Documentacion ejercicio 3 RI

Se exponen los pasos seguidos para llegar a la solucion.

1.- El ejercicio en cuestión es el siguiente:

Se desea obtener una lista exhaustiva de los medicamentos utilizados por los usuarios que

aparecen en la colección.

Describir de manera detallada los pasos seguidos para explorar la colección, cómo se ha

construído la consulta o consultas que han permitido llegar a una lista potencial y qué recursos

externos se han empleado para validar la información obtenida.

¡Atención! No es necesaria la validación automática de la lista de medicamentos obtenida pero

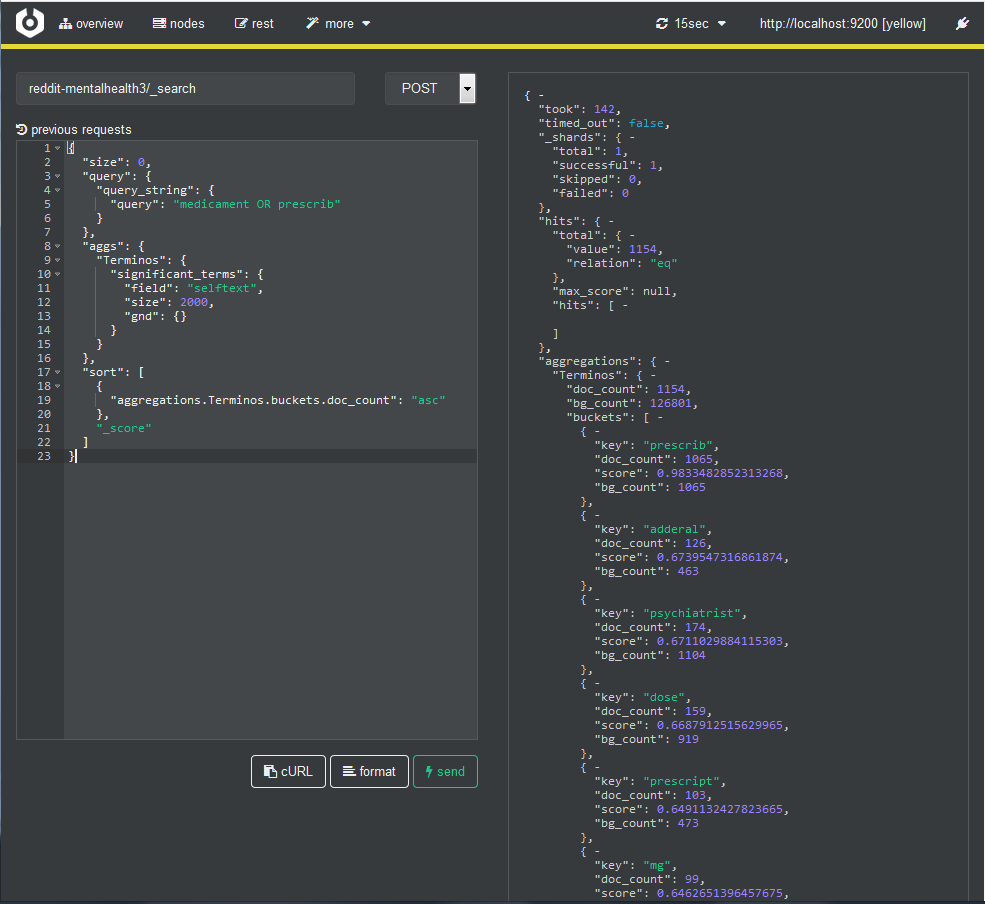
sí se valorará positivamente que se explore qué posibilidades de automatización existen al

respecto.

Si se usa algún conocimiento experto de partida debe indicarse claramente en la

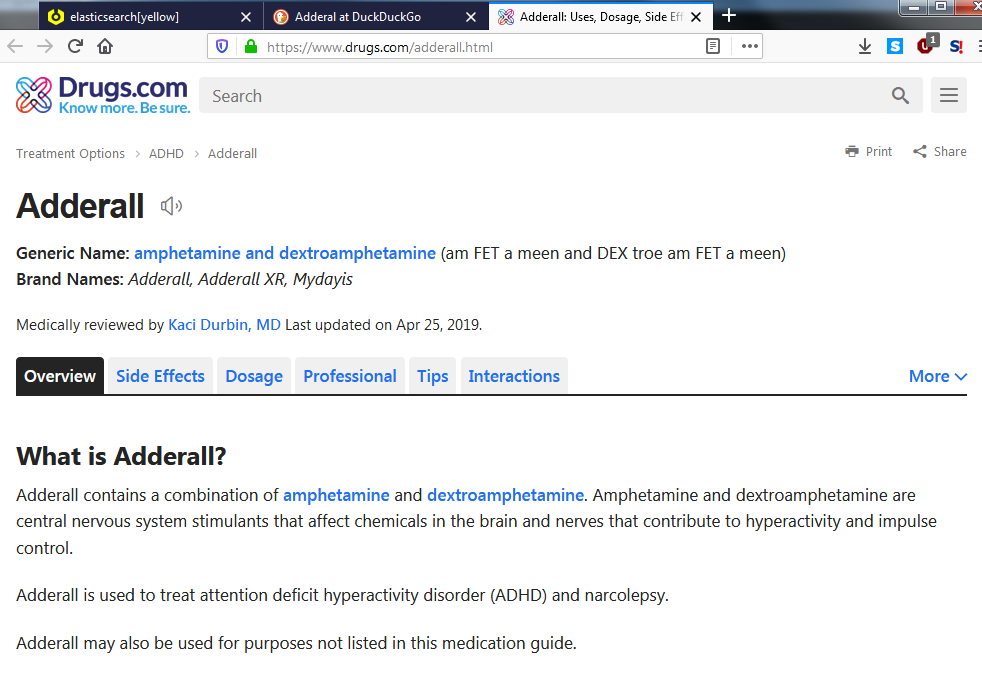
documentación.

2.- Se ha empezado por construir una pequeña consulta en elasticsearch en cerebro donde sacar las claves para luego pasarlo a python usando un script.



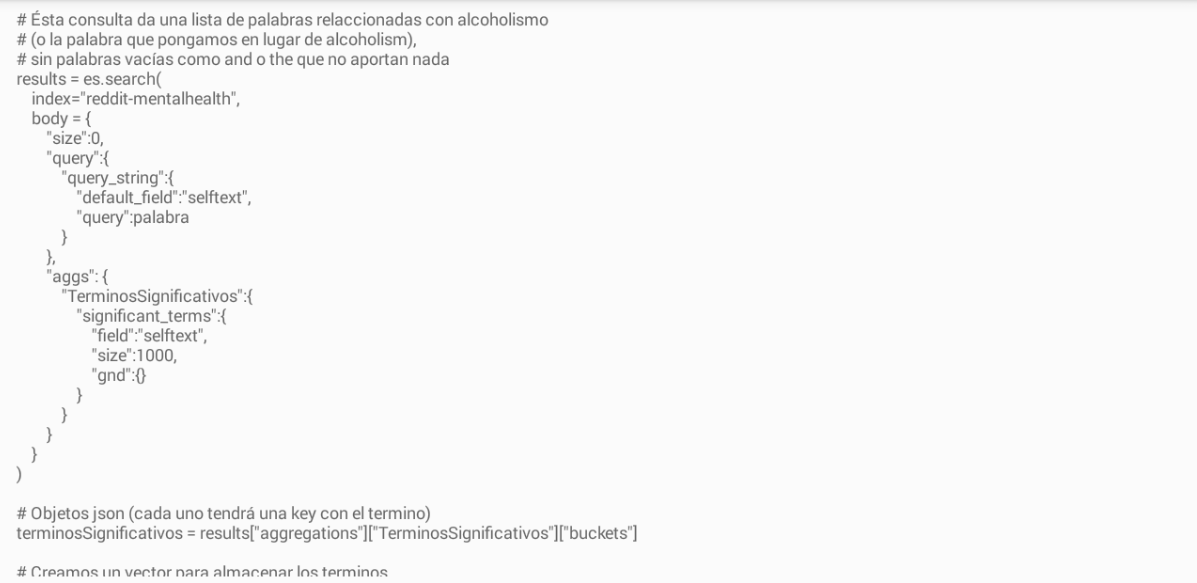
3.- Se ha buscado en google translate las palabras relacionadas con posible medicacion, en español normalmente se recetan o se hace mención al medicamento, se ha construido la consulta en torno a eso

4.- Se ha buscado algún termino y comprobar que efectivamente, es una medicina, a lo largo de la documentacion se ha usado Adderal, ya que por orden es de los primeros y sale como medicina en varias paginas.



5.- A partir de aquí, se ha construido una consulta en python, ayudada por un compañero de grupo, donde obtener los diferentes términos que aparecen en la consulta y guardarlos en un fichero. Se ha usado el entorno de PyScript por su facilidad para debuggear.

Codigo del compañero:

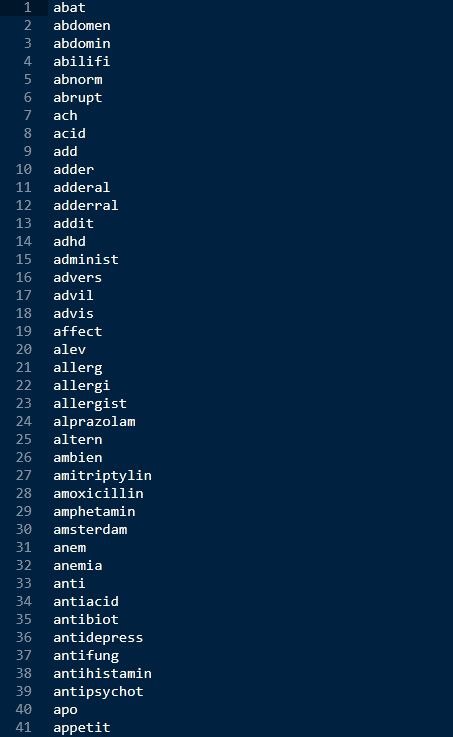


No se ha guardado una copia de la primera versión de este código, se pone un ejemplo aproximado:



6.- Tras abrir el resultado, se vio que aparecen las cantidades de la medicina, 10mg, 1g, 100mg… Como tiene facil solucion y solo esos terminos tienen numero, se añadio una sentencia if que comprueba si hay algun numero en el termino para que no se añadiese.



7.- Esta lista fue ordenada alfabeticamente y se fue comprobando diversos elementos para ver que algunos salian como medicinas (paracetamol, adderal, amoxicillin) y otros no (area, april, asian).

8.- En este punto se le pidio ayuda al profesor Daniel Gayo, quien amablemente me respondió con varias opciones

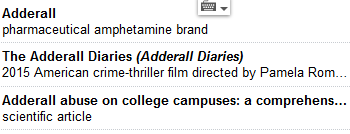
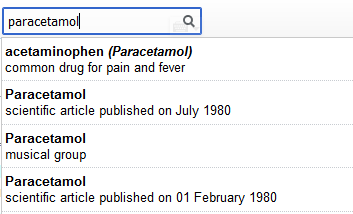
1 – Decir como se haria y enviar la lista tal cual

2 – Usar la API de wikidata

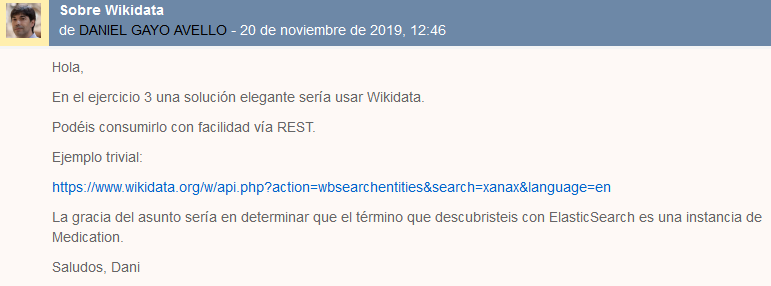
3 – Cruzar esta lista con otras que ya existen

Dado que habia un pequeño ejemplo de uso y todavia se tenian algunos dias hasta la entrega, se decidió tratar de usar la API de wikidata con Python

9.- Wikidata es una pagina que contiene elementos o palabras y se clasifican “segun lo que son”. Por ejemplo, Paracetamol es un medicamento pero tambien un grupo musical. Adderal es un medicamento pero tambien tiene otras caracteristicas.



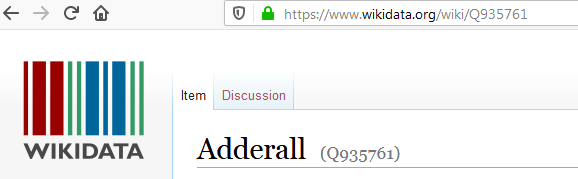
10.- A partir de aquí se uso el consejo de Daniel Gayo del foro, comprobar que el termino es una instancia de medication



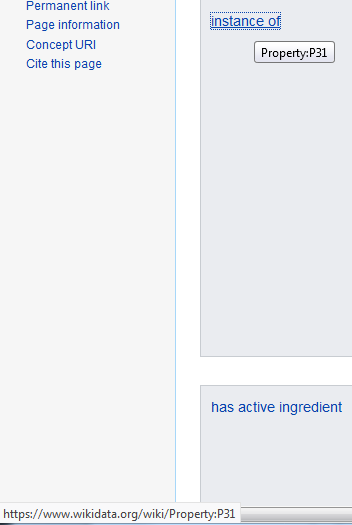
11.- Despues de investigar, se conoció como trabaja Wikidata:

Wikidata tiene muchos datos y palabras, asigna a cada tipo de dato un identificador unico, estos identificadores tienen propiedades, y las propiedades también estan identificadas por su id. Ejemplo:

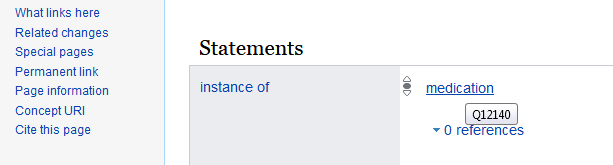
Adderal tiene de id Q935761 <https://www.wikidata.org/wiki/Q935761>



Tiene diferentes propiedades, una de ella instance of. Esta propiedad tiene el id P31, inicalmente se supo por la redirección del enlace (abajo del todo).



Dentro de esta propiedad, medicacion tiene el id Q12140

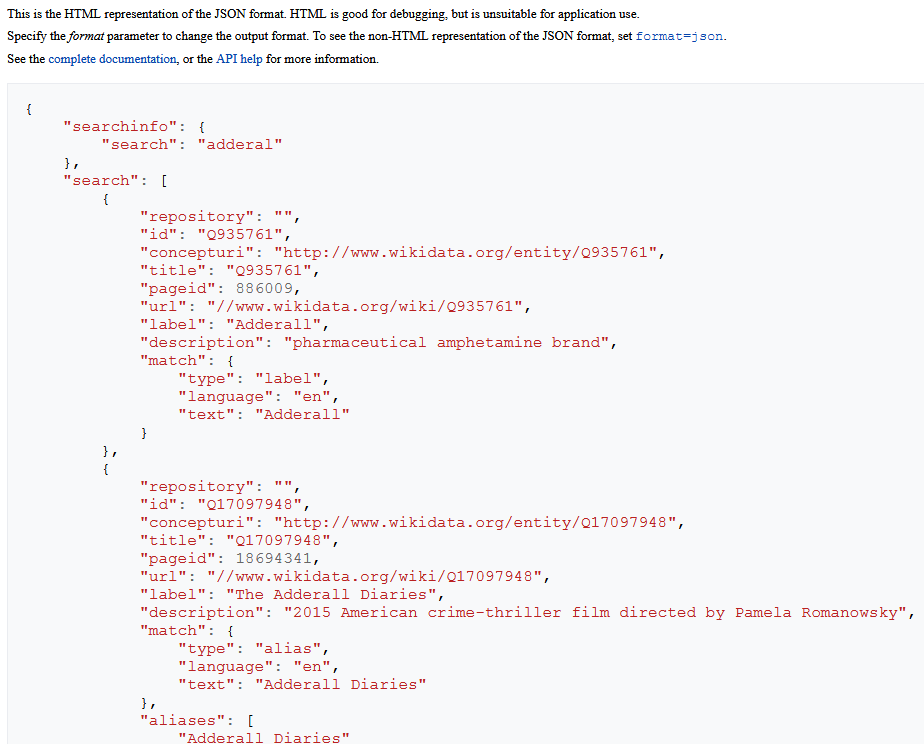


Resumiendo, se obtiene la id de un termino y luego se mira que tenga como propiedad P31 y dentro algun valor sea Q12140.

12.- Se estudió la API de wikidata para ver como forma los datos y de que forma se puede trabajar con ellos, primeramente de forma manual con el ejemplo de Adderal. El proceso manual que se siguió es como sigue:

Para averiguar la id se usa el ejemplo del foro, solo que en lugar de Xana se puso Adderal

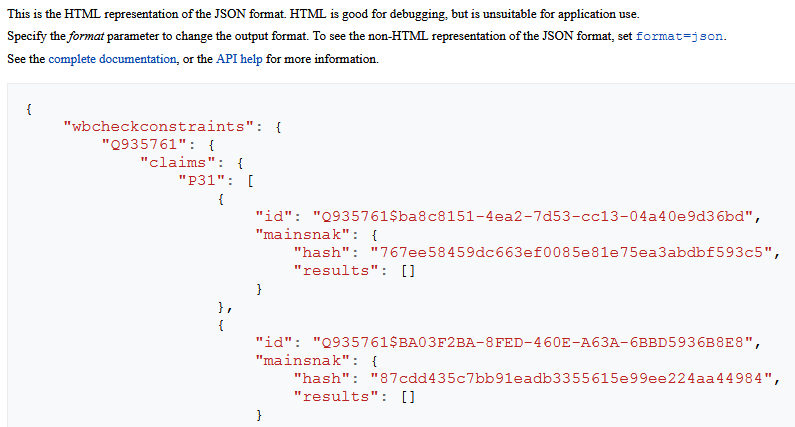
<https://www.wikidata.org/w/api.php?action=wbsearchentities&search=adderal&language=en>



Aparecen varias ids, la que nos interesa es la primera, Q935761.

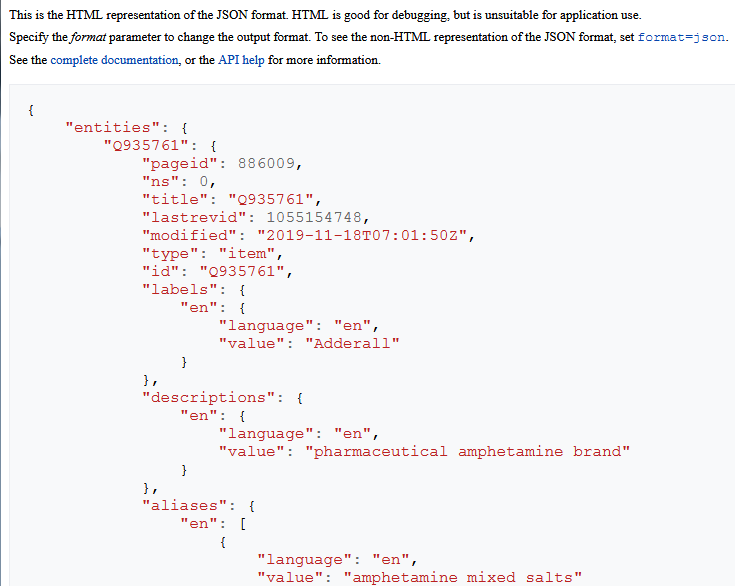
Con esta id, se averigua que constraints tiene, es decir, sus caracteristicas.

<https://www.wikidata.org/w/api.php?action=wbcheckconstraints&id=Q935761>

Y aquí falla, ya que no se puede saber si es un medicamento o no a través de este tipo de consulta, el hash es unico y no hay ningun dato que identifique a todos los medicamentos.

13.- Se siguió investigando y se encontró otra funcion de la API que si tiene los datos que se necesitan. Wbgetentities

<https://www.wikidata.org/w/api.php?action=wbgetentities&ids=Q935761&languages=en>



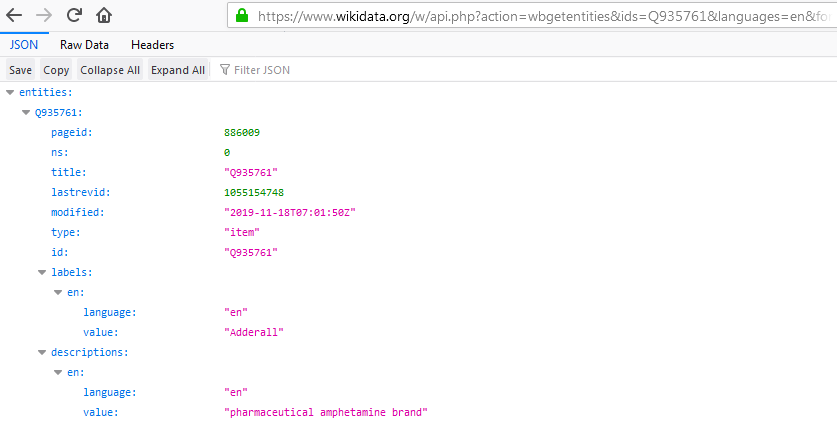
Este es el comienzo, la parte que nos interesa tiene que ver con P31 (propiedad “instance of”). Se busca…



Y sale que tiene la id que buscamos, Q12140, y es unica porque wikidata no repite ids, es mencionado en su manual.

14.- Entoces el proceso manual cambia a – se tiene un termino → se convierte a una id→ se mira si tiene P31 y un valor de id Q12140, el paso que falta es automatizarlo con un script.

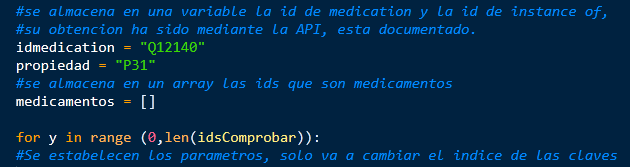
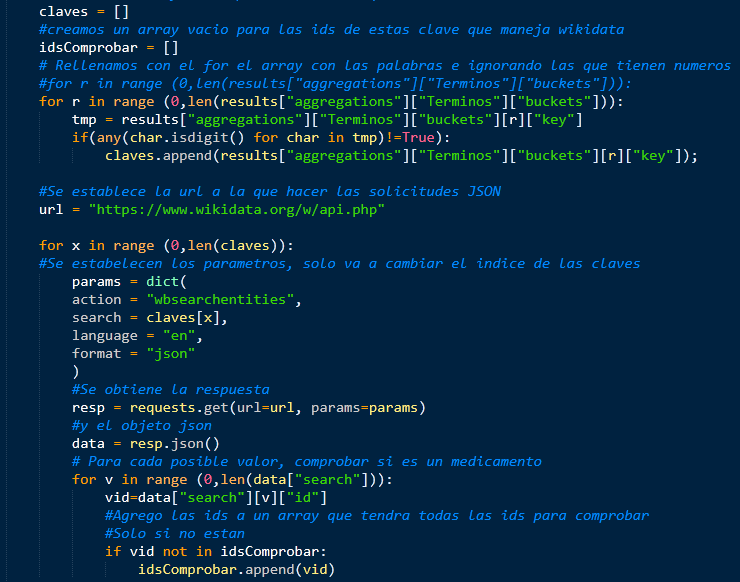
15.- Se investiga algo mas profundamente en como trabajar con JSON en python, también se fija en que la parte superior de cada solicitud tiene un format=json, se obtiene un json con exactamente el contenido html



16.- La forma mas sencilla de hacer solicitudes a la web es usando la librería requests de python, con un ejemplo sencillo de StackOverflow se pudo adaptar a este caso:



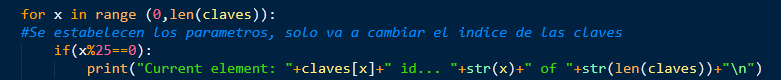
17.- Se altera el código añadiendo diferentes arrays para la id, las propiedades y la lista de medicinas final. Todo el código esta comentado para facilitar su comprensión



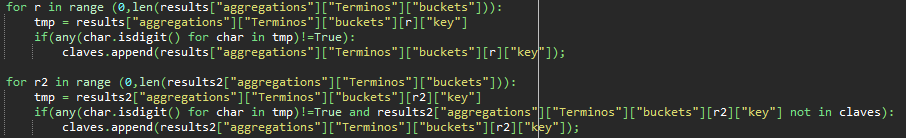
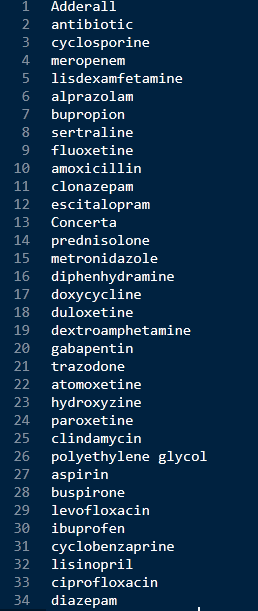
…

Recorre los arrays y hace consultas a la api de wikidata explicada anteriormente, al final guarda los datos en el fichero. Como no se sabe si el medicamento se corresponde a la primera id (ej, adderal es Q935761) u otras (adderal tambien tiene la id de Q17097948), hay que añadir y comprobarlas todas.

18.- Es un proceso bastante lento y puede parecer que este bloqueado, se decidió poner a modo de barra de proceso, una información de cuanto le queda y por donde va:



19.- Una vez finalizado el programa ,crea un fichero medicamentos.txt con la lista de medicamentos. Se confirmó que el programa funciona.



20.- A partir de aquí se puede obtener mas o menos exhaustividad que se desee con cualquier numero de consultas, hay que tener en cuenta que a mayor exhaustividad, mayor duración ya que tiene que ver y comprobar mas términos.