

PRÁCTICA 6.03 Asincronismo en React

Normas de entrega

- En cuanto al **código**:
 - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
 - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
 - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
 - el nombre de los componentes debe comenzar con letra **mayúscula**.
 - la **componentización** debe ser lógica y estar adaptada a las necesidades de la aplicación,
 - todos los formularios en **React** deben ser del tipo **controlados** (a través del **estado**). En caso contrario debe ser debidamente justificado,
 - toda comunicación con el exterior del **DOM** debe ser realizada con funciones asíncronas que dispongan de control de errores (**try{} catch{}**),
 - se debe implementar el uso de **async/await** para la comunicación asíncrona.
- En cuanto a la **entrega** de los archivos que componen los ejercicios de **React**:
 - entrega **la práctica** en un sólo proyecto (el nombre a tu discreción),
 - los componentes creados deben estar separados en carpetas (los creados en el Ejercicio1 dentro de una carpeta denominada **Ejercicio1**),
 - el código contendrá ejemplos de ejecución, si procede,
 - comprime la carpeta **src** junto con el fichero **package.json** en un fichero **ZIP**, y
- sube a **Aules** el fichero comprimido.

Ejercicio 1 - Enciclopedia de *Star Wars* versión en **React**.

Parte 1

Migra la enciclopedia básica realizada en **JavaScript Vanilla** a **React** creando los componentes que estimes oportunos (esta será la última migración, prometido).

Para todo esto deberás **dividir el código en funciones/componentes** que accedan a la **API** (traigan datos del exterior) y que pinten esos datos en el **DOM**. En ningún caso una función/componente realizará ambas tareas.

Para toda la comunicación con la **API REST** utiliza **async/await**.

Parte 2

Añade, en los datos de cada película, un listado de **diez** (sólo diez) de los protagonistas de cada una de ellas (se debe mostrar su nombre completo). Al pinchar sobre el nombre de cada intérprete se mostrará información detallada como su género, su altura, su peso, su color de pelo, el color de sus ojos y una imagen (opcional).

Utiliza la llamada en **conjunto** de **promises** en la solicitud de los intérpretes.

Muestra la información donde y como creas oportuno, pero recuerda, debidamente formateada.