

2018/2019

# RIWS – Memoria de la práctica

Daniel Núñez Sánchez ([daniel.nunezs@udc.es](mailto:daniel.nunezs@udc.es))

Rodrigo Dopazo Iglesias ([r.dopazo@udc.es](mailto:r.dopazo@udc.es))

## Índice

1. Desarrollo del proyecto.....	2
2. Funcionalidades implementadas.....	5
2.1. Crawler .....	5
2.2. Search Engine .....	8
2.3. Interfaz web .....	11
3. Tecnologías utilizadas.....	16
4. Guía para probar la práctica de manera rápida .....	18
5. Problemas encontrados realizando la práctica .....	19
6. Referencias .....	20

## 1. Desarrollo del proyecto

Este proyecto consiste en recoger y **recuperar información sobre fármacos** que se encuentran en el dominio <https://www.vademecum.es/>, para luego realizar búsquedas desde una **interfaz web desarrollada**, haciendo uso de peticiones *HTTP* asíncronas que interpretará un **Search Engine** (en este caso, **ElasticSearch**) para devolver las respuestas que más se adecúen a dicha búsqueda.

La página web **Vademecum** está reconocida oficialmente por el *Ministerio de Sanidad, Servicios Sociales e Igualdad* como soporte válido para ser consultado por profesionales sanitarios.

Este es un ejemplo de la información que nos ofrece la página web de Vademecum para un fármaco concreto, la más interesante puede ser: Nombre del medicamento, el prospecto (qué es y para que se utiliza), las alertas por composición, el sistema de clasificación ATC, el principio activo PA y los excipientes EXC. También muestra información referente a los diferentes envases: el tipo de envase, tipo de uso del medicamento, detalles adicionales, si está financiado por el SNS y los códigos identificativos.

The screenshot displays the Vademecum.es website interface. At the top, there is a search bar with the text "Su fuente de conocimiento farmacológico" and a "Buscar" button. Below the search bar is a navigation menu with links: Medicamentos, P.A., Monografías PA, Clasificación ATC, Laboratorios, Enfermedades, and Regístrate | Entra. The main content area is titled "A.A.S. Comp. 100 mg". On the left, there are tabs for "Datos generales", "Prospecto", and "Interacciones". Under "Datos generales", there are links for "Equivalencias internacionales" and "Dopaje/ deporte". Below this, there is a section for "Alertas por composición:" with icons for "Lactancia" and "Embarazo". Further down, there is a box containing "ATC: Acetilsalicílico ácido, antitrombótico", "PA: Acetilsalicílico ácido", and "EXC: Almidón de maíz, Amarillo naranja S (E-110), Manitol y otros." On the right side, there is a red button labeled "Ver las interacciones" and a section titled "Medicamentos de SANOFI" with a list of various medications including A.A.S. Comp. 100 mg, A.A.S. Comp. 500 mg, ACIDO ALENDRONICO SEMANAL ZENTIVA Comp. 70 mg, ACOVIL Comp. 10 mg, ACOVIL Comp. 2,5 mg, ACOVIL Comp. 5 mg, ADENOCOR Sol. iny. 6 mg/2 ml, ADENOSCAN Sol. para perfusión 30 mg/10 ml, ALDURAZYME Concentrado para sol. para perfusión 100 U/ml, AMARYL Comp. 2 mg, AMARYL Comp. 4 mg, ANTICONGESTIVA CUSI PASTA LASSAR Pasta 250 mg/g, ANTISTAX Comp. recub. con película 360 mg, APIDRA Sol. iny. 100 U/ml en cartucho, and APIDRA Sol. iny. 100 U/ml en vial.

Envases	
<div>Env. con 500</div> <p>TLD: Medicamento de dispensación renovable</p> <p><input type="radio"/> Dispensación sujeta a prescripción médica</p> <p>Comercializado: <b>No</b></p> <p>Situación: <b>Alta</b></p> <p>Código Nacional: 614537</p> <p>EAN13: 8470006145371</p> <p>Conservar en frío: No</p>	<div>Env. con 30</div> <p>TLD: Medicamento de dispensación renovable</p> <p><input type="radio"/> Dispensación sujeta a prescripción médica</p> <p>Fi: Medicamento incluido en la financiación del SNS</p> <p>Facturable SNS: Si</p> <p>Comercializado: <b>Si</b></p> <p>Situación: <b>Alta</b></p> <p>Código Nacional: 686580</p> <p>EAN13: 8470006865804</p> <p>Conservar en frío: No</p>
<p>Datos generales de A.A.S.</p> <p>Composición de A.A.S.</p> <p><b>Principio Activo:</b></p> <p>Acetilsalicílico ácido 100 mg/1 comprimido</p> <p><b>Excipiente:</b></p> <p>Almidón de maíz Amarillo naranja S (E-110) Manitol</p>	

- APIDRA SOLOSTAR Sol. iny. Plumas prec 100 U/ml
- APROVEL Comp. recub. con película 150 mg
- APROVEL Comp. recub. con película 300 mg
- APROVEL Comp. recub. con película 75 mg
- ARAVA Comp. con cubierta pelicular 10 mg
- ARAVA Comp. con cubierta pelicular 100 mg
- ARAVA Comp. con cubierta pelicular 20 mg
- ARIPIPRAZOL ZENTIVA Comp. 10 mg
- ARIPIPRAZOL ZENTIVA Comp. 15 mg
- ARIPIPRAZOL ZENTIVA Comp. 5 mg
- ATORVASTATINA ZENTIVA LAB Comp. recub. con película 10 mg
- ATORVASTATINA ZENTIVA LAB Comp. recub. con película 20 mg
- ATORVASTATINA ZENTIVA LAB Comp. recub. con película 40 mg
- ATORVASTATINA ZENTIVA LAB Comp. recub. con película 80 mg
- ATORVASTATINA ZENTIVA LAB Comp. recub. con película 80 mg
- ATROVENT NASAL Sol. para inhal. 0,30 mg/ml
- AUBAGIO Comp. recub. con película 14 mg
- AZITROMICINA ZENTIVA Comp. recub. con película 500 mg
- AZITROMICINA ZENTIVA Granulado pra susp. oral 500 mg
- BENEFLUR Comp. recub. 10 mg
- BENEFLUR Polvo para sol. iny. y para perfusión 50 mg/vial
- BENESTAN Comp. recub. con película 2,5 mg
- BENESTAN RETARD Comp. de liberación prolongada 5 mg
- BISOLFREN Comp. recub. con película 200/30 mg

Después de analizar diferentes fármacos y ver qué diferencias existen en la estructura de la información mostrada, se ha decidido seleccionar los **datos más relevantes**, como pueden ser:

- **Nombre** del fármaco.
- **Qué es** ese fármaco y **para qué se utiliza**.
- Sistema de Clasificación Anatómica, Terapéutica, Química (**ATC**).
- Principio Activo (**PA**).
- Excipiente (**EXC**).
- **Alertas por composición** (e.g. conducción de vehículos/maquinaria, Embarazo...).
- **Envases**:
  - Número de **envases disponibles**.
  - **Tipo de envase** (e.g. envases con 30 comprimidos, 500 comprimidos...).
  - **Tipo de uso del medicamento** (e.g. genérico, de uso hospitalario, de larga duración...).
  - **Detalles adicionales** del envase.
  - Si el envase está **financiado por el Sistema Nacional de Salud (SNS)** de España.
  - Si el envase **se comercializa** actualmente.
  - El estado de la **situación del envase**.
  - El **Código Nacional de Parafarmacia** (válido en España).
  - El Número de Artículo Europeo (**EAN-13**).
  - **Precio de venta al público** (está oculto en el documento "hidden").
  - **Precio de venta del laboratorio** (está oculto en el documento "hidden").
- **URL del fármaco**.
- **URL del prospecto**.

Conseguir recopilar la información de dicha página ha sido una tarea ardua, ya que esta no mantiene una estructura uniforme para todos los medicamentos. Hai secciones que en algunos casos aparecen muy detalladas, pero en otros casos ni se muestran o falta información de propiedades relevantes.

Una vez descargada y procesada la información procedente de la página *Vademecum*, se ha creado el índice “**vademecumindex**” en *ElasticSearch* en el cual se ha almacenado la colección de documentos. También se le asignó el tipo “**farmacos**” para indicar que esos documentos pertenecen a una categoría que mantiene una semántica común.

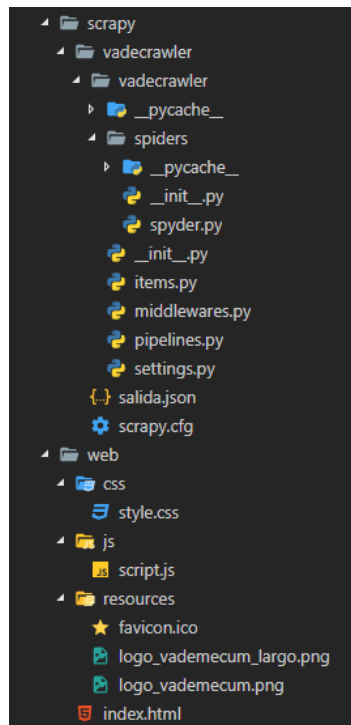
**Cada fármaco se almacena** y se representa dentro de *ElasticSearch* **como un documento indexado**, lo cual permite analizar en tiempo real grandes cantidades de datos y realizar búsquedas complejas sobre estos.

Por último, para realizar las consultas de una manera más cómoda para el usuario, **se ha decidido implementar una interfaz web** más amigable que una simple consola, haciendo uso de las tecnologías *HTML5*, *CSS3*, *JavaScript*, *jQuery*, *Bootstrap* y *Font Awesome*. Esta realizará peticiones **AJAX**, y en el *Body* se enviará la consulta (en formato *JSON*) que desea llevar a cabo el usuario de la aplicación y la recibe el servidor **ElasticSearch**. Este le enviará la respuesta de vuelta, y **utilizando JavaScript se formateará esa respuesta** para mostrársela al usuario en la interfaz de la página principal.

## 2. Funcionalidades implementadas

### 2.1. Crawler

La estructura principal de este proyecto consta de **2 partes diferenciadas**. La **primera** está relacionada con la **recopilación de información** de la página web *Vademecum*, para ello se ha utilizado **Scrapy**. Y la **segunda parte** es la **interfaz web** que será la encargada de realizar las consultas a través de peticiones *HTTP* asíncronas hacia el servidor *ElasticSearch* y también de mostrarle los resultados al usuario.



La estructura de **Scrapy** se ha generado utilizando el comando “**scrapy startproject vadecrawler**”. Esto crea los siguientes ficheros:

- **scrapy.cfg**: contiene la información referente al despliegue. No ha sido necesario modificarlo.
- **items.py**: se definen las diferentes propiedades que se almacenarán de los fármacos cuando se estén descargando y procesando las páginas web. En la imagen que se muestra a continuación se pueden ver todos los campos que se almacenaron para cada fármaco:

```

10 class VadecrawlerItem(scrapy.Item):
11     name = scrapy.Field()           # Nombre del farmaco
12     whatis = scrapy.Field()         # Que es ese farmaco y para que se utiliza
13     url = scrapy.Field()            # URL del farmaco
14     leafletUrl = scrapy.Field()     # URL del Prospecto del farmaco
15     ATC = scrapy.Field()            # Sistema de Clasificación Anatómica, Terapéutica y Química
16     PA = scrapy.Field()             # Principio Activo
17     EXC = scrapy.Field()            # Excipiente
18
19     numCompositionAlerts = scrapy.Field() # Numero de alertas por composicion
20     compositionAlerts = scrapy.Field()    # Alertas por composicion
21     numContainers = scrapy.Field()        # Numero de envases
22     containers = scrapy.Field()           # Informacion de los envases
23
24
25 class ContainerItem(scrapy.item.Field):
26     containerType = scrapy.Field()        # Tipo de envase (Envase con 30 comprimidos, 500 comprimidos...)
27     pubPrice = scrapy.Field()             # Precio de venta al publico
28     labPrice = scrapy.Field()             # Precio de venta del laboratorio
29     drugType = scrapy.Field()             # Tipo de uso del medicamento (Generico, de uso hospitalario...)
30     details = scrapy.Field()              # Detalles adicionales del envase
31     sns = scrapy.Field()                  # Si el envase esta financiado por el Sistema Nacional de Salud (SNS)
32     billableSNS = scrapy.Field()          # Facturable SNS
33     marketed = scrapy.Field()             # Comercializado
34     situation = scrapy.Field()            # Situacion (Alta o no)
35     nationalCode = scrapy.Field()         # Codigo Nacional de Parafarmacia (valido en ESP)
36     EAN13 = scrapy.Field()                # Numero de Artículo Europeo (EAN-13)
37
38
39 class DrugTypeItem(scrapy.item.Field):
40     EFG = scrapy.Field()                 # Medicamento Genérico
41     EFP = scrapy.Field()                 # Medicamento publicitario
42     H = scrapy.Field()                   # Medicamento de uso hospitalario
43     DH = scrapy.Field()                  # Medicamento de diagnostico hospitalario
44     ECM = scrapy.Field()                 # Medicamento de especial control médico
45     TLD = scrapy.Field()                 # Tratamiento de larga duracion
46     MTP = scrapy.Field()                 # Medicamento tradicional a base de plantas

```

- **middlewares.py:** aquí se pueden configurar funcionalidades personalizadas para procesar las respuestas que son enviadas a los spiders.
- **pipelines.py:** después de que un elemento ha sido recopilado por un spider, se envía al pipeline para ser procesado. Los usos típicos de los pipelines son: limpieza de datos HTML, validar los datos, buscar duplicados y almacenar dichos elementos.
- **settings.py:** la configuración de *Scrapy* permite personalizar el comportamiento de los componentes (core, extensiones, pipelines o spiders). En este caso se ha configurado *ScrapyElasticSearch* para almacenar los *Scrapy Items* recopilados con los spiders (documentos) en el índice deseado de *ElasticSearch*.
- **spyder.py:** es una araña que se encarga de descargar y recopilar las páginas web que va recorriendo (empezando por la página “<https://www.vademecum.es/medicamentos-a-1>”). También se debe indicar como tiene que seguir los enlaces en las páginas web y cómo analizar el contenido de la página descargada para extraer las propiedades deseadas.

La siguiente imagen es una captura del inicio del fichero “**spyder.py**”, donde se establece en **qué URL empezará el crawling** la araña, qué **dominios** están **permitidos** para recorrer y las **reglas** utilizadas para obtener los enlaces. Se definen concretamente dos, una que nos permita obtener la lista de medicamentos que queremos **crawlear** para obtener la información y que contiene una función de callback denominada **parse\_item**

en la cual se parsea el código html de cada página web obteniendo los datos a través de **expresiones XPATH** y otra, utilizada para extraer los enlaces del conjunto de agrupaciones por orden alfabético que hace la página. Ambas reglas obtendrán enlaces que cumplan con la expresión especificada en el campo **allow**.

```

1 import scrapy
2 from scrapy.spiders import CrawlSpider, Rule
3 from scrapy.linkextractors import LinkExtractor
4 from vademecum.items import VademecumItem, ContainerItem, DrugTypeItem
5
6 URL_BASE = 'https://www.vademecum.es'
7
8 class MySpyder(CrawlSpider):
9     name = 'spyder'
10    allowed_domains = ['vademecum.es']
11    start_urls = [
12        'https://www.vademecum.es/medicamentos-a-1',
13    ]
14
15    rules = (
16        Rule(LinkExtractor(allow=('https://www.vademecum.es/medicament*'), restrict_xpaths=('//div[@role="content"]/ul[@class="no-bullet"]/li//a')), callback='parse_item'),
17        Rule(LinkExtractor(allow=('https://www.vademecum.es/medicament*'), restrict_xpaths=('//div[@role="content"]/select/option'), tags=('a', 'option'), attrs=('value'))),
18    )
19
20    def parse_item(self, response):
21
22        ## NOMBRE FARMACO
23        item['name'] = response.xpath('//div[@role="content"]/div/h1/span/text()').extract_first()
24
25        ## URL FARMACO
26        item['url'] = response.request.url
27
28        ## URL PROSPECTO
29        leafletUrl = response.xpath('//div[@role="content"]/div/ul/li/a[@id="m1_2"]/@href').extract_first()
30        if (leafletUrl):
31            item['leafletUrl'] = URL_BASE + leafletUrl
32
33        ## PROPIEDADES: ATC, PA y EXC
34        properties = response.xpath('//h3["Envases"]/parent::*/table/tr')
35        for prop in properties:
36            propElement = prop.xpath('normalize-space(td/text())').extract_first()
37            if (propElement.startswith('ATC')):
38                item['ATC'] = prop.xpath('normalize-space(td/a/strong)').extract_first()
39
40            if (propElement.startswith('PA')):
41                paElement = prop.xpath('td/a')
42                for pa in paElement:

```

La estructura de **Web** se ha creado de manera manual y se han añadido ficheros según se iban necesitando.

- **index.html**: es la página web propiamente dicha, escrita en el lenguaje de marcado *HTML5*. Esta muestra una interfaz con el formulario para rellenar la información deseada y buscarla entre los 15000 medicamentos indexados.
- **style.css**: se ha creado este fichero para definir el diseño visual de la página web que se muestra al usuario, usando el lenguaje de diseño gráfico *CSS3*. Este permite aplicar los estilos, colores, márgenes, etc. a todo el contenido del documento estructurado.
- **script.js**: en este fichero se encuentra la lógica encargada de realizar las peticiones *HTTP* asíncronas con ayuda de *jQuery AJAX*. Está escrito en *JavaScript* y también realiza el parseo del objeto *JSON* resultante de la búsqueda devuelto por *ElasticSearch* al formato *HTML5* que será mostrado al usuario final de la aplicación.
- **resources**: esta carpeta contiene las imágenes estáticas como, por ejemplo, el icono de la página web y los logos de *Vademecum* que se muestran en la barra de navegación.



## 2.2. Search Engine

Una vez crawlados los datos, es necesarios indexarlos en nuestro search engine para que puedan ser consultados. En nuestro caso, utilizamos un plugin disponible en el crawler que nos permite automatizar la creación tanto del índice como del mapping de los campos necesarios. Estas acciones, se pueden realizar de manera manual empleando el API de elasticsearch (Search Engine utilizado en la práctica).

Ejemplo de creación del índice:

```
PUT http://[ip]:[puerto]/vademecumindex
```

Una petición PUT a nuestro servidor indicando el nombre del índice que queremos crear nos construiría un índice con dicho nombre y la configuración por defecto. Esta configuración se puede especificar en el cuerpo de la petición indicando por ejemplo:

- number\_of\_shards: 5
- number\_of\_replicas: 2

Estos valores de configuración serán importantes a lo hora de definir índices de gran volumen, que nos permitan dividir y escalar, en nodos, horizontalmente el volumen así como ofrecer disponibilidad en caso de que un nodo fallase.

En el propio cuerpo de la petición anterior se podría indicar el mapping deseado. En caso de no incluirlo, se podría realizar una petición específica que nos permita crear un mapping.

```
PUT http://[ip]:[puerto]vademecumindex/_mapping/farmacos
```

En esta petición, creamos el mapping “fármacos” para el índice “vademecumindex”. En el cuerpo de esta petición se incluiría la información del mapping que se quiere realizar.

Una vez indexados los documentos, procederemos a la creación de queries que nos permitan consumir dichos documentos. A continuación, se mostrará el ejemplo del esquema seguido para la realización de estas.

```
{
  "query": {
    "bool": {
      "must": [
        { "multi_match": {
          "query": "mocos y tos",
          "fields": ["name", "whatIs^2"]
        } },
        { "bool": {
          "must_not": [
            { "match": { "PA": "Cafeina" } }
          ]
        } }
      ],
      "should": [
        { "match": {
          "name": "**KABI*"
        } }
      ],
      "filter": [
        { "range": { "containers.pubPrice": { "lt": "30.0", "gt": "5.0" } } }
      ]
    }
  },
  "aggs": {
    "avg_public_price": {
      "avg": {
        "field": "containers.pubPrice"
      }
    },
    "max_public_price": {
      "max": {
        "field": "containers.pubPrice"
      }
    },
    "min_public_price": {
      "min": {
        "field": "containers.pubPrice"
      }
    }
  }
}
```

- **query:** Este campo determina la definición de la query que queremos realizar
  - **bool:** Permite obtener documentos en función de la combinación booleana de las definiciones incluidas en esta etiqueta.
    - **must:** Indica que la cláusula incluida en este campo debe aparecer en los documentos e influirá para el score de cada uno.
    - **must\_not:** La cláusula incluida en este campo no debe aparecer en el documento.
    - **should:** Indica que la cláusula debe aparecer en el documento. Si se incluye dentro de un bool con otras cláusulas must, esta actuará como un OR booleano, significando que su condición no es obligatoria para que se cumpla. Por esto mismo, esta cláusula se usa únicamente para influir en el score final.
    - **filter:** Indica que la cláusula incluida en este campo debe aparecer en los documentos, pero, a diferencia de la cláusula “must”, esta no influirá para el score final de cada documento.
- **aggs:** Permite agrupar los datos en función de los resultados de una búsqueda. Existen diferentes tipos de agregaciones y en nuestro ejemplo se usan agregaciones métricas que permiten obtener, en cada búsqueda, el valor máximo, mínimo y media del precio de los envases de cada fármaco.

En este ejemplo de query, se hará match de “mocos y tos” sobre los campos “name” y “whatis”, nombre y definición, respectivamente. Este último tiene un boosting (^2) que otorgar mayor score a los documentos que hagan match en dicho campo.

Al ser una boolean query, su resultado se combinará booleanamente con la condición siguiente, que el “PA” no sea “Cafeina”. A mayores, si el nombre incluye la palabra “KABI”, otorgará mayor score al documento. Por último, se aplica un filtro sobre el resultado de la query, que filtrará los documentos que contengan envases cuyo precio sea menor que 30 y mayor que 5.

## 2.3. Interfaz web

Esta es la interfaz web que aparece al abrir la página “**index.html**”. En ella se encuentran los filtros que permiten realizar búsquedas avanzadas sobre los 15000 fármacos indexados en *ElasticSearch*.

- Al pulsar el botón “**Realizar búsqueda**”, se mostrarán en la parte inferior de la interfaz la cantidad de resultados que se escoja en el selector de la derecha (permite **mostrar los 10, 20, 50 o 100 resultados** que más se ajusten a las condiciones de la búsqueda).
- El primer input es la “**búsqueda por palabras clave**”. El texto que se introduzca aquí se analizará contra los campos (“fields” en *ElasticSearch*): **nombre del fármaco** y la información del prospecto que explica **qué es ese medicamento y para qué se utiliza**.
- El siguiente input que aparece es el “**Nombre del fármaco**”, el cual se enviará a *ElasticSearch* y se priorizará en los resultados que contengan o se aproximen al texto que se escriba en este campo.
- A la derecha del input anterior, aparece una **barra deslizable** (o “slider”) de rangos que permite mover las dos palancas (o “handle”) y especificar que **solo aparezcan los medicamentos** que se encuentren **entre un precio de venta al público (PVP) mínimo y otro máximo**.

- Debajo de esas entradas de datos está el input para introducir el “**Principio Activo (PA)**”, la cual se puede escoger si se desea que los medicamentos **tengan un PA concreto** usando “**Igual a**” o si se desea encontrar medicamentos que **no tengan un PA** usando “**Diferente de**”.
- El siguiente input es “**Excipiente (EXC)**”, que se utilizaría de la misma forma que en el caso anterior del Principio Activo.
- Por último, si se desea encontrar **medicamentos que no tengan unas alertas por composición** concretas, se pueden marcar cualquiera de las **4 checkboxes**: “**Lactancia**”, “**Embarazo**”, “**Fotosensibilidad**” o “**Conducción de vehículos**”.

En este primer ejemplo se pulsa el botón “**Realizar búsqueda**” sin rellenar ni filtrar por ningún campo, y se mostrarían **10 resultados aleatorios** (ya que no se especificó ninguna faceta) de los **14252 medicamentos** que se encuentran **en total** indexados en *ElasticSearch*.

Práctica RIWS 2018

Página información original: Vademecum

Filtros

Mostrar 10 resultados

Búsqueda avanzada

Nombre del fármaco:

Rango de precios de venta al público: Desde 0€ hasta - €

Principio Activo (PA):

Igual a

Excipiente (EXC):

Igual a

Ocultar medicamentos con alertas por composición de:

☐ Lactancia
☐ Embarazo
☐ Fotosensibilidad
☐ Conducción de vehículos

La búsqueda realizada ha encontrado **10 resultados** de un total de **14252 documentos**

Información estadística:

- Precio **mínimo** de todos los envases: **0.08 €**
- Precio **máximo** de todos los envases: **22730.15 €**
- Precio **medio** de todos los envases: **102.02 €**

**VISKERN Colirio en solución en envase unidosis 5 mg/ml**

[Ver el prospecto completo...](#)

**Información general:**

- **ATC:** Carmelosa sódica
- **PA:** Carmelosa sódica

**Alertas por composición:**

- Lactancia
- Embarazo

Práctica RIWS 2018

Página información original: Vademecum

Filtros

fiebre y mocos

Realizar búsqueda

Mostrar 50 resultados

Búsqueda avanzada

Nombre del fármaco:

Rango de precios de venta al público: Desde 4€ hasta 257 €

Principio Activo (PA):

Diferente de

Excipiente (EXC):

Igual a

Ocultar medicamentos con alertas por composición de:

☐ Lactancia

☐ Embarazo

☒ Fotosensibilidad

☒ Conducción de vehículos

La búsqueda realizada ha encontrado 50 resultados de un total de 375 documentos

Información estadística:

- Precio mínimo de todos los envases: 1.00 €
- Precio máximo de todos los envases: 1436.98 €
- Precio medio de todos los envases: 31.41 €

ACETILCISTEÍNA NORMON EFG Granulado para sol. oral 200 mg

Ver el prospecto completo...



Información general:


- ATC: Acetilcisteína
- PA: Acetilcisteína

Alertas por composición:

- Lactancia
- Embarazo


Para **cada fármaco, se mostrará toda la información relevante del mismo**. Si se pulsa sobre el nombre del fármaco, se abrirá una nueva pestaña en el navegador con la información en *Vademecum*. Además, a la derecha del nombre se muestra un icono de Información, con un resumen que explica qué es ese fármaco y para qué se utiliza. En “Ver el prospecto completo...” también se puede leer el prospecto completo en la página web original. Los envases de cada fármaco se muestran en una disposición de 3 envases por fila.


**ACETILCISTEÍNA NORMON EFG Granulado para sol. oral 200 mg**



[Ver el prospecto completo...](#)

**Información general:**



- **ATC:** Acetilcisteína
- **PA:** Acetilcisteína
- **EXC:** Aspartamo (E-951), Sorbitol


 Env. con 30 sobres

- **Precio Público:** 5.4 €
- **Precio Laboratorio:** 3.46 €
- **EFG:** Medicamento genérico
- **SNS:** Medicamento excluido de la financiación del SNS
- **Facturable SNS:** No
- **Comercializado:** Sí
- **Situación:** Alta
- **Código Nacional:** 656242
- **EAN13:** 8470006562420

Acetilcisteína pertenece al grupo de medicamentos denominados mucolíticos. Actúa disminuyendo la viscosidad del moco, fluidificándolo y facilitando su eliminación. Este medicamento está indicado para reducir la viscosidad de los mocos y flemas, facilitando su expulsión, en catarros y gripes, para adultos y adolescentes a partir de 12 años. Debe consultar a un médico si empeora o si no mejora después de 5 días de tratamiento.

El siguiente ejemplo se muestra información de un medicamento con 4 envases, cada uno de ellos tiene un precio concreto y un código identificativo diferente, entre otras cosas.


**PARACETAMOL RATIO Comp. efervescente 1 g**



[Ver el prospecto completo...](#)

**Información general:**


- **ATC:** Paracetamol
- **PA:** Paracetamol
- **EXC:** Sodio, Sorbitol

**Alertas por composición:**


- Lactancia
- Embarazo


**Env. con 8 (tiras)**


- **EFG:** Medicamento genérico
- **SNS:** Medicamento no incluido en la financiación del SNS
- **Facturable SNS:** No
- **Comercializado:** No
- **Situación:** Alta
- **Código Nacional:** 684905
- **EAN13:** 8470006849057
- Detalles adicionales:
  - Dispensación sujeta a prescripción médica


**Env. con 20 (tiras)**

- **Precio Público:** 3,9 €
- **Precio Laboratorio:** 2,5 €
- **EFG:** Medicamento genérico
- **SNS:** Medicamento excluido de la financiación del SNS
- **Facturable SNS:** No
- **Comercializado:** Si
- **Situación:** Alta
- **Código Nacional:** 684906
- **EAN13:** 8470006849064
- Detalles adicionales:
  - Dispensación sujeta a prescripción médica


**Env. con 40 (tiras)**

- **Precio Público:** 4,84 €
- **Precio Laboratorio:** 3,1 €
- **EFG:** Medicamento genérico
- **SNS:** Medicamento excluido de la financiación del SNS
- **Facturable SNS:** No
- **Comercializado:** Si
- **Situación:** Alta
- **Código Nacional:** 684907
- **EAN13:** 8470006849071
- Detalles adicionales:
  - Dispensación sujeta a prescripción médica


**Env. con 20 (tubo)**

- **EFG:** Medicamento genérico
- **SNS:** Medicamento no incluido en la financiación del SNS
- **Facturable SNS:** No
- **Comercializado:** No
- **Situación:** Alta
- **Código Nacional:** 684908
- **EAN13:** 8470006849088
- Detalles adicionales:
  - Dispensación sujeta a prescripción médica



### 3. Tecnologías utilizadas

En este apartado se muestran las **tecnologías que se han utilizado** para llevar a cabo y completar esta práctica de Recuperación de Información.

Se ha optado por escoger **Scrapy** para crawlear la información de la web y el servidor de búsquedas **ElasticSearch** para indexar esa información, ya que estas dos tecnologías tienen una gran compatibilidad entre ellas. Ofrecen una alta rapidez de puesta en marcha, y no hay que dedicar excesivo tiempo a muchos ficheros de configuración (aunque también ofrecen la posibilidad de tener una configuración avanzada y minuciosa).

Nosotros personalmente estamos más cómodos utilizando Python como lenguaje de programación principal, por eso elegimos usar Scrapy. Además, la documentación ofrecida por ambas tecnologías, bajo nuestro punto de vista, es de mejor calidad que la elección de *Nutch + Hbase* y *Apache Solr*.

A continuación, se especifican **las versiones concretas del software** que se han utilizado y con las que **se puede asegurar que funciona correctamente**.

1. Instalar o tener instalado **Python 3.6.6**. Es el lenguaje de programación que se ha utilizado para realizar el tratamiento de información de las páginas web crawladas. Se necesita para ejecutar *Scrapy*:  
<https://www.python.org/downloads/release/python-371/>
2. Instalar o tener instalado **Scrapy 1.5**. Es un *framework* escrito en *Python* que permite analizar y extraer la información de páginas web. Este rastreador web tiene una arquitectura basada en “*spiders*”, los cuales son rastreadores independientes que reciben un conjunto de instrucciones. Si se tiene el sistema de gestión de paquetes PIP instalado en el sistema, se puede instalar con “**pip install scrapy**”.  
<https://scrapy.org/download/>

Para lanzar la araña de *Scrapy* se utiliza el comando “**scrapy crawl spyder -o salida.json**”.

3. Instalar o tener instalado **ScrapyElasticSearch 0.9.1**. Este plugin se utiliza para almacenar los *Scrapy Items* en el índice deseado de *ElasticSearch*. Si se tiene el sistema de gestión de paquetes PIP instalado en el sistema, se puede instalar con “**pip install ScrapyElasticSearch**”.  
<https://github.com/knockrentals/scrapy-elasticsearch>

Se especifica esta configuración en el fichero **settings.py** en la carpeta de *Scrapy*.

4. Tener descargado y descomprimido **ElasticSearch 6.4.2**: Es un motor de búsqueda y análisis distribuido *RESTful* que está basado en *Apache Lucene*. Se utiliza para búsquedas de texto completo, análisis de registros, inteligencia de seguridad, análisis de negocios...  
<https://www.elastic.co/es/downloads/elasticsearch>

**[\*] Antes de lanzar ElasticSearch, es necesario añadir dos líneas a su configuración.** Esto es requerido ya que, por seguridad, el servidor bloquea las conexiones de un origen o dominio distinto (**Control de acceso HTTP - CORS**) y no permitirá las peticiones AJAX. En la carpeta “**config**”, modificar el archivo “**elasticsearch.yml**” y añadir:

```
http.cors.enabled: true
http.cors.allow-origin: "*"

```

```
67  ##### SOLUCION PARA "Cross-origin Resource Sharing (CORS) blocked cross-origin response
68  http.cors.enabled: true
69  http.cors.allow-origin: "*"

```

Para lanzar *ElasticSearch*, ejecutar “**bin/elasticsearch**” (necesario tener **Java** instalado).

5. Utilizar un **navegador web** que soporten las tecnologías utilizadas para desarrollar la interfaz web del usuario, como por ejemplo **Google Chrome** o **Firefox**. Dichas tecnologías son: *HTML5*, *CSS3*, *JavaScript (ECMAScript 6)*, *jQuery*, *Bootstrap* y *Font Awesome*.

Abrir el “**index.html**” que se encuentra en la carpeta “**web**” y realizar las búsquedas.

## 4. Guía para probar la práctica de manera rápida

Para **probar esta práctica de manera rápida**, sin tener que esperar el tiempo de *crawlear* y recopilar la información del dominio de *Vademecum* (en nuestro caso, para recopilar todos los fármacos que son casi 15000, ha tardado en procesar todas las páginas sobre 1 hora y 20 minutos), se deben **realizar los siguientes pasos**:

1. Asegurarse de que está **habilitada la configuración de *ElasticSearch*** para permitir el ***CORS - Control de acceso HTTP*** (ver la **nota [\*]** del apartado 3.4.). Si no se modifica, el servidor no permitirá las peticiones *AJAX* realizadas desde el navegador web.
2. Antes de seguir, **renombrar la carpeta “data”** del directorio raíz de *ElasticSearch* (donde se encuentra “bin”, “config” ...) y ponerle **“data\_backup”**, para no sobrescribir tus datos actuales con los datos de nuestra práctica. **Descomprimir** el archivo **“data.zip”** y **mover** esa carpeta descomprimida **“data”** al directorio raíz de *ElasticSearch*.

**Este paso evitará tener que volver a *crawlear*** y lanzar el *spider* de *Scrapy* para recopilar la información. Ya están los casi **15000 medicamentos en esa carpeta**, los cuales cargará *ElasticSearch* una vez esté lanzado.

3. **Arrancar** el servidor *ElasticSearch*, ejecutándolo usando el script **“bin/elasticsearch”** (necesario tener *Java (JRE)* instalado).
4. En la carpeta **“web”** se encuentran los ficheros de la interfaz web de nuestra aplicación. Abriendo el archivo **“index.html”** en un navegador web se muestra la interfaz con la que puede interactuar el usuario para realizar consultas a través del formulario.
5. Este formulario permite realizar **búsquedas por palabras clave**, **seleccionar cuantos resultados se desean mostrar** (10, 20, 50 o 100), especificar el **nombre del fármaco**, seleccionar el **rango de precios de venta al público** entre un valor mínimo y otro máximo, si se desea buscar **“igual a”** o **“diferente de”** un **Principio Activo**, lo mismo para la faceta **Excipiente**. Por último, se filtra qué tipos de **alertas por composición se desean ocultar**.

## 5. Problemas encontrados realizando la práctica

Estos son algunos **problemas encontrados** realizando la práctica, **inicialmente** utilizando **Nutch + Hbase** y **Apache Solr**, y luego **cambiando de tecnologías** para usar **Scrapy** y **ElasticSearch**:

- Hemos tenido problemas al crawlear la información de *Vademecum* utilizando Nutch + Hbase, ya que a la hora de descargarse la página web, lo hacía de forma incorrecta, incluyendo gran cantidad de scripts en vez de los elementos HTML (posiblemente por lo mal construida que está la página). Por este motivo, no fuimos capaces de llegar ni a los elementos a parsear ni a los enlaces que nos permitiesen crear el spider.
- Problemas en la instalación de *Nutch + Hbase* en la plataforma *Windows*. Estábamos muy sujetos a, únicamente utilizar, un entorno *Linux*, en el cual si que fuimos capaces de instalar el entorno.
- Los motivos anteriores nos llevaron a probar con otro crawler diferente. Hemos escogido *Scrapy*, elaborado en *Python* y que es fácilmente portable. Con respecto a *Nutch*, *Scrapy* es mucho más rápido de emplear (en *Nutch* es necesario tener un sistema de ficheros distribuido, *HBase* en este caso) y mucho más sencillo de configurar y ejecutar.
- Utilizando *Scrapy*, a causa de que la estructura de la página web <https://www.vademecum.es/> deja mucho que desear, tuvimos que invertir mucho tiempo para recoger la información contemplando múltiples casos para cada campo. No es uniforme la manera que tienen de etiquetar los elementos de la página.

## 6. Referencias

- Scrapy Spiders:  
<https://docs.scrapy.org/en/latest/topics/spiders.html>
- Scrapy Selectors:  
<https://docs.scrapy.org/en/latest/topics/selectors.html>
- Scrapy Items:  
<https://docs.scrapy.org/en/latest/topics/items.html>
- ElasticSearch – Executing Searches:  
[https://www.elastic.co/guide/en/elasticsearch/reference/current/\\_executing\\_searches.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/_executing_searches.html)
- ElasticSearch – Executing Filters:  
[https://www.elastic.co/guide/en/elasticsearch/reference/current/\\_executing\\_filters.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/_executing_filters.html)
- ElasticSearch – Executing Aggregations:  
[https://www.elastic.co/guide/en/elasticsearch/reference/current/\\_executing\\_aggregations.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/_executing_aggregations.html)
- ElasticSearch – Control de acceso HTTP (CORS):  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-http.html>  
[https://developer.mozilla.org/es/docs/Web/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS)
- API jQuery AJAX:  
<http://api.jquery.com/jquery.ajax/>