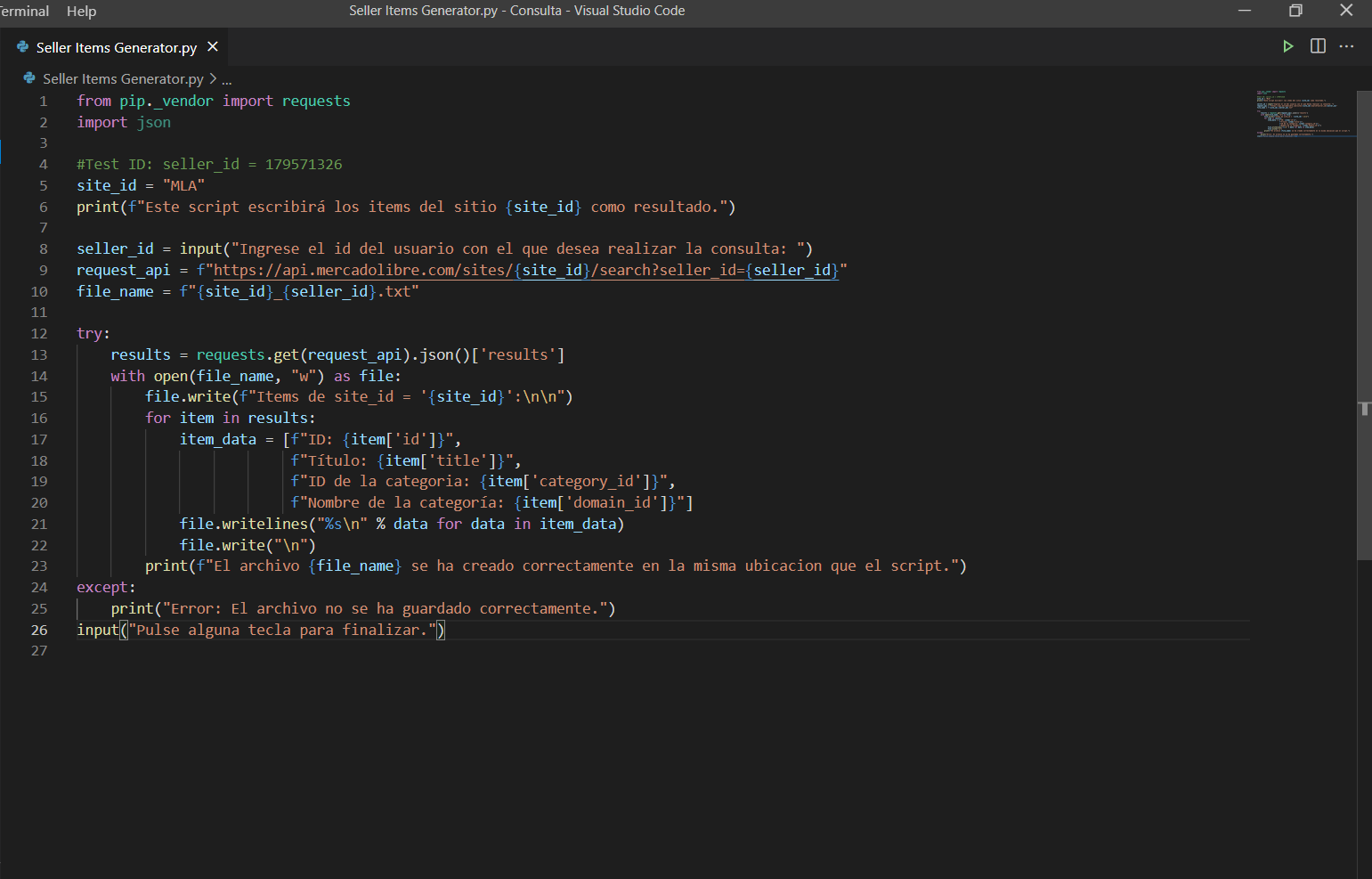
# Ejercicio - Investigación / Script

Cómo parte de nuestras tareas en soporte, tenemos la necesidad de consultar información masivamente. Para poder agilizar estas tareas necesitamos construir un script que nos permita realizar la siguiente tarea:

1. Recorrer todos los ítems publicados por el **seller\_id = 179571326** del

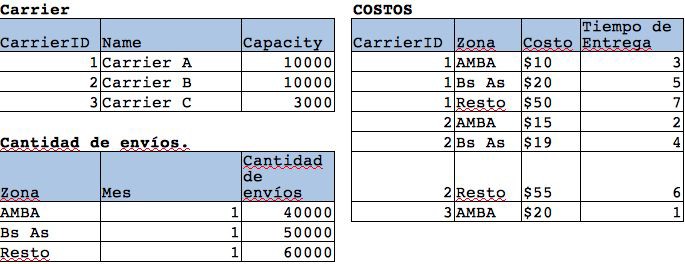
# site\_id = "MLA"

1. Generar un **archivo de LOG** que contenga los siguientes datos de cada ítem:
   1. **"id" del ítem, "title" del item, "category\_id" donde está publicado, "name" de la categoría.**



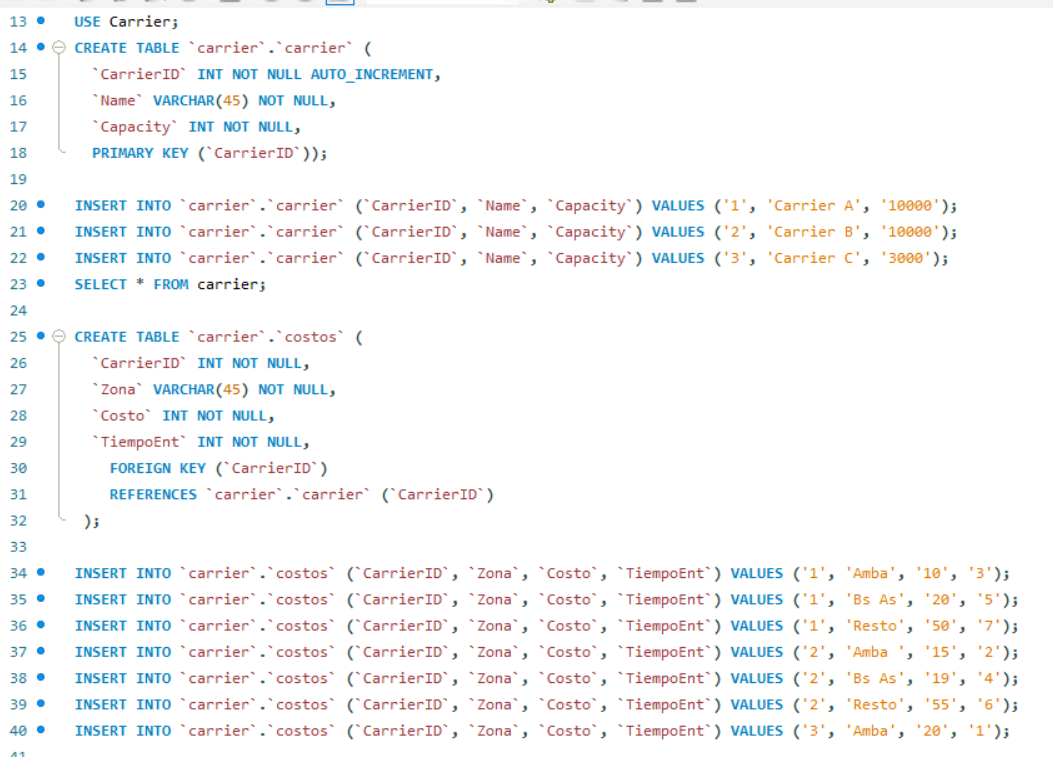
# Ejercicio - SQL + Propuesta

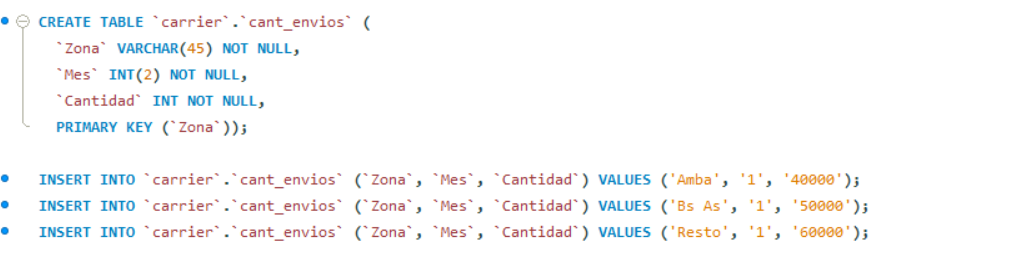
Teniendo las siguientes tablas:

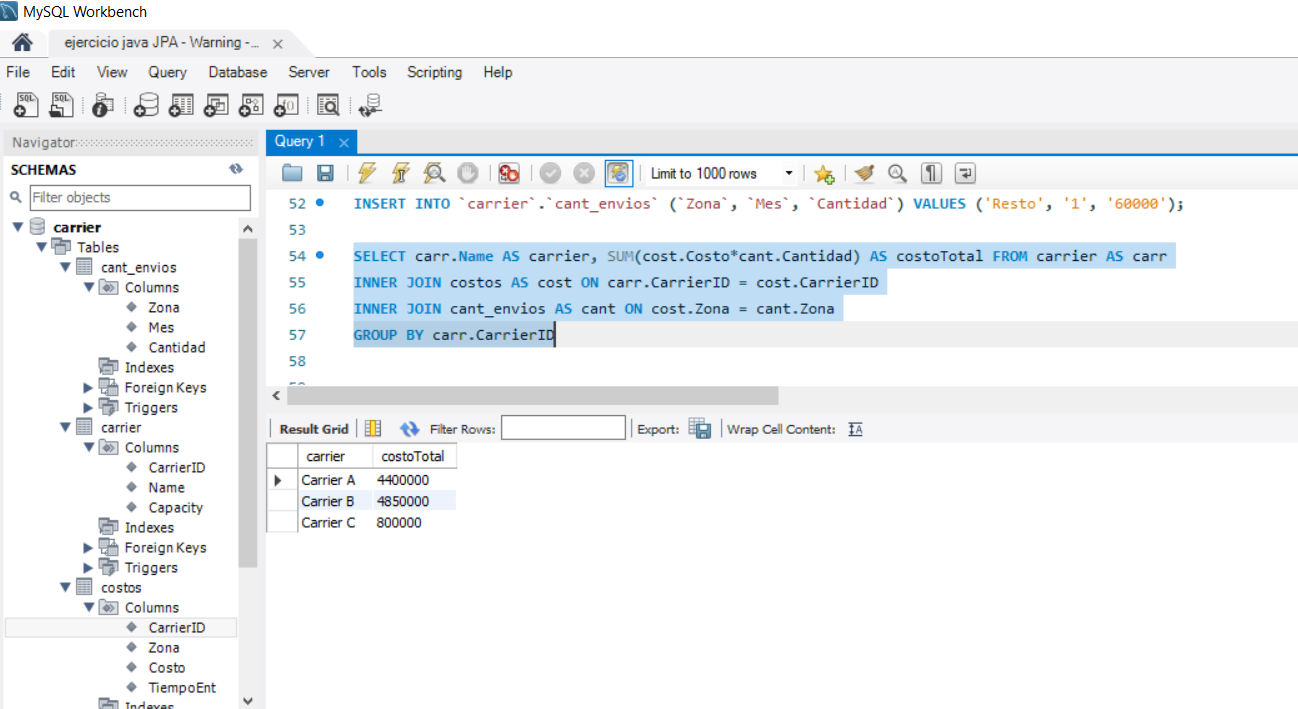


(\*) **carrier.capacity** → Representa la capacidad mensual máxima a transportar

* Obtener para el mes 1 cuánto costaría enviar con cada carrier los envíos de la tabla **Cantidad de envíos**.
* ¿Qué queries realizaste?







* ¿Que propuesta harías para el mes 1 considerando un presupuesto de

$3.000.000?

# En el mes 1 hay 150.000 envíos a realizar y con los carriers disponibles cuento con una capacidad operativa máxima de 23.000 envíos al mes. Dado que no se cuenta con un dato de prioridad hacia alguna de las distintas zonas de los envíos, se asume que las mismas tienen el mismo criterio de importancia.

# Por otra parte, tampoco se cuenta con ninguna especificación que permita aplicar el tiempo de entrega con exactitud, ya que no poseo información de cómo se distribuye la carga máxima de cada carrier por cada salida de reparto (únicamente se cuenta con carga máxima total que puede transportar por mes). Por esto, el presente análisis considera la carga máxima que los carriers transportan mensualmente y de sus respectivos costos la opción más rentable.

# Propuesta 1: Priorizando el criterio de minimizar costos, se realizan los envíos con menor costo de cada carrier correspondientes a la zona amba, resultando en un total de $310.000.

# 

# Como en la propuesta anterior solo se atendió los envíos de zona amba por cuestión costo, se plantea una segunda propuesta en la que se da prioridad atender a todas las zonas.

# 

# Propuesta 2:

# 

# Lectura y comprensión Script básico grails

* A tu entender, que se busca obtener como output del script?

import me.\*;

def upsPullTrkService = ctx.getBean('upsPullTrkService') def s = Shipment.get(27528954729)

def tn = s.trackingNumber

def trackingData = upsPullTrkService.getTrkEvents([tn]) trackingData.each { td ->

println " "

println "${td.sucursal} - ${td.eventDate} - ${td.description}"

}

"Done"

El script busca los datos de un shipment y luego muestra por pantalla los datos sucursal, fecha y descripción.

# Script básico bash

* A tu entender, que se busca obtener como output del script?
* Podrías detallar que se hace en cada línea del script?
* Cuántas líneas se imprimen como output?

#!/bin/bash

users\_id=(71665538 66146765 132961968 15096445 172753273 54152646)

for users\_id in ${users\_id[\*]} do

curl=$(curl -s "[api.mercadolibre.com/users/$users\_id/shipping\_preferences](http://api.internal.ml.com/users/%24users_id/shipping_preferences?caller.scopes=admin)" | jq -c '.services')

echo "$users\_id: $curl"

done

El script busca las preferencias de shipping de los users\_id y lo imprime por pantalla

Línea 1: Se ejecuta el intérprete de comandos en el sistema ubicado en el directorio bin

Línea 2: Se almacenan los user\_id

Línea 3: Se recorre con el for toda la lista de user\_id que se cargaron

Línea 4: Bucle do estructura de control que repite la acción mientras queden usuarios pendientes

Línea 5: Se conecta a una API de mercadolibre para obtener la información de preferencia de compra del usuario

Línea 6: Imprime el user\_id y las preferencias de shipping

Línea 7: Fin de ejecución

En el ejemplo se imprimen 6 líneas dado que se consultan 6 usuarios