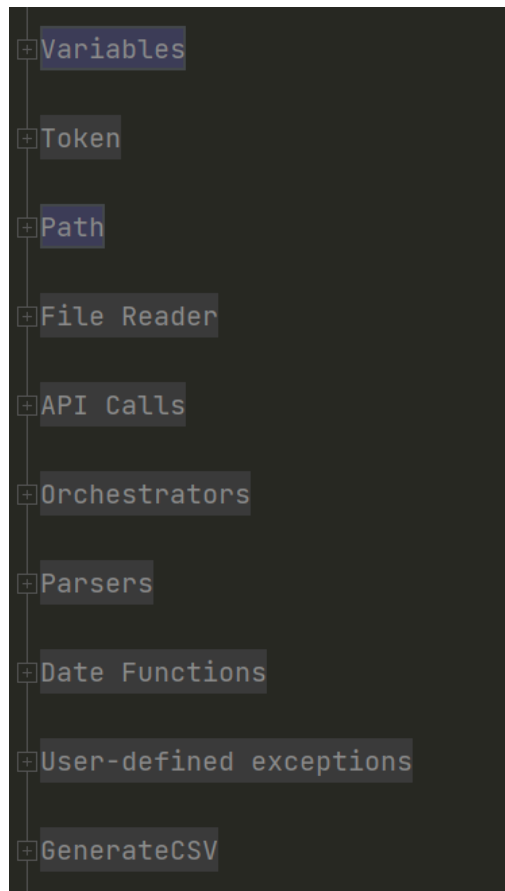
- **Ordenes:** Los id's de órdenes provistos en la consigna no pertenecen a mi usuario y no poseo permisos para consultar las mismas. Por lo tanto, trabajé con mis compras realizadas con anterioridad en la plataforma.
- Para imprimir la variación de los artículos concateno cada elemento de la lista `[variation_attributes]` con una `"/"` por cada elemento de dicha lista.
- En caso de que una orden tenga más de un artículo, se utiliza el mismo criterio del punto anterior. En el ambiente productivo no tengo esta casuística, pero la simulé con el mock en el entorno **DEV**.
- En las operaciones con fecha trabajo con el formato `%Y-%m-%d %H:%M:%S` , por lo tanto **no** se tiene en cuenta la zona horaria (UTC y el offset).
- Para realizar el cálculo de *entrega en tiempo y forma*, u *entrega retrasada* se tiene en cuenta si la fecha de entrega es mayor a la fecha prometida (la fecha límite de entrega **no** la utilizo para los cálculos)
- **Delay de entrega:** se calcula en base a la resta de la **fecha de entrega** y la **fecha prometida**

2.1.2 Fuera de alcance

La presente solución no contempla la eliminación de elementos de la estructura de los json utilizados como mocks.

2.2 Secciones del script entregado

Para organizar el código y agruparlo de acuerdo a una determinada funcionalidad utilicé **Regiones** (así se conocen en C# y los lenguajes de Microsoft). A continuación, se presenta una captura del ide de desarrollo para mayor claridad:



A continuación, una breve explicación de cada una:

Region	Detalle
Variables	Se agrupan variables de rutas de archivos (mocks y datos para generar el token)
Token	Código asociado a la generación del token para utilizar las APIS.
Path	Código utilizado para generar las rutas utilizando el objeto Path, permitiendo compatibilidad entre sistemas operativos Windows y Linux
File Reader	Código utilizado para la lectura de archivos.
API Calls	Código utilizado para realizar las invocaciones a las diferentes APIS.
Orchestrators	Código utilizado para dirigir la petición hacia las APIS o Mocks según el entorno que desee el usuario.
Parsers	Código utilizado para extraer / manipular los datos requeridos en la consigna.
Date Functions	Código utilizado para las conversiones y cálculos de fechas.
User – defined exceptions	Código utilizado para definir excepciones personalizadas
Generate CSV	Código utilizado para generar el output final en un archivo .csv
Main	Ejecución principal del programa. Desde aquí se invoca a las funciones necesarias para satisfacer el requerimiento solicitado.

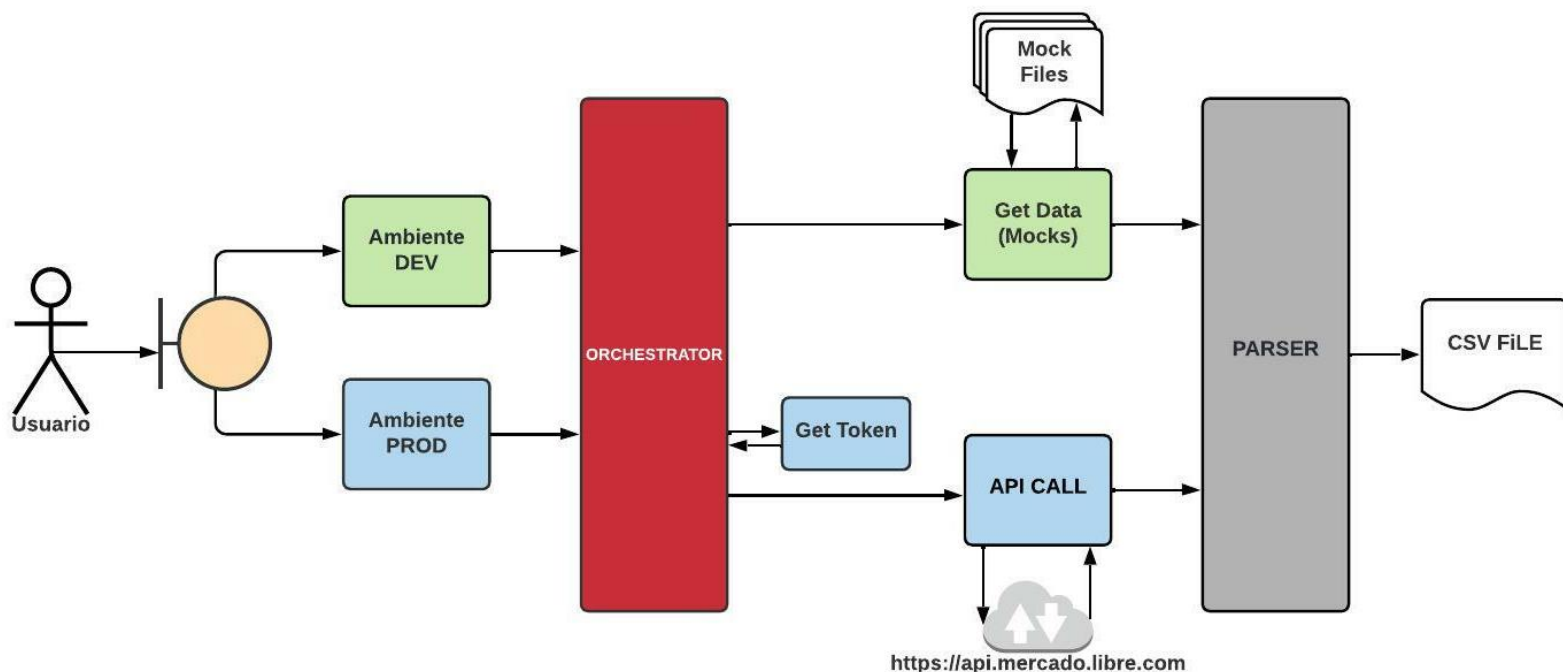
2.3 Mapeo de Datos

A continuación, se presenta un resumen de la procedencia de cada dato solicitado en la consigna.

Dato	API	Tag
Order ID (aux)	https://api.mercadolibre.com/orders/\$ORDER_ID	[id]
Descripcion	https://api.mercadolibre.com/orders/\$ORDER_ID	[order_items][0][item][title]
Variacion	https://api.mercadolibre.com/orders/\$ORDER_ID	[order_items][0][item][variation_attributes]
Monto Total Orden	https://api.mercadolibre.com/orders/\$ORDER_ID	[total_amount]
Monto Total Abonado	https://api.mercadolibre.com/orders/\$ORDER_ID	[paid_amount]
Moneda Utilizada	https://api.mercadolibre.com/orders/\$ORDER_ID	[currency_id]
Shipping ID (aux)	https://api.mercadolibre.com/orders/\$ORDER_ID	[shipping][id]
Tipo Logística	https://api.mercadolibre.com/shipments/SHIPPING_ID	[logistic][type]
Estado y SubEstado del envío	https://api.mercadolibre.com/shipments/SHIPPING_ID	[status] [substatus]
Origen: Deposito propio o del vendedor	https://api.mercadolibre.com/shipments/SHIPPING_ID	Si [logistic][type]=fulfillment deposito propio, sino deposito vendedor
Destino: Agencia (imprimir id agencia y carrier) Domicilio (imprimir direccion completa)	https://api.mercadolibre.com/shipments/SHIPPING_ID	Si [destination][shipping_address][agency][agency_id]distinto de null : [destination][shipping_address][agency][agency_id] [destination][shipping_address][agency][carrier_id] Sin Id agencia imprimir: [destination][shipping_address][address_line]
Que proveedor transporte	https://api.mercadolibre.com/shipments/SHIPPING_ID/carrier	[name]
Que día se entrego	https://api.mercadolibre.com/orders/ORDER_ID/shipments	[status_history][date_delivered]
Promesa de entrega y fecha limite de entrega	https://api.mercadolibre.com/orders/ORDER_ID/shipments	[shipping_option][estimated_delivery_time][date] (Promesa) [shipping_option][estimated_delivery_limit][date] (Fecha Limite de entrega)
Se entrego en tiempo y forma?	No Aplica	Campo calculado
Envios entregados fuera ventana estimada	No Aplica	Campo calculado

2.4 Diagrama de flujo de alto nivel

Para mayor comprensión de cómo funciona la solución planteada, se incluye un esquema a continuación:



2.5 Funciones

2.5.1 Token

2.5.1.1 def get_token()

Función utilizada para generar el token de seguridad y consultar las apis de Mercado Libre.

2.5.2 Path

2.5.2.1 def make_relative_path(relative_path, filename)

Función utilizada para generar las rutas relativas para acceder a los diferentes recursos. Se utiliza el objeto Path para obtener compatibilidad tanto con Windows y Linux.

2.5.3 Lectura de Files

2.5.3.1 def read_file(relative_path_and_filename)

Función utilizada para realizar la lectura de archivos json (tanto mocks como el archivo de recursos para generar el token. Retorna la información en formato json.

2.5.4 Orquestadores

Las siguientes funciones se utilizan para obtener la información de órdenes, envíos y carriers dependiendo del valor de la variable environment. En caso de que sea **PROD** consulta a las APIS, en caso de que sea DEV, busca los archivos mocks ya generados previamente de manera local dentro de la carpeta de recursos.

1. def orchestrator_get_order_data(order_id_for_api, file_id_for_mock, enviroment)
2. def orchestrator_get_shipment_data(shipping_id_for_api, file_id_for_mock, enviroment)
3. def orchestrator_get_lead_time_data(order_id_for_api, file_id_for_mock, enviroment)
4. def orchestrator_get_carrier_data(shipping_id_for_api, file_id_for_mock, enviroment)

2.5.5 Llamados a APIS

Las siguientes funciones se utilizan para realizar las invocaciones a las APIS correspondientes:

1. def api_call_orders_by_order_id(order_id)
2. def api_call_shipments_by_shipping_id(shipping_id)
3. def api_call_orders_lead_time(order_id)
4. def api_call_get_carrier_info_by_shipping_id(shipping_id)

2.5.6 Parsers

Las siguientes funciones se utilizan para dar formato a la respuesta obtenida del orquestador para satisfacer el requerimiento funcional. Esto se hace para las ordenes, envíos y carriers.

1. def order_parser(order)
2. def shipment_parser(shipment)
3. def lead_time_parser(lead_time)
4. def carrier_parser(carrier)

2.5.7 Funciones de Fechas

Funciones utilizadas para conversión de fechas y cálculos de entrega a tiempo y delay. A continuación, se detallan:

2.5.7.1 def convert_string_to_date(str_fecha)

Se utiliza para convertir la fecha que llega en formato string (como respuesta de la API o del mock) para luego poder realizar los cálculos solicitados. Tener en cuenta que se trabaja en formato **YYYY-MM-DD HH:MM:SS**

2.5.7.2 def check_lead_time(fecha_entrega, fecha_prometida)

Se utiliza para saber si el pedido fue entregado a tiempo o retrasado. La función realiza una comparación entre la fecha de entrega y la fecha prometida. Si la fecha de entrega es mayor a la prometida, entonces se entregó con retraso. Caso contrario sería una entrega en tiempo y forma.

2.5.7.3 def calculate_delay_time(fecha_entrega, fecha_prometida)

Por medio de esta función se calcula el tiempo de delay (si existiera) entre la fecha de entrega y la fecha prometida. La diferencia entre ambas se calcula en segundos. Luego se realizan divisiones enteras y módulo para expresar este delay en formato **HH:MM:SS**

2.5.8 Generación de archivo .csv

Por medio de la función **generate_csv_file** se almacena el resultado de todo el procesamiento realizado. La misma genera el archivo **ResultadoEjecucion.csv** delimitado por comas. El mismo se encuentra dentro de la carpeta en que está alojada la solución y cada vez que se ejecuta la misma, el archivo se sobre escribe.

2.6 Errores

Se utilizan **excepciones personalizadas** y **predefinidas** para el tratamiento de los errores que puedan ocurrir durante la ejecución del programa. En todos los casos se imprime un mensaje para tomar la acción correspondiente. A continuación, se detallan los más significativos:

Excepción	Detalle
EnvironmentInvalidException	Se produce cuando el ambiente de ejecución ingresado no es válido. Solo se admiten los valores PROD y DEV
IOError	Por lo general se produce en la escritura del archivo .csv cuando el mismo se encuentra abierto y tomado por el SO. En ese caso se debe cerrar el archivo.
FileNotFoundError	Se produce cuando no se encuentra un archivo determinado (mocks files o archivo de recursos)
Exception	Abarca cualquier error no contemplado anteriormente. Se mostrará un mensaje orientativo.

2.7 Output

El output final del procesamiento será un archivo en formato .csv. Para mayor detalle consultar el punto [2.5.8 Generación de archivo.csv](#)

2.8 Casos de Prueba

A continuación, se adjunta un archivo .csv como resultado de ejecución en el ambiente DEV



CasosDePrueba

Se detalla en cada casuística

Caso	Detalle
01	Producto sin variation attributes , tipo logística= self_service , estado del envío= delivered , origen= depósito vendedor , destino= domicilio completo ya que no tiene id de agencia, entregado en tiempo y forma y no registra delay .
02	Producto sin variation attributes , tipo logística= drop_off , estado del envío= delivered , origen= depósito vendedor , destino= id agencia/id Carrier ya que los datos se encuentran disponibles, entrega con retraso, se calcula delay .
03	Dos productos en la misma orden, con variation attributes , tipo logística= self_service , estado del envío= delivered , origen= depósito vendedor , destino= domicilio completo ya que no tiene id de agencia, no se puede calcular estado de entrega porque le falta una fecha, el delay no se puede calcular porque falta una fecha.
04	Producto con variation attributes , tipo logística= fulfillment , estado del envío= delivered , origen= depósito propio (debido al tipo de entrega), destino= domicilio completo ya que no tiene id de agencia, sin información del transportista, entregado en tiempo y forma y no registra delay .
05	Producto con variation attributes , con varios campos sin datos . Se verifica que el programa no presenta errores.
06	Producto con variation attributes , tipo logística= cross_docking , estado del envío= delivered , origen= depósito vendedor , destino= id agencia/id Carrier ya que los datos se encuentran disponibles, con datos del transportista, entrega retrasada, se calcula delay .