

Práctica 10

Patricia Córdoba Hidalgo

1. Objetivos

Mi objetivo en esta práctica es añadir las siguientes funcionalidades a la página creada en la Práctica 5:

- Añadir una cartelera con una serie de películas recomendadas, a la que se accederá mediante el botón **Películas recomendadas** situado en la barra de navegación.
- Añadir gráficas utilizando Highcharts. Podemos visualizarlas pulsando en la sección **Estadísticas** de la barra de navegación.

2. Desarrollo

2.1. Cartelera

El fichero `html` donde se despliega esta funcionalidad es *recomendadas.html*, que extiende del fichero *p4.html*, el fichero base. Este fichero contiene en el bloque *lista* el código para crear una sección donde se encontrará tanto la imagen de una de las películas recomendadas como dos enlaces a cada lado de esta imagen, uno a su izquierda para ver al poster de la película anterior y otro a su derecha para ver el de la siguiente.

La imagen por defecto incluida en esta sección es el poster de la película *West Side Story*, y se cambiará al pulsar los enlaces de **Previa** y **Siguiente**. Los códigos de las funciones que se ejecutan al pulsar en dichos enlaces, `prev()` y `next()`, se encuentran en el fichero *src.js*, en la carpeta *static*.

Para que se ejecuten las funciones definidas en el fichero *src.js* hay que incluir la siguiente línea en *p4.html*:

```
<script src='url_for('static', filename='src.js')'></script>
```

En este fichero, tras el código para cambiar al modo nocturno, programé las funciones que permiten deslizar las imágenes de la cartelera. Primero declaré el vector `images`, que contiene la dirección donde se encuentran los pósters de todas las películas que forman la cartelera, y la variable `num`, que señala cual de las imágenes del vector se está mostrando en ese momento. Las funciones `prev()` y `next()` actualizan el valor de `num`, decrementando e incrementando su valor módulo `images.length` respectivamente, para que se muestre la imagen del vector `images` situada en esa posición.

2.2. Gráficas

Primero instalamos Highcharts añadiendo en la cabecera de *p4.html* la línea:

```
<script src='https://code.highcharts.com/highcharts.js'></script>
```

como indica en [la documentación](#).

En el fichero *graficas.html* creamos las secciones que albergarán los dos gráficos de la página e introducimos el código de éstas. El código de estas gráficas se ejecuta cuando el documento HTML ha sido completamente cargado y parseado.

La primera gráfica la implementé basándome en el código que aparece en <https://www.highcharts.com/docs/getting-started/your-first-chart>. Es un gráfico de barras que representa el número de películas de la base de datos que se estrenó en cada década.

La segunda gráfica representa el número de películas estrenadas en cada uno de los años que conforman la década con más películas. Este gráfico es un gráfico semicircular, cuya implementación está inspirada en el código que aparece en <https://www.highcharts.com/demo/pie-semi-circle>.

Al final del fichero `src.js` está el código donde se declara el estilo de las gráficas. En el enlace <https://www.highcharts.com/docs/getting-started/how-to-set-options> encontramos un ejemplo del uso de `setOptions`, el método que utilicé para especificar el estilo de mis gráficas, en particular los colores usados, el .

La información que se representa en las gráficas se recopila en la función `graficas()` del fichero `app.py`, la última función definida en este fichero. Los datos que necesitamos para representar la primera gráfica son el vector con las décadas en las que se estrenaron las películas de nuestra base de datos (`decadas`) y el número de películas estrenadas en cada década (`pelis_decada`). El primer vector conforma el eje vertical de la gráfica y el segundo especifica la longitud de cada una de las barras.

Los datos que necesitamos para visualizar la segunda gráfica es el vector con los años que componen la década con más películas estrenadas (`anios`) y el total de películas de cada uno de estos años (`pelis_decada_oro`). El primer vector guarda las etiquetas de los sectores circulares y el segundo se usará para concretar su área.

3. Alternativas

4. Problemas y sus soluciones

Gracias a la claridad y los ejemplos de la documentación, no tuve apenas problemas durante el desarrollo de mi práctica. El único problema que tuve fue calculando el año máximo y mínimo en los uqe hay películas guardadas. Para calcular el año máximo y mínimo usé inicialmente la función `sort` como nos indica [la documentación](#):

```
max_year = db.video_movies.find().sort({'year':-1})[0]['year']
```

Este código producía el siguiente error: *if no direction is specified, key_or_list must be an instance of list*. Busqué el error en internet y encontré el enlace <https://stackoverflow.com/questions/10242149/using-sort-with-pymongo>, donde se explica que la manera correcta de usar `sort` es añadiéndole una lista como parámetro y no un diccionario, dado que el orden de los argumentos importa. Por tanto, la versión final del código es:

```
max_year = db.video_movies.find().sort([('year',-1)])[0]['year']
```