

Ejercicios del Tema 4

Crea un nuevo proyecto de React. En esta ocasión, vamos a crear un formulario más completo que el que hemos visto en este tema. Incluiremos los siguientes campos:

- Nombre
- Apellidos
- Edad
- Profesión

Parte del código visto en el tema para la creación del formulario:

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import './index.css';

class Formulario extends React.Component{

  constructor(props){
    super(props);
    this.state = {value: ''}
  }

  manejarCambios = (event) => {
    this.setState({value: event.target.value});
  }

  manejarEnvio = (event) => {
    console.log("Se ha enviado el valor " + this.state.value);
    event.preventDefault();
  }

  render(){
    return (
      <form onSubmit={this.manejarEnvio}>
        <label>Nombre:</label>
        <input type="text" value={this.state.value}
onChange={this.manejarCambios} />
        <input type="submit" value="Enviar" />
      </form>
    );
  }
}

ReactDOM.render(
  <Formulario/>,
  document.getElementById('root')
);
```

Ejercicio 1 - Crea un formulario en el que se pidan los datos anteriores. Cada elemento debe tener una casilla asociada para que el usuario pueda introducir sus datos en el formulario. Por último, crea un botón en la parte inferior que sirva para enviar los datos anteriores. Por consola deberá mostrarse una cadena de texto que muestre los valores enviados para verificar que son correctos.

Ampliación - Añade una cadena de texto vacía en la parte inferior. Cuando se pulse el botón de enviar, en la cadena de texto aparecerán los nombre de los atributos y los valores que el usuario le ha dado. Por ejemplo: "Nombre: Juan, Apellidos: García Martínez, Edad: 26, Profesión: Carpintero". Esto servirá como comprobación de que el usuario ha enviado sus datos.

Ejercicio 2 - Ahora pasaremos a trabajar con la aplicación para la listar tareas. Ya hemos visto cómo implementar las acciones básicas para crear una tarea, marcarla como completada o no y eliminarla. También hemos incluido la opción de editar una tarea ya creada.

Las aplicaciones de este estilo suelen incluir muchas más opciones, como poner las tareas de distinto color según quiera el usuario, marcar las tareas más importantes, añadir fechas, etc.

Crea un nuevo proyecto React y, a continuación, modifica el archivo `index.html`:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>JS Bin</title>
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
rel="stylesheet" type="text/css" />
</head>
<body>
<script src="https://fb.me/react-0.14.7.js"></script>
<script src="https://fb.me/react-dom-0.14.7.js"></script>
  <div id="container"></div>
</body>
</html>
```

Partiremos de nuestro código base, visto en las transparencias. El componente **App**:

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import TodoForm from './ToDoForm';
import ToDoList from './ToDoList';
class App extends Component{
  constructor() {
    super();
    this.state={tasks: [
      { text: "Tarea 1", completed: false },
      { text: "Tarea 2", completed: false },
      { text: "Tarea 3", completed: false },
      { text: "Tarea 4", completed: true }
    ]
  }
  this.addTask = this.addTask.bind(this);
  this.saveTask = this.saveTask.bind(this);
  this.removeTask = this.removeTask.bind(this);
  this.editTask = this.editTask.bind(this);
  this.taskAction = this.taskAction.bind(this);
}
addTask(task) {
  if( task.text ){
    this.state.tasks.push(task);
    this.setState( this.state );
  }
}
saveTask(index, task) {
  if( task.text ){
    this.state.tasks[index] = task;
    this.state.edit = null;
    this.setState( this.state );
  }
}
removeTask(index) {
  if( index >= 0 && this.state.tasks.length > 0 ){
    this.state.tasks.splice(index, 1);
    this.setState( this.state );
  }
}
editTask(index) {
  if( index >= 0 && this.state.tasks.length > 0 ){
    var task = this.state.tasks[index];
```

El componente **ToDoAction**, que asignará los elementos a realizar para una tarea:

```
import React, { Component } from 'react';
class TaskAction extends Component{
  constructor(props){
    super(props)
    this.handleStatus = this.handleStatus.bind(this);
    this.handleEdit = this.handleEdit.bind(this);
    this.handleRemove = this.handleRemove.bind(this);
    this.getButton = this.getButton.bind(this);
  }
  handleStatus(){
    this.props.setStatus(this.props.index);
  }
  handleEdit(){
    this.props.edit(this.props.index);
  }
  handleRemove(){
    this.props.remove(this.props.index);
  }
  getButton(){
    if( !this.props.completed ){
      return (
        <button type="button"
          onClick={this.handleStatus}
          className="btn btn-xs btn-success">
            <i className="glyphicon glyphicon-ok"></i>
          </button>
        )
      } else {
        return (
          <button type="button"
            onClick={this.handleStatus}
            className="btn btn-xs btn-info">
              <i className="glyphicon glyphicon-repeat"></i>
            </button>
          )
        }
      }
    render() {
      return (
        <div className="btn-group btn-group-xs pull-right" role="group">
```

El componente **ToDoForm**, que dará forma a la aplicación:

```
import React, { Component } from 'react';
class ToDoForm extends Component{
  constructor(props){
    super(props);
    this.handleAddSave = this.handleAddSave.bind(this);
  }

  componentWillReceiveProps(newProps){
    if(newProps.edit){
      this.refs.task.value = newProps.edit.text;
    }
  }
  handleAddSave(){
    if(this.props.edit){
      var task = {
        text: this.refs.task.value,
        completed: this.props.edit.completed
      };
      this.props.save( this.props.edit.index, task );
    } else {
      var task = {text: this.refs.task.value, completed: false};
      this.props.add( task );
    }
    this.refs.task.value = '';
  }
  render() {
    return (
      <div className="jumbotron padded">
        <form>
          <div className="row">
            <div className="col-md-9">
              <div className="form-group">
                <input type="text" className="form-control"
                  placeholder="Task" ref="task" />
              </div>
            </div>
            <div className="col-md-3">
              <button type="button" onClick={this.handleAddSave}
                className="btn btn-primary btn-block">{this.props.edit ? 'Save' :
                'Add'}</div>
            </div>
          </div>
        </form>
      </div>
    );
  }
}
```

Y finalmente el componente **ToDoList**, que creará la lista de tareas y añadirá los accionables correspondientes:

```
import React, { Component } from 'react';
import TaskAction from './ToDoAction'
class ToDoList extends Component{
  render() {
    return (
      <div className="row padded">
        <div className="col-lg-12">
          <h4>Tasks</h4>
          <hr/>
          <ul className="list-group">
            {
              this.props.tasks.map(function(task, index){
                var cssClass = 'list-group-item list-group-item-';
                if( task.completed ){
                  cssClass += 'info';
                } else {
                  cssClass += 'success';
                }
                return (
                  <li key={index} className = {cssClass}>
                    <TaskAction
                      index={index}
                      completed={task.completed}
                      setStatus={this.props.taskAction}
                      edit={this.props.edit}
                      remove={this.props.remove}/>
                    {task.text}
                  </li>
                )
              }, this)
            }
          </ul>
        </div>
      </div>
    )
  }
}
```

Implementa un modo de filtrar las tareas según su estado: completado, activas o ambos. Como ejemplo, puedes crear tres botones que cuando se pulsen hagan que el listado de tareas pase a mostrar únicamente las del tipo correspondiente.

Ampliación - Añade un nuevo estado “urgente” para las tareas. Al pulsar un nuevo botón se marcará la tarea como urgente, y podrá ser completada igual que las otras.