

Introducción a React

Herramientas a utilizar

React

☐ Herramientas que voy a utilizar

- ☐ Google Chrome
- ☐ React Developer Tools
- ☐ Visual Studio Code
- ☐ Node. Runtime de JavaScript basado en chrome V8 JavaScript Engine

Introducción a REACT

React

☐ **Que es React?**

- ☐ Una biblioteca JavaScript para crear interfaces de usuario
- ☐ Declarativo
- ☐ Basado en componentes
- ☐ Puede integrarse fácilmente con otros productos

React

❑ Historia de React.

- ❑ Diseñado por Jordan Walke
- ❑ Implementado por primera vez en las noticias de Facebook en 2011
- ❑ Código abierto en JSConf US en mayo de 2013
- ❑ Diseñado para la velocidad, la simplicidad y la escalabilidad

React

☐ **Vocabulario de React.**

☐ Válido para aplicaciones SPA

☐ JSX

☐ Componentes

☐ Hooks

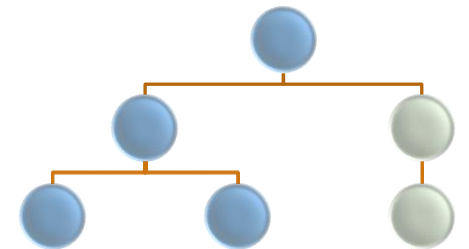
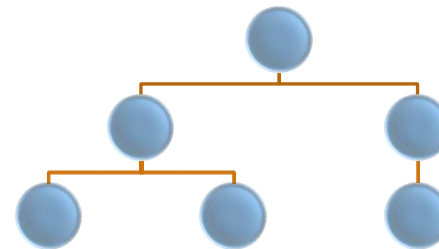
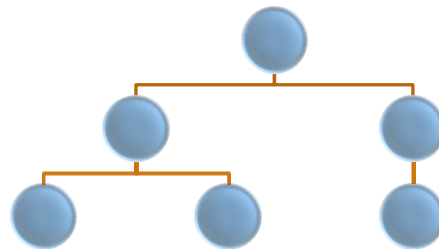
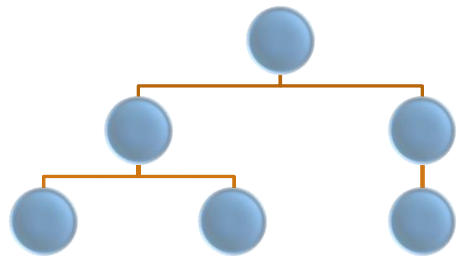
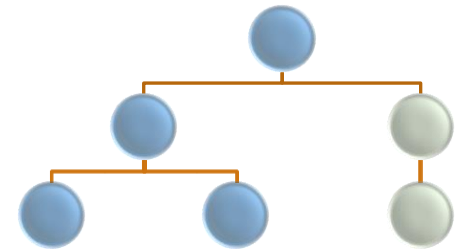
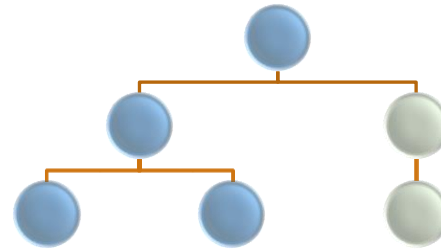
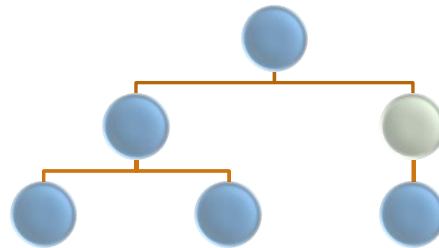
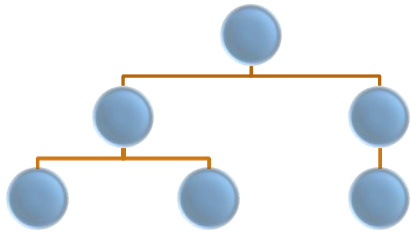
☐ Props

☐ Virtual DOM

React

Virtual DOM

VIRTUAL DOM



DOM DEL NAVEGADOR

JSX

React

□ JSX

- Extensión sintáctica a JavaScript
- Notación abreviada para representar llamadas a funciones JavaScript que evalúan objetos JavaScript
- Evita la separación artificial de la lógica de representación de otra lógica de interfaz de usuario

```
const element = <h1>Mi primer JSX</h1>;
```

React

□ Embebiendo expresiones en JSX

```
const alumno = { id: 0, nombre: 'Pedro', ... };  
return (  
  <div key={alumno.id} className="col-12 mt-5">  
    <Media tag="li">  
      <Media body className="ml-5">  
        <Media heading>{alumno.nombre}</Media>  
      </Media>  
    </Media>  
  </div>  
>);
```

Componentes

React

☐ Componentes

- ☐ Elemento básico de REACT
- ☐ Un componente devuelve un conjunto de elementos React que deberían aparecer en la pantalla
- ☐ Los componentes le permiten dividir su interfaz de usuario en piezas independientes y reutilizables
- ☐ Los componentes también aceptan entradas
- ☐ Se pueden definir diferentes tipos de componentes en React

React

❑ Componentes

- ❑ Los componentes permiten separar la interfaz de usuario en piezas independientes, reutilizables y pensar en cada pieza de forma aislada.
- ❑ Conceptualmente, los componentes son como las funciones de JavaScript.
- ❑ Aceptan entradas arbitrarias (llamadas “props”) y retornan elementos de React que describen lo que debe aparecer en la pantalla.

React

❑ Componentes

❑ Componentes funcionales y de clase

- ❑ Funcionales. La forma más sencilla de definir un componente es escribir una función de JavaScript:

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

- ❑ Clase: se usa una clase de ES6 para definir un componente:

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

React

❑ Componentes

- ❑ Los nombres de componentes definidos por el usuario siempre deben comenzar con una letra mayúscula.
- ❑ Estos compilan con `React.createElement(...)`
- ❑ Las etiquetas que comienzan con letras minúsculas se tratan como etiquetas DOM.
- ❑ Componentes integrados
 - ❑