# Desafíos para Desarrollador Backend

Utilizar Git como gestor de versiones. Todos los desafíos deben entregarse con el link del repositorio donde se encuentra subido. Cada desafío posee un directorio con sus recursos.

1. **Manejo de estructuras de datos**Desarrollar PHP un script que resuelva el siguiente problema:

*El sector de Networking del área de Sistemas detecta una situación sospechosa en la red y genera un archivo llamado* ***ip\_blacklist.csv*** *que contiene un listado de strings separados por coma correspondientes a una lista de IPs. El segundo octeto de la IP define la subred en la que se encuentra esta IP en la red. Afortunadamente, ya existe un endpoint de API que devuelve un JSON con campos de nombre del nodo y subred.*

*Se necesita cruzar esta información para identificar la cantidad de casos que existen en cada nodo.*

*Crear un script que lea el archivo que contiene la lista de IPs y devolver un archivo JSON llamado* ***cases\_by\_nodes.json*** *respetando el siguiente formato de ejemplo:*

{

"data":[

{

"nodo":"DECIMO",

"casos":9

},

{

"nodo":"HUDSON",

"casos":6

}

]

}

**NOTAS:**

* *Si el nodo no posee casos no debe figurar en el JSON.*
* *Por cuestiones de rendimiento se debe, en lo posible, evitar el uso de bucles en el código. Para ello investigue sobre las funciones nativas que posee PHP para cruzar la información entre arrays.*
* *Como los datos vienen de un archivo generado se debe tener en cuenta validaciones básicas como: que no se encuentre vacío, que su contenido sean IPs válidas, entre otras que ayuden a darle fiabilidad a los resultados.*

1. **Base de datos**Cargar el ***dump*** de base de datos de ejemplo que se brinda en el archivo en el motor de base de datos ***Mysql***. Una vez cargada, usar la base de datos **classicsmodels** para crear consultas basadas en los siguiente criterios:
   1. Crear una consulta que devuelva la cantidad de **customers** que se encuentran en cada ***country***. La respuesta debe contener columnas como: ***states*** y ***quantities***.
   2. Crear una consulta que devuelva la sumatoria de ***amount*** de los **payments** de cada uno de los **customers** de ***country***=’USA’ y ***state***=’Boston’ entre las fechas (***paymentDate***) 01/01/2004 y 01/06/2004. Además, la respuesta debe estar ordenada de forma descendente por ***totalAmount***. La respuesta debe contener columnas como: ***customerNumber***, ***customerName***, ***totalAmount*** y ***creditLimit***.
   3. Crear una consulta que devuelva ítems que pertenecen a la ***orderNumber***=10313. La respuesta debe contener columnas como: ***productCode***, ***quantityOrdered***, ***priceEach***, ***orderLineNumber***, ***productName***, ***productLine*** y ***textDescription***.
   4. Crear una consulta que devuelva un listado de jefes y empleados que le reportan a él. La respuesta debe contener columnas como: ***employeeNumber***, ***firstName*** y ***lastName*** (*en una sola columna*), ***email***, ***jobTitle***, ***officeCode*** y ***subordinate*** (*que contiene una cadena de texto separada por coma de todos los* ***employeeNumber*** *que se encuentran a su cargo por medio del campo* ***reportsTo***).

Para más información ver la siguiente figura que muestra un **Modelo Entidad-Relación** que ayuda resolver estos ítems.

