

# Práctica 4, 5 y 6

**Introducción:** Tras lo implementado en la práctica 2 y 3, se han añadido una serie de funcionalidades a la web existente, donde se mezclan los requerimientos de las prácticas 4, 5 y 6 por tanto se tomará como una única práctica divididas en las diferentes funcionalidades implementadas.

## 1) Páginas web incorporadas

- Restaurants (Búsqueda o creación de nuevo restaurante)
- Restaurants/All (Muestra de todos los restaurantes existentes en la DB con un paginador AJAX)
- Restaurants/Search (Para obtener la información de un restaurante)
- Restaurants/Edit (Para modificar los datos de un restaurante)
- Restaurants/Maps (Para ver la localización a nivel mundial de los restaurantes restringiendo a uno por cada cierto valor de kilómetros cuadrados)
- Restaurants/Statistics (Para ver diferentes gráficos de las características de los restaurantes)
- RSS (Para realizar búsquedas de noticias de un medio digital)

## 2) Arquitectura de la aplicación

Esta aplicación se encuentra organizada de la siguiente forma:

- / (Raíz de la aplicación)
  - pr.py (Programa en Python)
  - static (Contenido estático que utilizará la aplicación)
    - css
    - fonts
    - images
    - js
  - templates (Archivos HTML para dar formato a la web)
  - etree\_parser.py (Lectura de un archivo XML con ETREE)
  - sax\_parser.py (Lectura de un archivo XML con SAX)
  - portada.xml (archivo RSS en formato XML)

## 3) Funcionalidades

### Base de datos MongoDB

Se ha utilizado una base de datos de la propia [documentación](#) de MongoDB que tiene que ver con los diferentes restaurantes que hay por todo el mundo y una serie de características como son: su nombre, ciudad donde se encuentra, tipo de cocina, dirección y valoración.

Se van a utilizar los datos anteriores para realizar una serie de funcionalidades con las APIs y librerías que se proponen en las prácticas.

Para la interacción de la base de datos se ha realizado:

- Búsqueda de un restaurante a partir de su nombre.
- Inserción de un nuevo restaurante con sus datos.
- Modificación de los datos de un restaurante existente:
  - Lectura de texto para la modificación de las características.
  - Lectura de datos de un mapa para la modificación de coordenadas.

## jQuery y AJAX

Para la implementación del paginador en la dirección “<http://localhost:5000/restaurants/all>” se ha utilizado llamadas AJAX con *GET*. Esta página muestra 10 restaurantes y un mapa donde aparece la localización de todos estos y pulsando en los botones del paginador podemos mostrar sin recargar la página 10 restaurantes diferentes con su ubicación.

Todo esto se realiza al decirle a AJAX que la llamada sea con *GET* donde se dan dos parámetros de entrada, *start* y *end* que equivalen a la posición de los restaurantes en la DB. Por tanto dependiendo del botón del paginador, estaremos accediendo a diferentes restaurantes.

Un ejemplo sería pulsando un botón de la página 3, donde estaríamos realizando una llamada AJAX hacia la página web “<http://localhost:5000/restaurants?start=21&end=30>”, obtiene el HTML correspondiente a esa página y modifica el DOM del elemento en cuestión que muestra la información de los restaurantes.

Cada elemento de botón para acceder a una página tiene un evento *onclick* para la llamada a la función que realiza la llamada AJAX.

Tanto en la página de modificación de un restaurante existente como para introducir un nuevo restaurante, se ha añadido una funcionalidad para obtener las coordenadas que usa Google Maps mediante el nombre de una calle y posteriormente mostrar el mapa con el marcador. De esta forma el usuario sólo tiene que escribir el nombre de la calle, el número y la provincia y automáticamente el algoritmo obtiene las coordenadas X e Y referentes a la calle. Mediante jQuery se ha realizado un botón que es el encargado de realizar esto; al escribir la calle en el *input* correspondiente éste botón manda los datos a la API de Google Maps para obtener las coordenadas; luego con jQuery se muestra el mapa (inicialmente oculto) y cambia el valor de los *input* de las coordenadas X e Y.

Otra funcionalidad de jQuery de las que aparecía en el guión de la práctica 5 es la creación de un botón para cambiar el estilo de la fuente de la página web. Este botón se puede encontrar en la raíz de la web “<http://localhost:5000>”.

## Análisis de RSS con SAX y ETREE

El objetivo de este apartado es la lectura de un archivo XML para la interpretación de sus datos. Todo lo siguiente se ha realizado tanto con SAX parser en el archivo *sax\_parser.py* como ETREE parser en *etree\_parser.py*:

- Contar el número de noticias.
- Contar el número de imágenes.
- Búsqueda de un término concreto en las noticias (el resultado lo guarda en un archivo html).
- Descarga de las imágenes del feed.

Posteriormente se ha integrado el de SAX en la dirección "<http://localhost:5000/rss>" para que el usuario, mediante un buscador, pueda encontrar las noticias deseadas y con un formato correcto para su visualización.

## Gráficos con Highcharts

Con los datos que tenemos actualmente de todos los restaurantes podemos crear diferentes gráficos estadísticos para poder valorar y obtener aún más información relevante. Se han implementado dos gráficos que podremos ver en la dirección "<http://localhost:5000/restaurants/statistics>":

- Cuisine treemap: Mapa de áreas en función de la cantidad de restaurantes de un tipo de cocina específico.
- Circle Borough: Gráfico circular donde se representan las zonas donde se encuentran los restaurantes.

## Google Maps API

Para poder acceder a la API de Google Maps es necesario tener una cuenta registrada y dicha API activa y obtener los tokens para poder validar la conexión. Teniendo esto podremos mostrar los restaurantes en los mapas con la información de las coordenadas que tenemos.

A continuación se va a explicar en qué ocasiones y para qué se ha accedido a esta API:

- Paginador AJAX: En cada página del paginador aparecen 10 restaurantes diferentes y sus localizaciones en el mapa mediante marcadores; se obtienen las coordenadas de la base de datos y se representan.
- Añadir nuevo restaurante: Cuando queremos añadir un nuevo restaurante es difícil saber las coordenadas X e Y que corresponden a la dirección por lo que se ha implementado, como se ha explicado anteriormente, una funcionalidad para que dado el nombre de una calle, se acceda a la API de Google Maps y devuelva las coordenadas X e Y mostrando el mapa con un marcador en dicha calle.

- Editar restaurante existente: Aquí se ha implementado lo mismo que en el punto anterior.
- Mapa de localización global: Esta funcionalidad se puede encontrar en la dirección "<http://localhost:5000/restaurants/maps>". Debido a que puede darse el caso de que se inserten muchos restaurantes de la misma zona así como que la base de datos pueda ser muy grande (y la visualización de todos los restaurantes sea ineficiente), se quiso implementar un algoritmo para mostrar el mapa del mundo con los marcadores de los restaurantes en éste de forma que no hubiese más de dos restaurantes juntos dado un número X de radio o km cuadrado determinado. Se considera el mapa como una matriz de X filas e Y columnas que son prefijadas con antelación, donde en cada posición (X,Y) solo existe 1 posible restaurante.

## Twitter API

Para acceder a la API de Twitter ocurre lo mismo que con la de Google Maps, hace falta una cuenta y una serie de claves, secret key y token para poder validar la conexión a ésta.

Cuando queremos buscar un restaurante en "<http://localhost:5000/restaurants>" se realiza una consulta en la base de datos buscando el restaurante que se llama como la cadena que hemos introducido y una vez encontrado, al ser esta consulta del tipo *GET*, pasaremos a una nueva página que será "<http://localhost:5000/restaurants/search?name=cadena>".

En dicha página se mostrarán los datos del restaurante y también los últimos tweets que se han realizado sobre él, y también podremos seguir al autor del tweet, responder, hacer retweet o darle a me gusta.