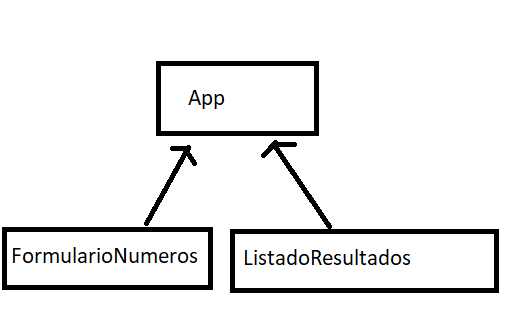
8 - Componentes: eventos generados por una componente

Una aplicación consta de una componente principal (utilizando la herramienta create-react-app se llama App), en esta definimos objetos de otras componentes y así sucesivamente cada componente puede estar construida en base a otras componentes.

Veremos ahora que una componente puede emitir un evento para informar a la componente padre un suceso.

Problema

Modificaremos el proyecto del concepto anterior para implementar el formulario de entrada de datos en otra componente llamada 'FormularioNumeros'. Las componentes que ahora tendrá la aplicación son:



Debemos crear el archivo 'FormularioNumeros.js' en la carpeta 'src' con el siguiente contenido:

import React, {Component} from 'react'

class FormularioNumeros extends Component {

render() {

return (

<form onSubmit={this.props.onSumar}>

<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>

);

}

}

export default FormularioNumeros;

En el evento onSubmit le pasaremos la referencia a un método que llega como una propiedad en el atributo 'props' y lo llama en la clase padre con el nombre 'onSumar:

<form onSubmit={this.props.onSumar}>

Ahora el código de la componente 'App' es:

import React, { Component } from 'react';

import './App.css';

import ListadoResultados from './ListadoResultados';

import FormularioNumeros from './FormularioNumeros';

class App extends Component {

constructor(props) {

super(props);

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

this.state = {

resultados: []

}

}

render() {

return (

<div>

<FormularioNumeros onSumar={this.sumar} />

<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;

Debemos importar primero los dos archivos que contienen las componentes:

import ListadoResultados from './ListadoResultados';

import FormularioNumeros from './FormularioNumeros';

El método render() ahora ha quedado muy simple, hemos inicializado la propiedad 'onSumar' con la referencia del método 'sumar':

render() {

return (

<div>

<FormularioNumeros onSumar={this.sumar} />

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

</div>

);

}

Como vemos el método sumar se ejecuta cuando la componente FormularioNumeros lo requiera.

Finalmente el archivo 'ListadoResultados' no varía con respecto al problema anterior:

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;

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