

Universidad Tecnológica Nacional

Facultad Regional Rosario

Tecnicatura Universitaria de la Programación



Laboratorio de Computación III

Unidad P2.1

Manejo de estado en React

Práctica principal

Crear un componente llamado Beers y dentro de él colocar este arreglo de objetos:
(simulando una conexión a una api donde obtendremos estos datos)

```
const beers = [  
  {  
    id: 1,  
    beerName: "American",  
    beerStyle: "IPA",  
    price: 3,  
    available: true,  
  },  
  {  
    id: 2,  
    beerName: "Argenta",  
    beerStyle: "IPA",  
    price: 4,  
    available: false,  
  },  
  {  
    id: 3,  
    beerName: "Irish",  
    beerStyle: "Red",  
    price: 4,  
    available: true,  
  },  
  {  
    id: 4,  
    beerName: "Scotish",  
    beerStyle: "Red",  
    price: 3,  
    available: true,  
  },  
  {  
    id: 5,  
    beerName: "DeEssoCiTratta",  
    beerStyle: "APA",  
    price: 3,  
    available: true,  
  },  
  {  
    id: 6,  
    beerName: "Santa APA",  
    beerStyle: "APA",  
    price: 3,  
    available: true,  
  },  
]
```

```

    },
    {
      id: 7,
      beerName: "German",
      beerStyle: "Pilsen",
      price: 1,
      available: true,
    },
    {
      id: 8,
      beerName: "London Porter",
      beerStyle: "Porter",
      price: 2,
      available: false,
    },
    {
      id: 9,
      beerName: "Scotish ALE",
      beerStyle: "Red",
      price: 5,
      available: false,
    },
  ],
];

```

- 1 - Crear un componente llamado Change Dollar que nos permita actualizar el dólar y automáticamente se actualice el precio de cada cerveza en el listado de las mismas.
- 2 - Crear un state para que se muestre o se esconda el componente para cambiar el valor del dólar.
- 3 - Crear un componente NewBeer que tenga un formulario para agregar una nueva cerveza.

Práctica adicional

1. Deseamos crear dos componentes para el ingreso de la cantidad de patas que va a tener una mesa. Los componentes se denominan *TableForm* y *TableInput*, y necesitan cumplir los siguientes requisitos:

TableInput:

- Este componente solo retornará un input en donde el usuario ingresará la cantidad de patas que va a tener la mesa.

- No es obligatorio pero se considerará puntos extra la buena utilización del *two-way data binding*

TableForm:

- Debe tener un mensaje *h2* que indique “Ingrese la cantidad de patas que tendrá la mesa”
- Debe importar y utilizar el componente *TableInput*.
- Debe tener un elemento *p* que:
 - En el caso de que el usuario escriba el valor 4 en el input, mostrará el mensaje “Mesa correcta”.
 - Caso que no sea 4 el valor del input, mostrará “Mesa inestable”

Generales:

- El componente debe ser funcional.
 - No es necesario ningún tipo de estilizado en la respuesta.
 - Solo escribir dos componentes
 - El único *hook* necesario para la resolución del ejercicio es *useState*.
2. Deseamos saber qué tipo de envío desea el usuario para su encomienda. Para ello, vamos a construir dos componentes que cumplan los siguientes requisitos:

SendingRadio

- Este componente devolverá tres *inputs* del tipo radio (con sus respectivos *labels*), donde el cliente pueda seleccionar que tipo de envío prefiere (“Retiro en el local”, “Envío express”, “Envío común”).
- Un *div* que recubra a los 3 *inputs* + *labels*.

SendingForm

- Debe tener un elemento *h2* que diga “Seleccione el tipo de envío”
- Debe importar y utilizar el componente **SendingRadio**
- En el caso de que se haya elegido una opción:
 - Si la opción que se eligió es **Retiro en el local**, devolvemos un elemento *p* con el texto “No hay recargo”.
 - Si elige la opción **Envío común** o **Envío Express**, devolvemos un elemento *p* con el texto “Tendrá recargo”
- En el caso de que la opción aún no haya sido elegida, no mostramos el elemento *p*.

Generales

- El componente debe ser funcional.
- Solo escribir dos componentes
- El único *hook* necesario para la resolución del ejercicio es *useState*.

No es necesario la utilización de un elemento *form* ni de un botón *submit* (ni de ningún botón), ya que la función deseada para la resolución es un *onChange*, no un *onSubmit*

Fecha	Versionado actual	Autor	Observaciones
03/09/2023	1.0.0	Gabriel Golzman / Esteban Ferreyra	Primera versión