



Renderización dinámica de componentes



**Activen las cámaras los que puedan y
pasemos asistencia**





Inicio

{desafío}
latam_



*/** Crear componentes que realicen operaciones de creación, eliminación y listado a partir de una estructura de datos como un arreglo u objeto. **/*

**Objetivo de
la unidad**

`/* Utilizar un arreglo para mostrar información en componentes de React. */`

`/* Agregar un elemento a un arreglo utilizando un formulario */`

`/* Utilizar un Json para mostrar información en componentes. */`

`/* Hacer uso del estado en react para almacenar una lista de elementos que sean mostrados dinámicamente en componentes. */`

Objetivos

Activación de conceptos

Contesta la pregunta correctamente y gana un punto

Instrucciones:

- Se realizará una pregunta, el primero en escribir “YO” por el chat, dará su respuesta al resto de la clase.
- El docente validará la respuesta.
- En caso de que no sea correcta, dará la oportunidad a la segunda persona que dijo “Yo”.
- Cada estudiante podrá participar un máximo de 2 veces.
- Al final, el/la docente indicará el 1º, 2º y 3º lugar.
- Esta actividad no es calificada, es solo una dinámica para recordar los conceptos clave para abordar esta sesión.





Activación de conceptos



¿Qué hook de React utilizamos para definir el valor inicial de un componente?



Activación de conceptos



¿Si necesitamos implementar un botón que al darle click muestre un mensaje en la consola del navegador, qué evento utilizarías?



Activación de conceptos



¿Qué tipos de datos podemos almacenar como estado inicial de un componente en React?



Activación de conceptos



Primer lugar:



Segundo lugar:



Tercer lugar:



Desarrollo

{desafío}
latam_



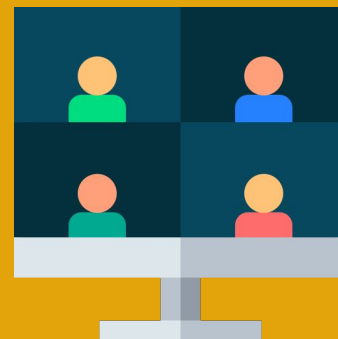
Ejercicio

En esta actividad vamos a renderizar un arreglo de JavaScript que almacene un conjunto de datos. Esto lo haremos usando JSX.

- Crear una aplicación con el nombre tienda-react.
- Limpiaremos el proyecto para trabajar solo con los archivos:
 - index.js
 - index.css
 - App.js
- Dentro del App.js vamos a definir un arreglo de Productos que contenga los siguientes strings:
 - Producto 1
 - Producto 2
 - Producto 3

{desafío}
latam_

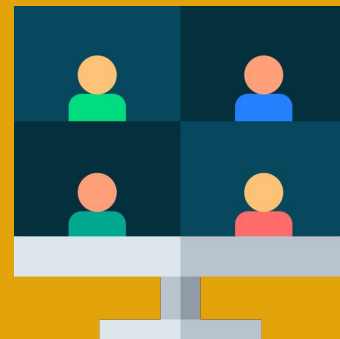
Demostración



Ejercicio

- Utilizaremos JSX para mostrar el arreglo en nuestro App.js.
- Implementaremos el uso de las etiqueta `` y `` para listar los elementos del arreglo.
- Aplicaremos el uso del método `.map()` para recorrer el arreglo.
- Visualizamos los cambios en el navegador y en el inspector de elementos.
- Aplicar el uso de la propiedad "key" al mostrar una lista de elementos en React

Demostración



Ejercicio

Practiquemos lo aprendido

1. En una aplicación nueva llamada pintar-colores.
2. Crea un arreglo llamado colores
3. Mostrar cada elemento del arreglo.
4. Opcional: Mostrarlo de su color respectivo

Ejercicio ¡Manos al teclado!



La propiedad key

“Las keys ayudan a React a identificar qué ítems han cambiado, son agregados, o son eliminados. Las keys deben ser dadas a los elementos dentro del array para darle a los elementos una identidad estable”

Fuente: react.org

Ejercicio

Realizamos el siguiente ejercicio guiado:

- Vamos a intencionar la comprobación del key value cuando intentamos renderizar una lista de elementos.
- Dentro del App.js quitemos la propiedad key cuando hacemos el recorrido de nuestro arreglo.
- Veamos el siguiente error que nos informa la consola del navegador.

Error de la consola:

```
✖ Warning: Each child in a list should have a unique "key" prop.  
Check the render method of `App`. See https://reactjs.org/link/warning-messages.  
    at ul  
    at App
```

Ejercicio ¡Manos al teclado!



La propiedad key

Entonces, este error de la propiedad key hace referencia de que cada elemento que se muestre de una lista debe tener un identificador único para que react pueda detectar cuando hayan cambios en dicho elemento.

/* Utilizar un arreglo para mostrar información en componentes de React. */ 

/* Agregar un elemento a un arreglo utilizando un formulario */

/* Construcción de componentes en React que importen datos a través de un arreglo de objetos de JavaScript. */

/* Creación dinámica de componentes a partir de una estructura de datos de tipo JSON. */

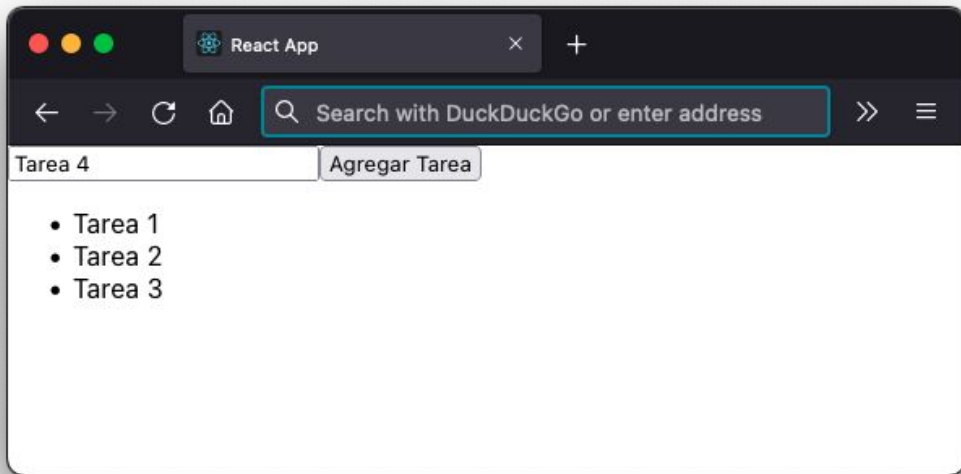
Objetivos

Agregar productos a una lista

¿Qué necesitamos?

Necesitaremos 3 elementos nuevos:

1. Un Input con un listener con el de evento onChange
2. Un formulario con un listener de evento onSubmit
3. Un botón de tipo submit



Agregar productos a una lista

Las componentes se vuelven a renderizar cuando el estado cambia

Para actualizar la lista necesitamos que el arreglo sea un estado, de esta forma solo con agregar un elemento nuevo la lista se actualizará. Este nuevo estado será el sustituto del arreglo tarea que teníamos en el código.

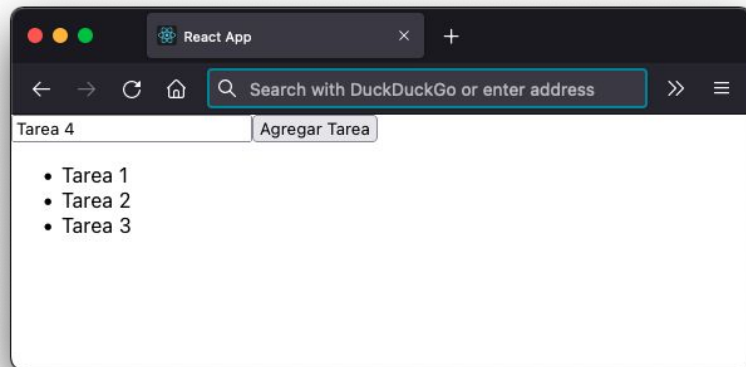
También necesitaremos ir guardando en un estado cada vez que se modifique el input, dado que este será el valor que guardaremos en el arreglo.

Agregar productos a una lista

1. Cada vez que modifique el input, actualizaremos el estado que guarda su información
2. Cada vez que se envíe el formulario actualizaremos el estado que guarda el arreglo.



Nota: Este ejercicio también es posible hacerlo desde el clic del botón en lugar del envío del formulario.



Agregar un elemento a una lista

El historial del estado como una cadena

Supongamos que partimos con el valor de un estado en 1, luego después de un evento, como por ejemplo un clic cambia el valor a 2, y luego a 3, etc. Aunque el valor cambie, sabemos que en el estado anterior tenía otro valor

Estado 1



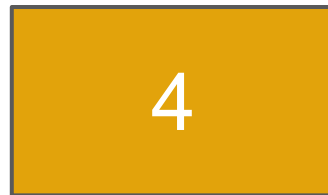
Estado 2



Estado 3



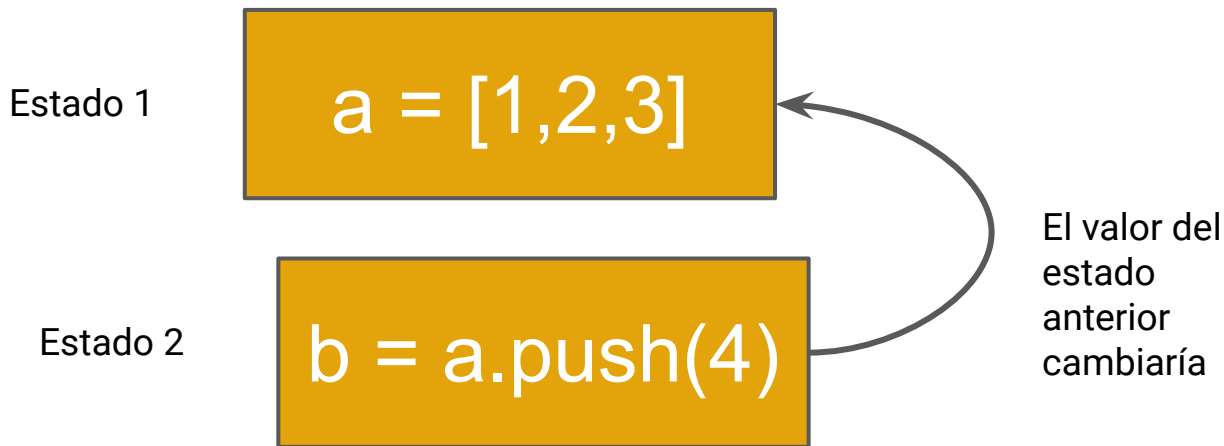
Estado 4



Agregar un elemento a una lista

El historial del estado como una cadena

Con los arreglos y objetos es un poco más complejo, si no tenemos cuidado al asignar un estado nuevo podríamos modificar el estado anterior.



Agregar un elemento a una lista

El historial del estado como una cadena



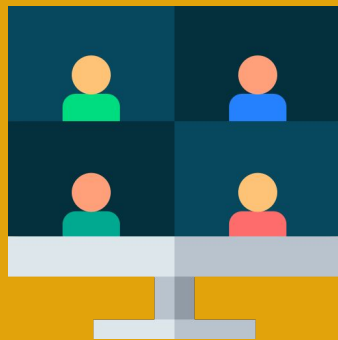
Evitemos mutar el estado directamente, o, en otras palabras debemos crear un objeto nuevo al momento de cambiar el estado

Ejercicio

Realicemos un ejercicio guiado. Crearemos un proyecto llamado react-toDo

1. Agregaremos el formulario un input y un submit
2. Agregaremos el estado tarea con useState y obtendremos la función setTarea, por defecto tarea estará en blanco.
3. Al cambiar el input se llamará a setTarea con el valor del input

Demostración



Ejercicio

Utilizando como base el proyecto anterior:

1. Agregaremos los listeners al formulario
2. Evitaremos que el formulario se envíe
3. Removemos el arreglo donde definimos las tareas, ya no lo necesitaremos
4. Usando useState agregaremos el estado tareas con un valor a inicial igual al arreglo de tareas definido previamente, la función setter se llamará setTarea
5. Al enviar el formulario agregaremos una tarea nueva utilizando
6. Comparte el código :)

Demostración



Ejercicio

Practiquemos lo aprendido. Repite los pasos previos en el proyecto pintar-colores.



Tip: La guía contiene el paso a paso

Ejercicio

¡Manos al teclado!





Cierre

{desafío}
latam_



¿Existe algún concepto que no
hayas comprendido?

Reflexionemos

- Revisemos el material asincrónico de la unidad.
 - En la guía encontraremos cómo editar elementos, por ejemplo marcar una tarea como lista.
 - En la guía encontraremos cómo borrar elementos.
- Revisemos el desafío de la unidad leyendo la descripción y requerimientos.

¿Qué sigue?

¿Tienen alguna duda respecto al Desafío?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam