



## Ejercicio desarrollo y tecnología (Python + REST API + SNMP)

### Introducción

Un cliente nos plantea la necesidad de inventariar los cablemódems (CMs) que tiene en su red, y para ello nos solicita desarrollar una herramienta que le permita obtener información de un cablemódem a partir de su MAC.

Adicionalmente se le suma la problemática de que la información de los cablemódems tiene que ser ingresada en un sistema que sólo soporta importación por base de datos.

Por esta razón se solicita desarrollar una aplicación que dada una determinada dirección MAC de un cablemódem, se obtenga vía el protocolo SNMP la información de fabricante, modelo y versión de software. Estos campos se deben almacenar en el destino dado.

### Condiciones solicitadas

La aplicación presentará una REST API, por lo que se solicita que la MAC se pueda pasar como parámetro de la consulta (ver ejemplos más adelante).

En caso de que un fabricante, modelo y versión de software ya esté almacenado, se tiene que devolver un error.

El endpoint a implementar es:

**`api/device/inventory/cm/<mac>`**

Los parámetros que se ingresarán en la API serán los siguientes:

- mac: MAC del cablemódem.

### Validaciones a considerar

- Se tiene que validar que la MAC respete el formato de una dirección MAC en minúscula, sin separación por ":". Ejemplo de MAC válida: a415881f6f22
- Que el fabricante, modelo y versión de software no esté repetido en la base de datos.



## Otras consideraciones

- El CM puede no tener IP.
- El CM puede no responder a la consulta SNMP, dando timeout.
- Como comunidad SNMP se asume "private" para todos los CM.

## Requerimientos no funcionales

- Se debe utilizar Python 3.4 o superior como lenguaje para desarrollar la aplicación.
- Se asume que todos los cablemódems soportan SNMP v2c.
- Se debe utilizar MySQL, MariaDB o MongoDB como motor de base de datos.
- Se pueden utilizar todas las librerías que considere necesarias, sin restricciones.

## Ejemplo de secuencia de peticiones y respuestas

1.

### Query

POST api/device/inventory/cm/48f7c0900592

### Response 200 OK

```
{
  "errorCode": 0,
  "errorStr": null,
  "result": "success",
  "items": {
    "vendor": "Arris Interactive, L.L.C.",
    "model": "TG862G",
    "softwareVersion": "9.1.103S1"
  }
}
```

Se debe almacenar la información de fabricante, modelo y versión de software en la base de datos.

2.

### Query

POST api/device/inventory/cm/48f7c0900592

### Response 200 OK

```
{
  "errorCode": 409,
  "errorStr": "Cable Modem info already stored",
  "result": "failure",
}
```



```
"items": {}  
}
```

Se debe responder con un error, ya que la información de este CM ya se encuentra almacenada (se almacenó en el punto 1).

3.

#### Query

POST api/device/inventory/cm/f46befcc3d68

#### Response 200 OK

```
{  
  "errorCode": 404,  
  "errorStr": "Cable Modem not found",  
  "result": "failure",  
  "items": {}  
}
```

Se debe responder con un error, ya que no existía en DB una IP lease para esta MAC.

4.

#### Query

POST api/device/inventory/cm/46f2f091ed2c

#### Response 200 OK

```
{  
  "errorCode": 408,  
  "errorStr": "SNMP timeout",  
  "result": "failure",  
  "items": {}  
}
```

Se debe responder con un error, ya que para la IP asignada al CM no se obtuvo respuesta SNMP.

## Información adicional

## Almacenamiento en base de datos

La tabla/collection para almacenar la información de los cablemódems se tiene que llamar **cm\_models** y tiene que tener la siguiente información:

- vendor: VARCHAR
- model: VARCHAR
- software\_version: VARCHAR



Las direcciones IP de los CMs se encuentran ya almacenadas en la tabla/collection llamada **leases**, con los siguientes campos:

- mac: VARCHAR
- ip: VARCHAR

## SNMP

El protocolo SNMP es un protocolo simple para gestionar dispositivos de red. Los cablemódems suelen presentar la capacidad de gestión de algunas de sus funcionalidades mediante este protocolo.

La OID SNMP sugerida para consultar los datos del cablemódem es:

Name: sysDescr  
OID: 1.3.6.1.2.1.1.1

La misma siempre devuelve un string con la descripción del dispositivo, e incluye los valores de VENDOR, MODEL y SW\_REV siguiendo el formato que se apreciará en el ejemplo a continuación.

### Ejemplo

Suponemos que el CM tiene la IP: 127.0.0.1.

Para hacer una consulta SNMP al puerto 1024:

```
snmpget -v2c -c private 127.0.0.1:1024 1.3.6.1.2.1.1.1.0
```

Respuesta:

```
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco DPC2325 DOCSIS 2.0 2-PORT Gateway  
<<HW_REV: 1.0; VENDOR: Cisco; BOOTR: 2.1.71R3; SW_REV:  
dpc2325-v202r12811-110715as; MODEL: DPC2325>>"
```

### Simulador SNMP

Puede probar la interacción SNMP con un cablemódem haciendo uso del simulador SNMP snmpsim, disponible en PIP:

[Quick start — SNMP Simulator 0.4 documentation](#)

NOTA: por razones de permisos **se recomienda usar el puerto 1024**, diferente del puerto SNMP por defecto (161). Además se recomienda que el simulador se asocie a la interfaz localhost.



Para iniciar el simulador usando una grabación el el directorio "data" y escuchando en el puerto 1024:

```
snmpsim-command-responder --data-dir=./data  
--agent-udpv4-endpoint=127.0.0.1:1024
```

## Archivos incluidos

1. challenge\_db.sql: dump de la BDD SQL con la tabla **leases**.
2. challenge\_db.json: dump de la BDD MongoDB con la collection **leases**.
3. private.snmprec: grabación SNMP de un CM con comunidad "private".

## Entregables

Deberá entregar todos los archivos que considere necesarios para poder probar la aplicación, incluyendo código fuente y un dump de la base de datos.

Deberá proporcionar instrucciones para poder probar el correcto funcionamiento de la aplicación.