

10

Single Page Application (SPA) con React

Visual Studio 2017 Web Developer

Objetivos

Al finalizar el capítulo, el alumno logrará:

- Implementar Frontend con ReactJs.
- Utilizar los componentes ReactJS.
- Administrar estados con Redux



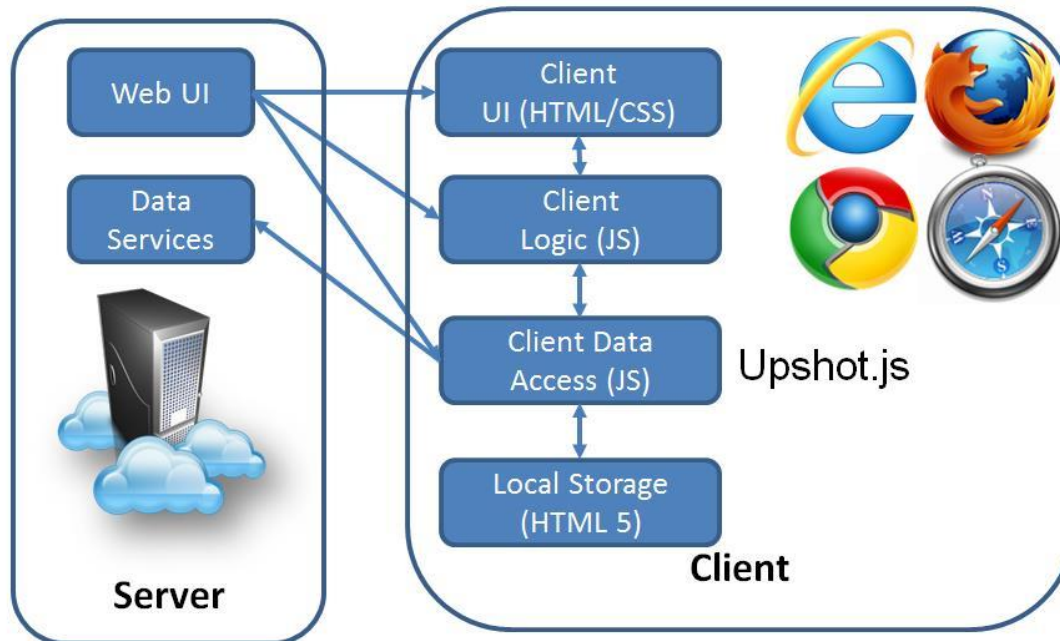
Agenda

- Definición de SPA
- Estructura de una SPA con React
- Fundamentos React.
- Componentes React.
- Ruteo con React.
- Manejo de estados con Redux.



Definición de SPA

Las SPA (Single Page Applications) son páginas web que en los últimos años se han puesto de moda para la realización de aplicaciones full responsive para los dispositivos móviles y las tabletas.



Las principales ventajas de las SPA

- Se reducen considerablemente las fases de trabajo en arquitectura de información por utilizar una única plantilla.
- La plantilla única hace que también se reduzca la cantidad de código utilizado.
- Se simplifica la gestión de análisis de la web ya que solamente se realiza de una página.
- Las SPA funcionan como una landing única, concentrando mejor las conversiones y con un punto de conversión y optimización muy centrado y definido.
- Facilita el SEM ya que concentramos todas las llamadas y posibilidades de interacción en una única página.
- Al contar con un único diseño general se le puede dedicar todo el tiempo a optimizar los pequeños detalles y a probar todo tipo de diseños.



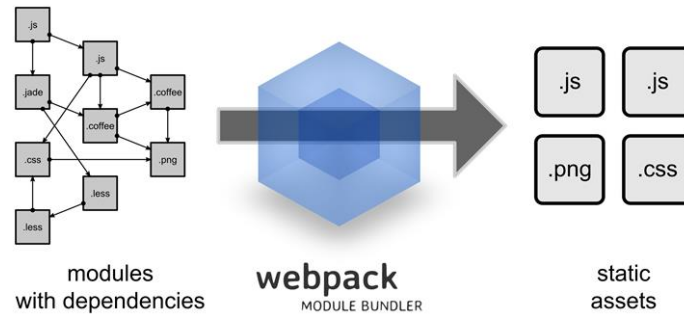
Fundamentos React

- React (React / React.js) – <https://facebook.github.io/react/>
 - Es una librería JavaScript para construir interfaces de usuario(UI)
 - Open-source, declarativo, basado en components y con el soporte de Facebook

```
class HelloMessage extends React.Component {  
  render() {  
    return <div>Hello {this.props.name}</div>;  
  }  
}  
ReactDOM.render(<HelloMessage name="Maria" />,  
  document.getElementById('container'));
```



Fundamentos React



Componentes React

```
let element =  
  <h1>Hello, <b>React</b>!</h1>;  
console.dir(element);
```

```
let age = 18;  
let ageElement =  
  <p>Age: {age}</p>;  
ReactDOM.render(ageElement,  
  document.getElementById('app'));
```

▼ Object

```
  $$typeof: Symbol(react.element)  
  _owner: null  
  _self: null  
  _source: null  
  ► _store: Object  
    key: null  
  ▼ props: Object  
    ▼ children: Array[3]  
      0: "Hello, "  
      ► 1: Object  
        2: "!"  
        length: 3  
      ► __proto__: Array[0]  
    ► __proto__: Object  
  ref: null  
  type: "h1"  
  ► __proto__: Object
```



Componentes React

Es posible crear componentes del tipo clase.

```
class Person extends React.Component {  
  render() {  
    return (  
      <div>  
        <h1>{this.props.person}</h1>  
        <p>Tel. {this.props.phone}</p>  
      </div>  
    )  
  }  
}
```



Componentes React

Es posible crear componentes del tipo objeto.

```
let Person = React.createClass({  
  render: function() {  
    return (  
      <div>  
        <h1>{this.props.person}</h1>  
        <p>Tel. {this.props.phone}</p>  
      </div>  
    )  
  }  
});
```

```
ReactDOM.render(  
  <Person person="Nakov"  
    phone="+359 2 981 981" />,  
  document.getElementById('app')  
);
```



Ruteo con React

Parte muy importante de los SPA es la administración del ruteo, para ese efecto en React tenemos react-router, la cual se configura como la imagen siguiente:

```
<Provider store={store}>
  <Router history={hashHistory}>
    <Route path="/" component={Login}/>
    <Route path="/customers" component={CustomerList}/>
    <Route path="/customerview/:id" component={CustomerDisplay}/>
    <Route path="/customercreate" component={CustomerCreate}/>
    <Route path="/customeredit/:id" component={CustomerUpdate}/>
    <Route path="*" component={NotFound} />
  </Router>
</Provider>
```



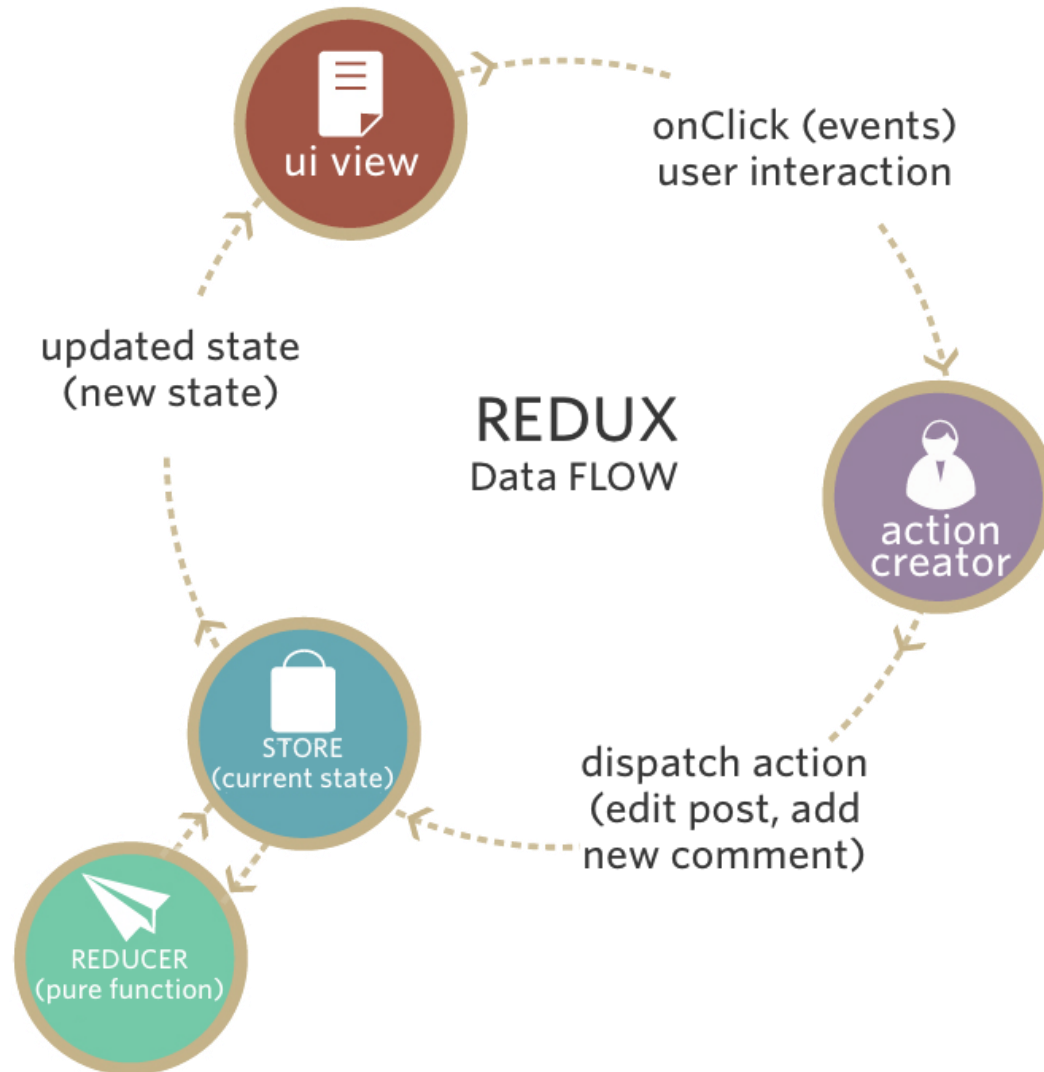
Manejo de estados con Redux

Redux es una librería para controlar el estado de nuestras aplicaciones web fácilmente, de una forma consistente entre cliente y servidor, testeable y con una gran experiencia de desarrollo.

Redux está en gran parte influenciado por la arquitectura **Flux** propuesta por Facebook para las aplicaciones de **React.js** y por el lenguaje **Elm**, esta muy pensado para **React.js**, pero también se puede usar con **Angular.js**, **Backbone.js** o simplemente con **Vanilla JS**.



Manejo de estados con Redux



Lecturas adicionales

Para obtener información adicional, puede consultar los siguientes enlaces:

- Tutorial de React:
 - <https://miguelgomez.io/react/tutorial-react-js-introduccion/>
- Introduccion a React
 - <http://vensign.com/introduccion-a-React/>
- Redux:
 - <https://medium.com/react-redux/introducci%C3%B3n-a-redux-js-8bdf4fe0751e>



Ejercicio N° 10.1: Instalación y configuración de Nodejs y Visual Studio Code

Configurar el entorno básico para el desarrollo de un SPA con React y Visual Studio Code.

Al finalizar el laboratorio, el alumno logrará:

- Configurar su entorno de desarrollo para trabajar con React.



Ejercicio N° 10.2: Configuración de proyecto web con React

Implementar un proyecto básico con React y Visual Studio Code.

Al finalizar el laboratorio, el alumno logrará:

- Instalar paquetes necesarios para el desarrollo con React y crear su primer componente React.



Ejercicio N° 10.3: Implementación de Redux para el módulo de Login

Implementar el manejo de estados con Redux en un proyecto SPA de React y Visual Studio Code.

Al finalizar el laboratorio, el alumno logrará:

- Implementar el manejo de estados con Redux en un proyecto SPA de React.



Ejercicio N° 10.4: Listado y paginación con React y Bootstrap

Implementar paginación en un proyecto SPA de React y Visual Studio Code.

Al finalizar el laboratorio, el alumno logrará:

- Crear componentes que administren paginación en aplicaciones SPA con React.



Ejercicio N° 10.5: Mantenimiento de Customer con React

Implementar CRUD en un proyecto SPA de React y Visual Studio Code.

Al finalizar el laboratorio, el alumno logrará:

- Implementar un mantenimiento (CRUD) para la tabla Customer con React.



Resumen

En este capítulo, usted aprendió a:

- Implementar aplicaciones web haciendo de uso de React.
- Consumir una Web API desde un SPA.



Tarea N° 10: Implementa Redux y componentes para la navegación el módulo Producto

Implementar lo aprendido para el módulo productos

Al finalizar el laboratorio, el alumno logrará:

- Aplicar los conceptos sobre React.

