

HW13 | React Hooks - Integration

DURACIÓN ESTIMADA

2 horas

RICK AND MORTY APP

INTRODUCCIÓN

En esta homework crearemos dos cosas que harán más completa nuestra aplicación 😊.

- Haremos un **filtrado** para nuestros personajes favoritos. Vamos a filtrar todos los personajes según su género: **Male**, **Female**, **Genderless** y **unknown**.
- Por otro lado haremos un **ordenamiento** para nuestros personajes favoritos. Vamos a ordenar todos los personajes por su **id** (de mayor a menor y viceversa).

INSTRUCCIONES

EJERCICIO 1 | Actions

Dirígete al archivo **actions** y crea las siguientes funciones:

1. **filterCards**: esta función recibe por parámetro un **gender**. Debe retornar una action con el **type** igual a **"FILTER"** y el payload será igual al parámetro recibido.
2. **orderCards**: esta función recibe por parámetro un **orden** (será: **A**: ascendente o **D**: descendente). Debe retornar una action con el **type** igual a **"ORDER"** y el payload será igual al parámetro recibido.

EJERCICIO 2 | Reducer

Dirígete al archivo **reducer** y sigue estos pasos:

1. En tu estado inicial crea una nueva propiedad llamada **allCharacters** que debe ser igual a un arreglo vacío.
2. Modificaremos el caso **ADD_FAV** de la siguiente manera:
 - Dentro de la copia de tu estado global, reemplaza la propiedad **myFavorites** por **allCharacters**.
 - Cuando retournes tu estado, agrega la propiedad **allCharacters** que también sea igual a la copia en la que agregaste el nuevo personaje.

3. Crea un nuevo caso con el nombre "**FILTER**". Aquí debes crear una copia de tu estado global **allCharacters**. A partir de esta copia filtra todos aquellos personajes que tengan el mismo género que recibes por payload. Finalmente retorna una copia de tu estado, pero que la propiedad **myFavorites** sea igual a este filtrado.
4. Crea un nuevo caso con el nombre "**ORDER**". Aquí vamos a ordenar nuestros personajes favoritos de forma ascendente y descendente. Para esto:
 - Crea una copia de tu estado global **allCharacters**.
 - Utiliza el método **sort** para ordenar tus personajes de acuerdo a su **id**.
 - Si el payload es igual a "**A**", los personajes deben ordenarse de menor a mayor.
 - Si el payload es igual a "**D**", los personajes deben ordenarse de mayor a menor.
 - Finalmente retorna tu estado global y en la propiedad **myFavorites** guarda el ordenamiento que hiciste.

[NOTA]: investiga en la web cómo funciona el método **sort**.



EJERCICIO 3 | Filtro & Ordenamiento

Dirígete a tu componente **Favorites**. Dentro de él deberás:

1. Crea una etiqueta **select**. Dentro de este selector:
 - Crea una etiqueta **option** con el atributo **value** igual a "**A**" (ascendente).
 - Crea una etiqueta **option** con el atributo **value** igual a "**D**" (descendente).

```
<option value="Ascendente">Ascendente</option>
```

2. Crea una segunda etiqueta **select**. Dentro de este selector deberás:
 - Crear 4 etiquetas **option**. Cada una con su atributo **value** igual a los siguientes valores: **Male**, **Female**, **Genderless** y **unknown**.

```
<option value="Male">Male</option>
```

[NOTA]: ten en cuenta que la propiedad **unknown** debe escribirse en minúsculas, ya que esa es la forma como proviene de la API.

3. Importa las actions que creaste en esta homework y el hook **useDispatch**.
4. Crea una función llamada **handleOrder**. En su interior solo debe despachar la action **orderCards** pasándole como argumento **e.target.value**.
5. Crea una función llamada **handleFilter**. En su interior solo debe despachar la action **filterCards** pasándole como argumento **e.target.value**.

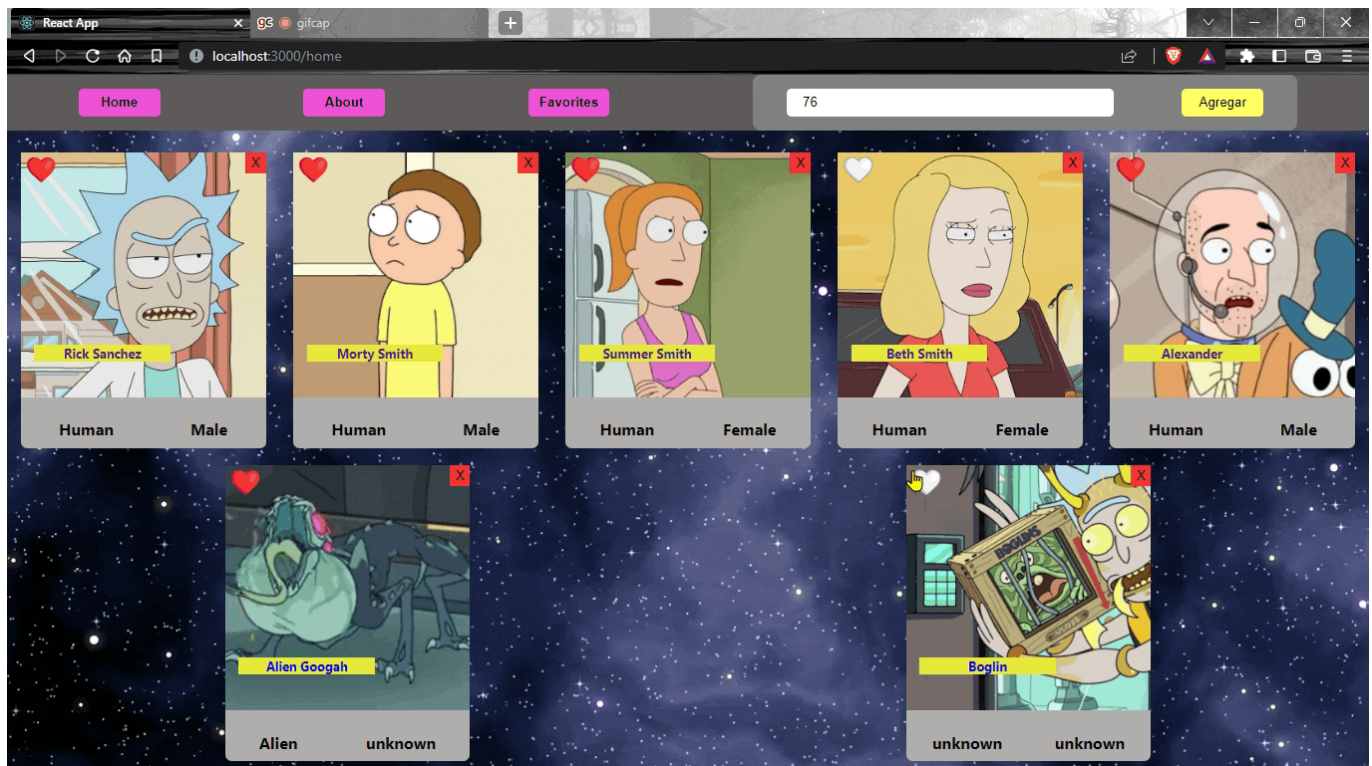
6. Agrega el atributo **onChange** a las etiquetas **select** pasándoles las funciones correspondientes a cada una.

EJERCICIO 4 | Forzado de render

Ahora solo nos queda, en el componente **Favorites** crear un estado local que se llama **aux** e inicialo en **false**.

Una vez creado, dentro del **handleOrder** setea este estado en su valor opuesto.

A esta altura, tu filtro y ordenamiento debería estar funcionando de la siguiente manera:



Extra Credit

Agrega una opción adicional en el select del filtro para que muestre todos los personajes. Desarrolla la lógica para que ello ocurra.