

Anotaciones

1. Deberíamos delimitarlo localmente en principio, que funcione, y en base a eso ver si es factible o no ampliar el alcance.
2. Nada que decir.
3. Nada que decir.
4. Nada que decir. No tenemos respuesta.
5. En nuestra investigación acerca de lo que son los Web Bots y de su funcionamiento, nos encontramos con Scrapy, un framework con todas las herramientas necesarias para poder crear una aplicación que extraiga datos de la web, tal y como nosotros necesitamos. Todavía tenemos que ver la compatibilidad con Django, para asegurarnos de que funcione de manera transparente (y de que no necesite de instalación en el servidor, porque si no vamos a tener problemas).

Además, en cuanto a sus **requisitos de instalación**, vemos que utiliza Python 2.7, como lo especifica el proyecto:

- Python 2.7.
- Instalar Pywin32 (que es Python para extensiones de Windows).
- Instalar PIP junto con SetupTools.
- Por último, instalar Scrapy con el comando PIP install Scrapy.

En base a la documentación leída, concluimos que es factible integrar Django con Scrapy.

Otro punto es que Django es un MVC, pero desde nuestro punto de vista, nuestra aplicación no va a tener vistas, por lo que esa capa va a estar sin utilizarse.

6. Apache Solr es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas". Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices, y actualmente se está usando en muchos de los sitios más grandes de Internet. La principal característica de Solr (o al menos la más útil) es su API estilo REST, ya que en vez de usar drivers o APIs programáticas para comunicarnos con Solr podemos hacer peticiones HTTP y obtener resultados en XML o JSON.
7. Va a ser menos complejo, en teoría, al tener que trabajar con menos indicadores.