DWEC - Tarea 06

# Enunciado:

Después de la tarea del Whac-a-Mole vamos a realizar una tarea que a muchos os gustará. Una página para consultar la web/base de datos sobre star wars llamada SWAPI.

¿Cómo lo vamos a implementar? Pues en un solo archivo html y con un sólo js.

¿Cómo se realiza una llamada para obtener un JSON de ciudad de un país? Pues realizas una llamada a AJAX con GET y JSON con la API gratuita de:

* <https://swapi.co/>

¿Ventajas? Pues no tenemos que crear nada con PHP siendo el contenido de la página web dinámico. Se simplifica la  tarea y además podemos hacer algo muy interesante (y realmente frikki).

Esta API te puede devolver JSON y wookie.  En esta práctica nos centraremos en JSON. Es más sencillo de entender.

¿Qué cosas se van a realizar? Veamos un algoritmo de uso de la aplicación que va a haceros comprender que es lo que quiero:

1. El usuario va a poder navegar entre las 7 películas
   1. Listado en  de las películas navegable. O sea, con un botón para mostrar la siguiente película.
   2. Exacta. Se busca una película solamente. Nada más.
   3. En los dos caso debe mostrarse el título de la película, el director, el productor, el día del estreno en EEUU, número de personajes.
2. Personaje. Mostrar la descripción y los datos de un personaje.  Si el resultado tiene más de un personaje mostrar solo el primero. De los datos que tenemos que tener en cuenta son el nombre, sexo, año de nacimiento, peso y altura.

En el punto 3 se deben mostrar los registros seguidos uno de otro en la misma página.

En esta tarea tenéis una parte opcional (para sacar más nota con un máximo de 13). Básicamente lo que se pide es ampliar  el contenido a mostrar en nuestra página.

* Pensar una forma para que las películas aparezcan ordenadas (o sea piensa como solucionar ese problema).  (1 punto).
* Mostrar el listado de personajes con ese nombre. Por ejemplo poder navegar por todos los personajes llamados skywalker. (0,5 puntos).
* Hacer que  cuando muestres una película aparezca abajo un listado de personajes. (1 punto).
* Realizar una búsqueda general que se encargue de buscar lo pedido entre los distintos listados (1 puntos).
* Realizar una de las búsquedas (la que quieras) con el objeto xmlhttprequest (1 punto).
* Hacer la consulta en wookie, ;) (0,25 puntos).

Observaciones:

* Se debe utilizar jQuery para la práctica, aunque sólo sean cuestiones básicas.  Ya verás como te soluciona la vida. Sino quieres que sepas que puedes sacar hasta un 9, pero no podrás realizar las partes extra (o podrás pero yo no te las puntuaré).
* Para obtener el valor de un input del formulario con**id="titulo"** se puede utilizar, por ejemplo, **$("#titulo").val()** que te devuelve el valor actual (no el predefinido) de ese campo.
* Si creas una función para gestionar un evento en jQuery puede llevar un parámetro evento (es opcional) pero muy interesante para abortar la propagación del evento. Podrías usar por ejemplo:

$("button").click(function (evt) {

//// Código

evt.preventDefault();

});

* Para cambiar el CSS con jQuery podéis utilizar:

$('#elemento').css('propiedad-css', 'valor a asignar');

* Hay tareas opcionales. Te pueden servir para asegurarte un 10 si metes la pata en algo, pero no es necesario para sacar un apto. Las he puesto ya que se acercan los exámenes os puede servir para comprobar vuestro conocimiento (y para compensar algo de inexperiencia en las primeras tareas).

## GUIA RAPIDA:

Aquí quiero ayudaros a comprender un poco más como funciona esta API realmente fácil. Cómo está en inglés os recomiendo un diccionario y que os leáis esta página y la que se que se encuentra oficialmente la documentación.

Veamos por ejemplo el resultado de buscar la película la amenaza fantasma (en inglés y la segunda peor de toda la saga).

* [https://swapi.co/api/films/?search=the phantom menace?api=json](https://swapi.co/api/films/?search=the%20phantom%20menace?api=json)

En la petición tenemos lo siguientes campos:

* De la ruta */api/* cuelgan las posibles búsquedas.  Nosotros vamos a necesitar dos para el desarrollo una *films*  para las películas y otra llamada *people* que nos va a permitir buscar personajes.
* *search*nos va a buscar por el título o el nombre del personaje.
* *api* Sirve para indicar si queremos una petición ajax json o una en wookie.

El resultado de nuestra petición será sin indentar similar al siguiente (indentado):

{

"**count**": 1,

"next": null,

"previous": null,

"**results**": [

{

"**title**": "The Phantom Menace",

"**episode\_id**": 1,

"opening\_crawl": "Turmoil has engulfed the\r\nGalactic Republic. The taxation\r\nof trade routes to outlying star\r\nsystems is in dispute.\r\n\r\nHoping to resolve the matter\r\nwith a blockade of deadly\r\nbattleships, the greedy Trade\r\nFederation has stopped all\r\nshipping to the small planet\r\nof Naboo.\r\n\r\nWhile the Congress of the\r\nRepublic endlessly debates\r\nthis alarming chain of events,\r\nthe Supreme Chancellor has\r\nsecretly dispatched two Jedi\r\nKnights, the guardians of\r\npeace and justice in the\r\ngalaxy, to settle the conflict....",

"**director**": "George Lucas",

"**producer**": "Rick McCallum",

"**release\_date**": "1999-05-19",

"characters": [

"<http://swapi.co/api/people/2/>",

"<http://swapi.co/api/people/3/>",

"<http://swapi.co/api/people/10/>",

"<http://swapi.co/api/people/11/>",

"<http://swapi.co/api/people/16/>",

"<http://swapi.co/api/people/20/>",

"<http://swapi.co/api/species/21/>",

"<http://swapi.co/api/species/22/>",

"<http://swapi.co/api/species/23/>",

"<http://swapi.co/api/species/24/>",

"<http://swapi.co/api/species/25/>",

"<http://swapi.co/api/species/26/>",

"<http://swapi.co/api/species/27/>"

],

"created": "2014-12-19T16:52:55.740000Z",

"edited": "2015-04-11T09:45:18.689301Z",

"url": "<http://swapi.co/api/films/4/>"

}

]

}

¿Qué pasa si el título buscado no existe?

{  
**"detail"**:"Not found"  
}

**count** va a indicar el número de registros que aparecen.  Si aparece 0 es que no ha habido resultados tampoco.

Pues si  **detail**  es "*Not found*" entonces es que se ha producido un fallo por no existir ese título.

Hay muchas formas para buscar personajes, pero nosotros nos vamos a realizar una consulta similar a esta:

* <https://swapi.co/api/people/?format=json&search=skywalker>

{

**"count"**: 3,

"next": null,

"previous": null,

"**results**": [

{

"**name**": "Luke Skywalker",

"**height**": "172",

"**mass**": "77",

"hair\_color": "blond",

"skin\_color": "fair",

"eye\_color": "blue",

"**birth\_year**": "19BBY",

"**gender**": "male",

"homeworld": "<http://swapi.co/api/planets/1/>",

"films": [

"<http://swapi.co/api/films/6/>",

"<http://swapi.co/api/films/3/>",

"<http://swapi.co/api/films/2/>",

"<http://swapi.co/api/films/1/>",

"<http://swapi.co/api/films/7/>"

],

"species": [

"<http://swapi.co/api/species/1/>"

],

"vehicles": [

"<http://swapi.co/api/vehicles/14/>",

"<http://swapi.co/api/vehicles/30/>"

],

"starships": [

"<http://swapi.co/api/starships/12/>",

"<http://swapi.co/api/starships/22/>"

],

"created": "2014-12-09T13:50:51.644000Z",

"edited": "2014-12-20T21:17:56.891000Z",

"url": "<http://swapi.co/api/people/1/>"

},

{

"name": "Anakin Skywalker",

"height": "188",

"mass": "84",

"hair\_color": "blond",

"skin\_color": "fair",

"eye\_color": "blue",

"birth\_year": "41.9BBY",

"gender": "male",

"homeworld": "<http://swapi.co/api/planets/1/>",

"films": [

"<http://swapi.co/api/films/5/>",

"<http://swapi.co/api/films/4/>",

"<http://swapi.co/api/films/6/>"

],

"species": [

"<http://swapi.co/api/species/1/>"

],

"vehicles": [

"<http://swapi.co/api/vehicles/44/>",

"<http://swapi.co/api/vehicles/46/>"

],

"starships": [

"<http://swapi.co/api/starships/59/>",

"<http://swapi.co/api/starships/65/>",

"<http://swapi.co/api/starships/39/>"

],

"created": "2014-12-10T16:20:44.310000Z",

"edited": "2014-12-20T21:17:50.327000Z",

"url": "<http://swapi.co/api/people/11/>"

},

{

"name": "Shmi Skywalker",

"height": "163",

"mass": "unknown",

"hair\_color": "black",

"skin\_color": "fair",

"eye\_color": "brown",

"birth\_year": "72BBY",

"gender": "female",

"homeworld": "<http://swapi.co/api/planets/1/>",

"films": [

"<http://swapi.co/api/films/5/>",

"<http://swapi.co/api/films/4/>"

],

"species": [

"<http://swapi.co/api/species/1/>"

],

"vehicles": [],

"starships": [],

"created": "2014-12-19T17:57:41.191000Z",

"edited": "2014-12-20T21:17:50.401000Z",

"url": "<http://swapi.co/api/people/43/>"

}

]

}

Si os fijáis **count** va a dar el número de registros y **results** va a ser un array con un número de elementos indicados en **count**.  Así que el valor resultado del primer personaje sería algo así**datos\_devueltos.results[0]**.

**Para saber más de la API**

<https://swapi.co/documentation#search>

# Criterios de corrección:

Los puntos que tiene esta tarea están asignados de la siguiente forma:

* Fichero HTML con el formularios. **1,5 puntos.**
* Uso de jQuery (para cross-browser): **2 puntos**.
* Peticiones AJAX con jQuery y datos en formato JSON para buscar un título de película. **1,5 puntos**
* Peticiones AJAX con jQuery y datos en formato JSON para listar las películas con el botón siguiente. **2,5 puntos**
* Peticiones AJAX con jQuery y datos en formato JSON para búsqueda de personajes. **1,5 puntos**
* Presentación de los resultados. Uso de imágenes para indicar que se está utilizando AJAX (la típica imagen de cargar que es un gif animado) :**0,5 puntos**.
* Claridad y presentación del código:**0,5 puntos.**

 Opcional (si haces lo anterior ya tienes un 10, pero podrías querer sacar**un 13 máximo**). Y si no te ha salido algo bien, pero aún quieres sacar un 10, puedes realizar cualquiera de los siguientes puntos:

* Pensar una forma para que las películas aparezcan ordenadas (o sea piensa como solucionar ese problema).  **1 punto**
* Mostrar el listado de personajes con ese nombre. Por ejemplo poder navegar por todos los personajes llamados *skywalker*. **0,5 puntos**.
* Hacer que  cuando muestres una película aparezca abajo un listado de personajes. **1 punto**.
* Realizar una búsqueda general que se encargue de buscar lo pedido entre los distintos listados. **1 puntos**.
* Realizar una de las búsquedas (la que quieras) con el objeto xmlhttprequest. **1 punto**.
* Hacer la consulta en wookie 0,25 puntos.

# Recursos necesarios:

* Ordenador personal con conexión a Internet.
* Uso de API gratuita [swapi.co](https://swapi.co/)
* Documentación de [swapi.com](https://swapi.co/documentation#search)
* Editor web para teclear el código de la aplicación.
* jQuery para realizar la aplicación cross-browser.

# Consejos y recomendaciones:

Revisa los ejemplos y vídeos de la unidad que te pueden ayudar bastante para hacer la tarea.

# Corrección: