

Módulo 3: Ejercicio de autoevaluación intermedia

El ejercicio consiste en desarrollar una página web sencilla con React.

Antes de empezar, tenéis que crear un nuevo repositorio en GitHub desde vuestra cuenta (así se quedará guardado en vuestro repositorio como un proyecto más 😊). Una vez creado, lo clonaremos en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.

Listado de países

Este ejercicio consiste en desarrollar una aplicación web utilizando React que permita a las usuarias visualizar información sobre diferentes países, añadir nuevos países y filtrar la lista de países según sus necesidades.

The screenshot shows the 'Country Info App' interface. At the top, there's a green header bar with the title 'Country Info App' and a sub-instruction: 'Explore information about countries, capitals, and flags. Add new countries and filter through the list!'. Below the header, there are two sections: 'Filters' and 'Add Country'. The 'Filters' section contains two input fields: 'By Country' (with a placeholder 'Spain...') and 'By Continent' (with a dropdown menu showing 'All'). To the right of these is the 'Add Country' button. The main area displays a grid of 16 country cards, each containing a flag icon, the country name, its capital, and its continent. Each card also has a red 'X' icon in the top right corner. The countries listed are: French Polynesia (Papeeté, Oceania), Saint Martin (Marigot, North America), Bolivarian Republic of Venezuela (Caracas, South America), Réunion Island (Saint-Denis, Africa), Republic of El Salvador (San Salvador, North America), Commonwealth of Dominica (Roseau, North America), Gibraltar (Gibraltar, Europe), Republic of Kenya (Nairobi, Africa), Federative Republic of Brazil (Brasília, South America), Republic of the Maldives (Malé, Asia), United States of America (Washington, D.C., North America), Cook Islands (Avarua, Oceania), Niue (Alofi, Oceania), Republic of Seychelles (Victoria, Africa), Central African Republic (Bangui, Africa), and Tokelau (Fakaofo, Oceania).

A continuación detallamos lo que queremos hacer.

1º Pintar el listado de países

Country Info App

Explore information about countries, capitals, and flags. Add new countries and filter through the list!

 French Polynesia Papeete Oceania	 Saint Martin Marigot North America	 Bolivarian Republic of Venezuela Caracas South America	 Réunion Island Saint-Denis Africa
 Republic of El Salvador San Salvador North America	 Commonwealth of Dominica Roseau North America	 Gibraltar Gibraltar Europe	 Republic of Kenya Nairobi Africa
 Federative Republic of Brazil Brasília South America	 Republic of the Maldives Malé Asia	 United States of America Washington, D.C. North America	 Cook Islands Avarua Oceania
 Niue Alofi Oceania	 Republic of Seychelles Victoria Africa	 Central African Republic Bangui Africa	 Tokelau Fakaofa Oceania

Para empezar el ejercicio queremos:

1. Pintar una cabecera con un título que ponga **Country Info App**.
2. Pintar el listado de países inicial. Para ello os facilitamos los datos en la siguiente API: [este enlace](#)

Lo que se desea es que, usando la información del API, crees un array de objetos en tu aplicación para poder mostrar información de cada país. Entre esa información, debes mostrar el nombre y la bandera (pero puedes mostrar otra información si quieras, como la capital o el continente al que pertenece el país).

NOTA: No tienes que hacer un fetch pero... os dejamos la dirección del API para que más adelante podáis completar el ejercicio.

2º Filtrar por nombre el listado de países

Debe haber una funcionalidad que permita a las usuarias filtrar la lista de países en tiempo real, mostrando únicamente aquellos países cuyos nombres coincidan con la cadena de texto proporcionada por la usuaria en un campo de búsqueda.

Los pasos a seguir para filtrar por nombre del país son:

1. Añade un input sobre el listado de países donde la usuaria pueda escribir texto.
2. Añade la funcionalidad para que, cuando la usuaria escriba en ese <input>, el listado de países se repinte mostrando solo las que incluyen el texto que ha escrito la usuaria (aplicando también el filtrado por continente, si ya lo tuvieras implementado).

Algunas pistas para pensar como usar ese <input>:

- ¿Vas a guardar los datos del select en algún sitio? ¿y del input?

- ¿Qué **tipo** de datos vas a guardar?
- ¿Qué **valor inicial** le vas a poner?

Y siempre que filtramos algo debemos hacernos las preguntas:

- ¿Qué datos de cada país tienes que comparar con la opción seleccionada por la usuaria, para saber si un país cumple o no con el filtro?
- ¿Debo guardar los países filtrados en algún sitio o puedo calcularlos al vuelo?

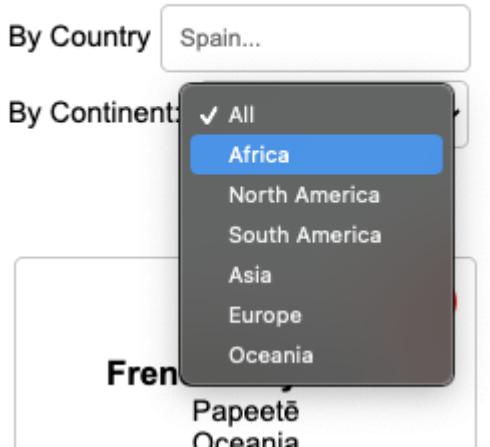
3º Dividir tu aplicación en componentes

Organiza el código de tu aplicación web en distintos componentes. **Hemos pensado que los componentes indispensables son: App, Header, CountryList, CountryItem y Filters** (recuerda esto solo es una recomendación, puedes poner otros nombres, organizarlos en carpetas, hacer menos componentes o inclusive mas 😊)

4º Filtrar por continente el listado de países (BONUS)

Adicionalmente, puedes añadir una lista desplegable para que la usuaria pueda filtrar por el continente seleccionado en esa lista:

Filters



1. Añade un **select** a la cabecera que tenga las siguientes opciones:

- All
- Africa
- North America
- South America
- Europe
- Asia
- Oceania

2. Añade la funcionalidad para que cuando la usuaria cambie este select el listado de países se repinte mostrando solo los que coincidan con la opción seleccionada.

Volvemos a las pistas, esta vez sobre ese **<select>**:

- ¿Vas a guardar los datos del select en algún sitio?
- ¿Qué **tipo** de datos vas a guardar?
- ¿Qué **valor inicial** le vas a poner?

Y algunas preguntas que debemos hacernos sobre la funcionalidad que queremos proporcionar cuando filtramos usando un `<select>`:

- ¿Hay alguna opción del select con la que deba mostrar todos los países?
- ¿Hay alguna opción del select con la que no deba mostrar ningún país?
- ¿Debo poner textos distintos entre las etiquetas `<option>` y el `value=""` de esa `<option>`?

5º Añadir una nuevo país (BONUS)

The image shows a user interface for adding a new country. At the top, it says "Add Country". Below that are four input fields: "Country Name", "Capital", "Flag Icon", and "Continente". To the right of these fields is a green button labeled "Add Country".

A continuación queremos añadir un nuevo país. Para ello:

1. Crear un formulario con los campos:
 - Nombre.
 - Capital
 - Icono de la bandera
 - Continente
 - Un botón para **Añadir el nuevo país**.
2. Cuando la usuaria pulse en el botón hay que añadir el país al listado de países para que este aparezca en el listado.

Te queremos dar unas pistas en forma de preguntas. Antes de que la usuaria pulse en el botón de Añadir debes pensar:

- ¿Vas a guardar los datos del formulario en algún sitio?
- ¿Qué **tipo** de datos vas a guardar?
- ¿Qué **valores iniciales** tienen estos datos?
- ¿Cómo vas a añadir los datos del nuevo país a los datos de la aplicación?

6º Borrar países (BONUS)

Para finalizar la gestión de los países queremos crear la funcionalidad para borrarlos de uno en uno. Para ello hay que:

1. Añadir un botón de borrar a cada uno de los países.
2. Cuando la usuaria pulse en el botón hay que programar la funcionalidad para que este país desaparezca de pantalla.

Y ya que estamos también te damos unas pistas:

- ¿Recuerdas con qué **método funcional** podemos borrar un elemento de un array?
- Para borrar un elemento de un array debemos saber qué elemento queremos borrar. ¿Se te ocurre alguna forma de identificar de manera única cada uno de los países pintados en el HTML? ¿Se te ocurre alguna forma de recuperar ese identificador cuando la usuaria pulse en el botón de borrar?

7º Maqueta a tu gusto (BONUS)

No te vamos a decir cómo debes poner bonita tu web porque tienes más estilo y flow que nosotras.

Entrega

En este enunciado os proponemos dos ejercicios básicos y algunos ejercicios avanzados marcados como **BONUS**. Hemos calculado que hacer los 5 ejercicios durante el tiempo de la clase es casi imposible, aspiramos a que únicamente hagas **los dos primeros ejercicios**.

Al turrón^