7 - Componentes: propiedades

El elemento fundamental que tenemos para comunicarnos con una componente son las propiedades. Mediante las propiedades podemos enviar datos a la componente para que los muestre.

En el concepto anterior vimos como pasar un valor entero a la componente Dado mediante la sintaxis:

<Dado valor={this.state.valor1} />

Luego dentro de la componente 'Dado' accedemos al valor pasado mediante el atributo 'props':

<div className="Dado-recuadro">{this.props.valor}</div>

Podemos pasar cualquier tipo de datos a una componente, no solo tipos primitivos como enteros, reales, string.

Problema

Confeccionar una aplicación que permita ingresar por teclado dos enteros y nos muestre la suma. Agregar a una lista todas las operaciones ejecutadas hasta este momento.

Crear un nuevo proyecto con la herramienta create-react-app llamado **proyecto007**

Primero creamos el archivo 'ListadoResultados.js' en la carpeta 'src' donde definimos la componente con el mismo nombre:

import React, {Component} from 'react'

class ListadoResultados extends Component {

render() {

return (

<ul>

{this.props.resultados.map((elemento) => (

<li>La suma de {elemento.valor1} y {elemento.valor2} es {elemento.resultado}</li>

))}

</ul>

)

}

}

export default ListadoResultados;

La componente dispone de una propiedad llamada 'resultados' que llegará desde la componente 'App'. La misma es un vector con componentes de tipo objeto que almacenan los dos valores y el resultado de su suma.

Podemos imaginar que llega algo similar a esto:

[

{

valor1: 10,

valor2: 12,

resultado: 22

},

{

valor1: 5,

valor2: 20,

resultado: 25

},

{

valor1: 1,

valor2: 1,

resultado: 2

}

]

La componente App es:

import React, { Component } from 'react';

import './App.css';

import ListadoResultados from './ListadoResultados'

class App extends Component {

constructor(props) {

super(props);

this.sumar=this.sumar.bind(this);

this.state = {

resultados: []

}

}

render() {

return (

<div>

<form onSubmit={this.sumar}>

<p>Ingrese primer valor:<input type="text" name="valor1" /></p>

<p>Ingrese segundo valor:<input type="text" name="valor2" /></p>

<input type="submit" value="Sumar" />

</form>

<ListadoResultados resultados={this.state.resultados} />

</div>

);

}

sumar(event) {

event.preventDefault();

const v1=parseInt(event.target.valor1.value,10);

const v2=parseInt(event.target.valor2.value,10);

const suma=v1+v2;

const nuevo = {

resultado: suma,

valor1: v1,

valor2: v2

}

const vec = this.state.resultados;

vec.unshift(nuevo);

this.setState({

resultados: vec

});

}

}

export default App;

Importamos el archivo que contiene la clase ListadoResultados:

import ListadoResultados from './ListadoResultados'

En el constructor de la clase 'App' definimos el método que captura el clic de 'sumar' y en el 'estado' de la componente almacenamos un vector vacío:

constructor(props) {

super(props);

this.sumar=this.sumar.bind(this);

this.state = {

resultados: []

}

El método render() además del formulario de entrada de datos propiamente dicho definimos un elemento de tipo 'ListadoResultados' y le pasamos la propiedad 'resultado' con el valor almacenado en this.state.resultados:

render() {

return (

<div>

<form onSubmit={this.sumar}>

<p>Ingrese primer valor:<input type="text" name="valor1" /></p>

<p>Ingrese segundo valor:<input type="text" name="valor2" /></p>

<input type="submit" value="Sumar" />

</form>

<ListadoResultados resultados={this.state.resultados} />

</div>

);

}

Cada vez que se presiona el botón sumar se ejecuta el método donde agregamos al estado un nuevo elemento al principio del vector, esto hará que se actualice la lista de resultados en la página sin tener que recargarla:

sumar(event) {

event.preventDefault();

const v1=parseInt(event.target.valor1.value,10);

const v2=parseInt(event.target.valor2.value,10);

const suma=v1+v2;

const nuevo = {

resultado: suma,

valor1: v1,

valor2: v2

}

const vec = this.state.resultados;

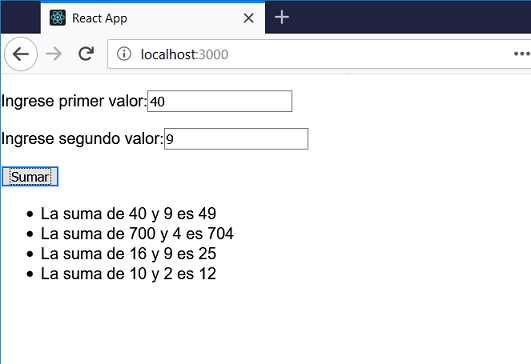
vec.unshift(nuevo);

this.setState({

resultados: vec

});

}



[Retornar](https://www.tutorialesprogramacionya.com/reactya/index.php?inicio=0)