

Tema 3 - React a fondo: composición vs herencia

Comunicación hijo -> padre

- ¿Y si queremos comunicarnos de hijo de padre?
- Hacemos que el componente tenga una API definida, usando funciones
- El padre pasará al hijo funciones como props
- El componente hijo llamará a esos “callbacks” con datos específicos de su dominio

Comunicación hijo -> padre

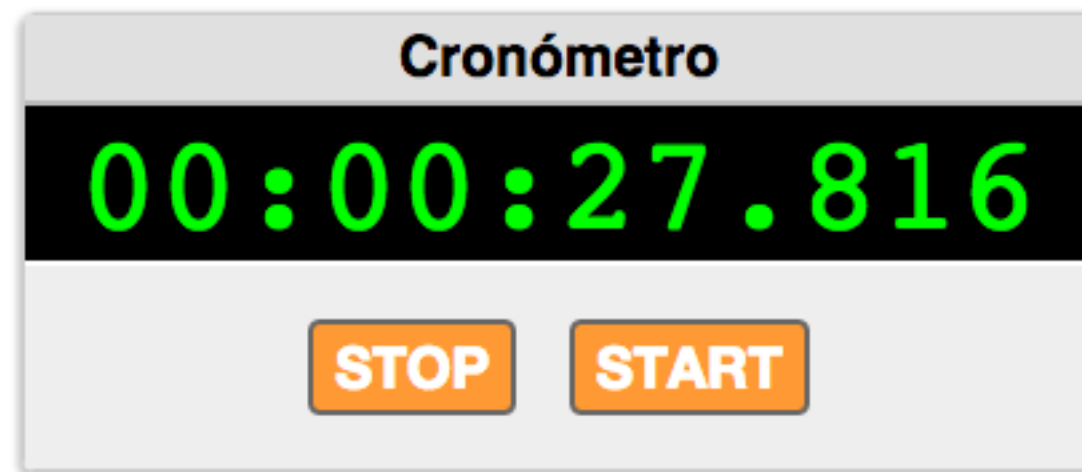
```
var Buttons = React.createClass({
  propTypes: {
    onStart: React.PropTypes.func.isRequired,
    onStop: React.PropTypes.func.isRequired
  },
  render: function() {
    return (
      <div className="actions">
        <button onClick={this.props.onStop}>STOP</button>
        <button onClick={this.props.onStart}>START</button>
      </div>
    );
  }
});
```

Comunicación hijo -> padre

```
var Parent = React.createClass({
  handleStart: function(e) {
    console.log("Start en Buttons!");
  },
  handleStop: function(e) {
    console.log("Stop en Buttons!");
  },
  render: function() {
    return (
      <div className="parent">
        <Buttons onStart={this.handleStart} onStop={this.handleStop} />
      </div>
    );
  }
});
```

Ejercicio: cronómetro

- Ya podemos crear por fin nuestro primer componente interactivo usando props, estado interno y eventos
- Vamos a implementar un cronómetro como éste: el botón START inicia el temporizador, y el botón STOP lo detiene en el primer clic y lo reinicia a 0 en el segundo.



Ejercicio: cronómetro

- Disponéis de la plantilla para este ejercicio en `/ejercicios/tema3/src/templates/cronometro.html`
- Tenéis funciones auxiliares para manipular el tiempo con fechas en `/ejercicios/tema3/src/lib/utils.js`



Cronómetro: pistas

- Utilizar **composición**: el cronómetro completo debe contener un componente Header, un componente Screen y un componente Buttons.
- Se pueden pasar funciones como **props** de modo que un evento sea “atendido” por el componente padre de quien lo registra y recibe.
- Intentar basar el paso de datos padre-hijo en **props**
- No almacenar información **derivada** en el estado (que pueda ser calculada a partir de props o estado)

Acceso al DOM

- React gestiona el DOM por nosotros, nosotros generamos VirtualDOM y la librería hace el diff automáticamente
- Si necesitamos acceder a un nodo montado en el DOM, tenemos que marcarlo en JSX con una referencia:

```
<button ref="miboton">Click me</button>
```

- Después podemos obtener la referencia en código con **this.refs.miboton**. Si React elimina o sustituye ese nodo, actualizará la referencia para nosotros (o será *undefined*)
- Para acceder al DOM nativo y sus propiedades, podemos llamar a **ReactDOMNode()** sobre la referencia obtenida con **this.refs.X**

Formularios

- Los controles de formulario HTML son problemáticos para React
- Son inherentemente mutables mediante interacciones de usuario (comportamiento definido por el navegador)

Formularios

```
var TextInput = React.createClass({  
  getInitialState: function() {  
    return {  
      value: ""  
    };  
  },  
  render: function() {  
    return (  
      <input type="text" value="Introduce tu nombre">  
    );  
  }  
});
```

Si intentamos escribir en esa caja de texto, no pasará nada
¿Por qué?

Formularios

```
var TextInput = React.createClass({
  getInitialState: function() {
    return {
      value: ""
    };
  },
  render: function() {
    return (
      <input type="text" value="Introduce tu nombre">
    );
  }
});
```

Porque **render** dice que, invariablemente, el valor de ese INPUT es "Introduce tu nombre"

Formularios

```
var TextInput = React.createClass({  
  getInitialState: function() {  
    return {  
      value: ""  
    };  
  },  
  render: function() {  
    return (  
      <input type="text" value="Introduce tu nombre">  
    );  
  }  
});
```

Si fuera HTML y no React, podríamos borrar ese texto y escribir otro...

Formularios

- Es un “choque” conceptual con el Virtual DOM de React, que gestiona por nosotros todo el HTML producido
- Tenemos props **específicas** para controles de formularios
- Y un evento muy útil: **onChange**

Formularios

- **value** - establece el valor en:
 - `<input type="text" .../>`
 - `<input type="password" .. />`
 - `<textarea .. />`
 - `<select />` (valor del elemento seleccionado)

Formularios

- **checked** - (Boolean) recupera/establece si están activos:
 - `<input type="checkbox" .../>`
 - `<input type="radio" .. />`

Formularios

- **selected** - (Boolean) recupera/establece si están seleccionados los elementos **option** de un desplegable:
- `<select>`
 - `<option value="1">Uno</option>`
 - `<option value="2">Dos</option>`
 - `</select>`

Formularios: componentes controlados

- La salida del método **render** define el estado de la UI en cualquier momento determinado

- Si escribimos

```
<textarea value="Introduce tu comentario"></  
textarea>
```

- El usuario **no puede modificar** el contenido. Está “hard-coded” en el código Javascript generado a partir de JSX

Formularios: componentes controlados

- La solución es utilizar el estado interno del componente como fuente para el control del formulario
- Implica que tenemos que modificar “manualmente” el estado cada vez que el usuario modifique el control
- **onChange** funciona en todos los controles

Formularios: componentes controlados

```
var UserLogin = React.createClass({
  getInitialState: function() {
    return { email: "" };
  },
  onEmailChange: function(e) {
    this.setState( { email: e.target.value } );
  },
  render: function() {
    return (
      <div>
        Email:
        <input type="text" value={ this.state.email }
          onChange={this.onEmailChange} />
      </div>
    );
  }
});
```

Formularios: componentes controlados

```
var UserLogin = React.createClass({
  getInitialState: function() {
    return { email: "" };
  },
  onEmailChange: function(e) {
    this.setState( { email: e.target.value } );
  },
  render: function() {
    return (
      <div>
        Email:
        <input type="text" value={ this.state.email }
          onChange={this.onEmailChange} />
      </div>
    );
  }
});
```

En **cada cambio** notificado por **onChange**, actualizamos el estado interno...

...y esta actualización ejecutará de nuevo **render** y mostrará el valor correcto

Formularios: componentes no controlados

- ¿Y si no queremos el control total del formulario?
- Entonces usamos componentes no controlados
- En lugar de **value** definimos **defaultValue** que es sólo el valor por defecto
- Tendríamos que usar **referencias** y acceso al DOM con **getNode** para extraer el valor del campo más adelante.

Ejercicio: formularios

- Buscador de personajes de Juego de Tronos (spoiler alert)

Buscador Juego de Tronos

Actor / personaje

Familia

Todas

Sólo personajes vivos☐

Aparece en temporada

1☐ 2☐ 3☐ 4☐ 5☐

Personaje	Actor	Nº Ep	Vivo
Eddard Stark	Calvin Hobbs	45	Sí
Eddard Stark	Calvin Hobbs	45	Sí
Eddard Stark	Calvin Hobbs	45	Sí

Encontrados 25 personajes

Ejercicio: formularios

- Queremos un buscador que actualice los resultados en vivo, según se modifican los parámetros de búsqueda (al estilo **onChange**)
- Los datos en JSON están en `/ejercicios/tema3/src/data/got.js`
- La plantilla en `/ejercicios/tema3/plantillas/buscador.html`

Composición

- Una aplicación entera de React se pinta **a partir de un componente raíz**, que a su vez incluye componentes hijos y así sucesivamente
- El componente que incluye otro en su método render es el **dueño** de ese nodo hijo
- El padre puede pasarle props al hijo, configurándolo, y será el responsable del ciclo de vida del hijo
- Cuando no aparezca en su **render**, React desmontará el componente por nosotros

Composición

- La “manera React” es intentar hacer componentes específicos con el mínimo estado posible
- Recuerda: $UI = f(datos)$
- Es decir: **render** depende sólo de los **props** y **state** actuales del componente
- Separación de Responsabilidades a nivel de UI
- Cada componente hace una cosa

Composición

- De esta forma los componentes son cajas negras que “cableamos” mediante sus props.
- Le damos datos via props
- Atendemos sus *notificaciones* pasando una función vía props (ej: onChange en el buscador)

Composición

centralLog - Events

Anonymous (Guest)

Go to

MODE
LIVE

Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
<div></div>	<div></div>	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
<div></div>	<div></div>	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
<div></div>	<div></div>	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
<div></div>	<div></div>	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
<div></div>	<div></div>	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
<div></div>	<div></div>	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
<div></div>	<div></div>	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

Composición

<body>

Root

EventsLayout

FilterBox

Toolbar

Menu

EventList

EventListHeader

Event

Event

Event

</body>

Composición

centralLog - Events							
<div> <div></div> <div></div> </div>						<div> <div>Go to</div> <div> <div></div> <div>MODE LIVE</div> </div> </div>	
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
✖	🔊	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
ⓘ	🔊	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
⚠	🔊	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
⚠	🔊	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
ⓘ	🔊	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
✖	🔊	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
⚠	🔊	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
✖	🔊	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
✖	🔊	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
✖	🔊	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
ⓘ	🔊	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
ⓘ	🔊	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
✖	🔊	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
⚠	🔊	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
ⓘ	🔊	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
⚠	🔊	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
⚠	🔊	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<EventsLayout />

Composición

centralLog - Events

Anonymous (Guest)

Go to

MODE
LIVE

Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
<div></div>	<div></div>	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
<div></div>	<div></div>	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
<div></div>	<div></div>	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
<div></div>	<div></div>	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
<div></div>	<div></div>	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
<div></div>	<div></div>	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
<div></div>	<div></div>	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<FilterBox />

Composición

centralLog - Events							
<div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div></div> <div></div> </div> </div>							
<div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div>							
<div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div>							
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
✖	🔊	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
ⓘ	🔊	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
⚠	🔊	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
⚠	🔊	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
ⓘ	🔊	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
✖	🔊	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
⚠	🔊	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
✖	🔊	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
✖	🔊	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
✖	🔊	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
ⓘ	🔊	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
ⓘ	🔊	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
✖	🔊	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
⚠	🔊	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
ⓘ	🔊	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
⚠	🔊	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
⚠	🔊	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<Toolbar />

Composición

centralLog - Events							
<div> <div></div> <div></div> </div>						<div> <div>Anonymous (Guest)</div> <div></div> </div>	
<div> <div></div> <div></div> </div>						<div> <div>Go to</div> <div></div> </div>	
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
✖	🔊	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
ⓘ	🔊	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
⚠	🔊	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
⚠	🔊	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
ⓘ	🔊	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
✖	🔊	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
⚠	🔊	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
✖	🔊	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
✖	🔊	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
✖	🔊	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
ⓘ	🔊	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
ⓘ	🔊	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
✖	🔊	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
⚠	🔊	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
ⓘ	🔊	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
⚠	🔊	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
⚠	🔊	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<Menu visible={false} />

Composición

centralLog - Events

Anonymous (Guest)

Go to

MODE
LIVE

Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
<div></div>	<div></div>	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
<div></div>	<div></div>	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
<div></div>	<div></div>	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
<div></div>	<div></div>	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
<div></div>	<div></div>	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
<div></div>	<div></div>	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
<div></div>	<div></div>	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
<div></div>	<div></div>	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
<div></div>	<div></div>	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
<div></div>	<div></div>	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
<div></div>	<div></div>	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
<div></div>	<div></div>	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
<div></div>	<div></div>	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

Composición

centralLog - Events

Anonymous (Guest)

Go to

MODE LIVE

Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
		04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
		04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
		04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
		04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
		04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
		04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
		04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
		04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
		04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
		04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
		04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
		04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
		04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
		04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
		04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
		04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
		04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
		04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
		04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
		04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
		04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
		04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<EventListHeader />

Composición

centralLog - Events							
						Go to	MODE LIVE
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
✖	🔊	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
ⓘ	🔊	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
⚠	🔊	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
⚠	🔊	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
ⓘ	🔊	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
✖	🔊	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
⚠	🔊	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
✖	🔊	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
✖	🔊	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
✖	🔊	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
ⓘ	🔊	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
ⓘ	🔊	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
✖	🔊	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
⚠	🔊	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
ⓘ	🔊	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
⚠	🔊	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
⚠	🔊	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<Event severity="error" state="acked" />

Composición

centralLog - Events							
						Go to	Anonymous (Guest)
						MODE LIVE	
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
		04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
		04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
		04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
		04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
		04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
		04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
		04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
		04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
		04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
		04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
		04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
		04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
		04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
		04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
		04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
		04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
		04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
		04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
		04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
		04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
		04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
		04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<Event severity="info" state="acked" />

Composición

centralLog - Events							
Anonymous (Guest)							
Go to		MODE LIVE					
Sev	Ack	Time	Site	Module	Domain	Type	Message
04/08/2015							
✖	🔊	04/08/2015 09:53:05.000	Ryadh	FDS	BBDB5	R	test1_1
🔊	🔊	04/08/2015 09:53:05.000	Madrid	Fleet	BBDB5	N	test2_2
⚠	🔊	04/08/2015 09:53:04.000	Ryadh	Fleet	AB4FB	N	test1_1
⚠	🔊	04/08/2015 09:53:04.000	Madrid	Fleet	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	test1_1
✖	🔊	04/08/2015 09:53:03.000	Ryadh	Fleet	BBDB5	N	error enormous
🔊	🔊	04/08/2015 09:53:03.000	Ryadh	FDS	AM01M	N	test1_1
✖	🔊	04/08/2015 09:53:01.000	Ryadh	FDS	AM01M	R	test1_1
⚠	🔊	04/08/2015 09:53:01.000	Madrid	RTS2	AB5C	A	error enormous
✖	🔊	04/08/2015 09:53:00.000	Ryadh	Fleet	AM01M	R	test1_1
✖	🔊	04/08/2015 09:53:00.000	Madrid	Fleet	AM01M	A	test2_2
✖	🔊	04/08/2015 09:52:59.000	Madrid	FDS	AB4FB	A	error enormous
🔊	🔊	04/08/2015 09:52:59.000	Ryadh	Fleet	AM1AC	N	test1_1
🔊	🔊	04/08/2015 09:52:58.000	Ryadh	FDS	AB5C	A	test1_1
✖	🔊	04/08/2015 09:52:58.000	Madrid	FDS	AB4FB	M	error enormous
⚠	🔊	04/08/2015 09:52:58.000	Ryadh	Fleet	AB4FB	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	FDS	AM01M	N	error enormous
⚠	🔊	04/08/2015 09:52:56.000	Madrid	RTS2	AM1AC	N	test1_1
✖	🔊	04/08/2015 09:52:55.000	Madrid	Fleet	AB5C	N	test1_1
🔊	🔊	04/08/2015 09:52:54.000	Ryadh	RTS1	AB4FB	M	test1_1
⚠	🔊	04/08/2015 09:52:52.000	Ryadh	RTS2	AM1AC	R	test2_2
⚠	🔊	04/08/2015 07:53:03.000	Ryadh	RTS1	AB5C	N	test2_2

<Event severity="warning" state="active" />

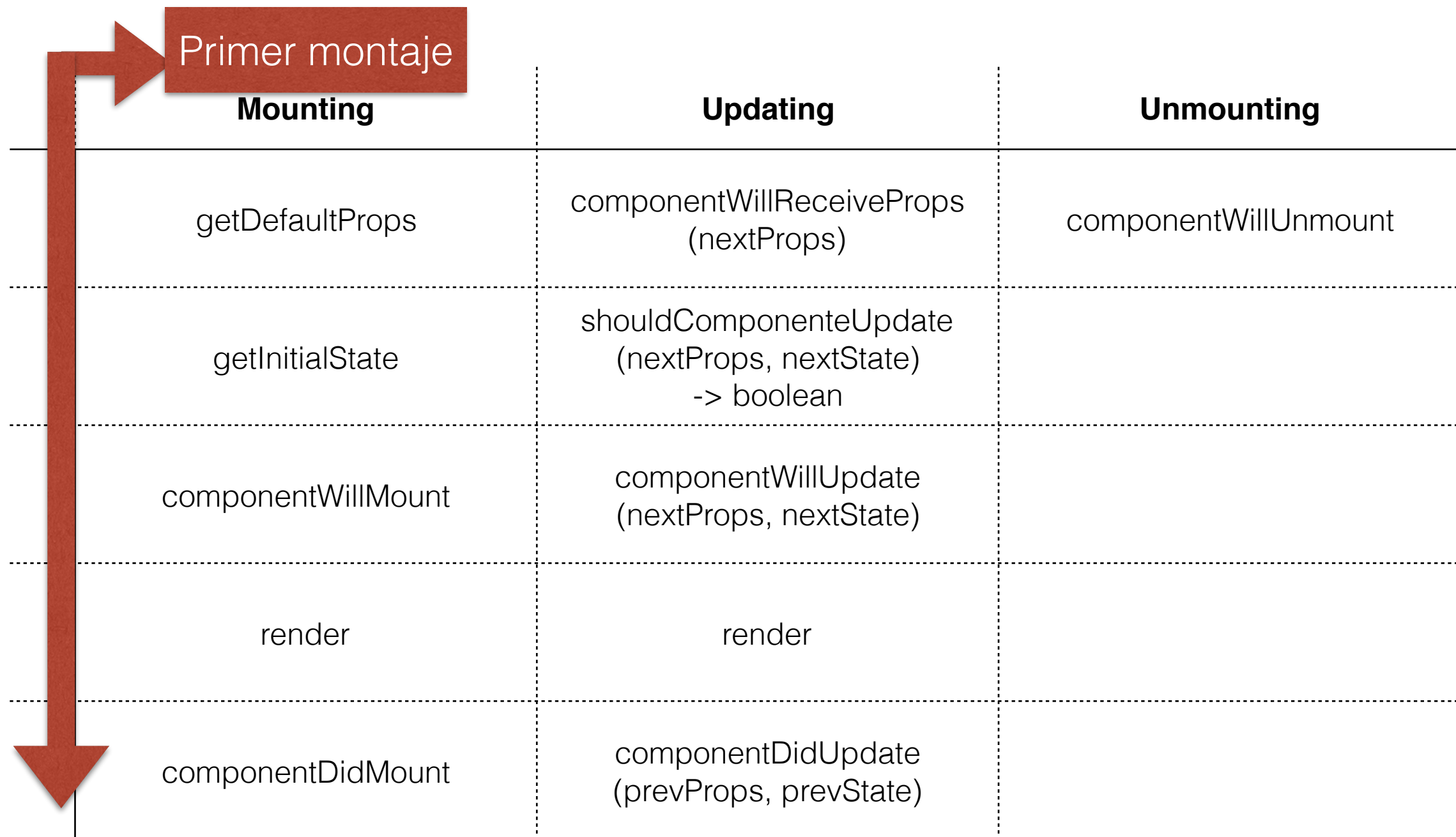
Ciclo de vida de un componente

- Un componente React tiene una serie de métodos que React llama en un orden predefinido
- Ya hemos utilizado **getInitialState** y **getDefaultProps**
- El único obligatorio es **render** ya que sin ese método el componente no genera UI alguna
- Existen 3 momentos en el ciclo de un componente: mounting, updating, unmounting (creación, actualización, destrucción)

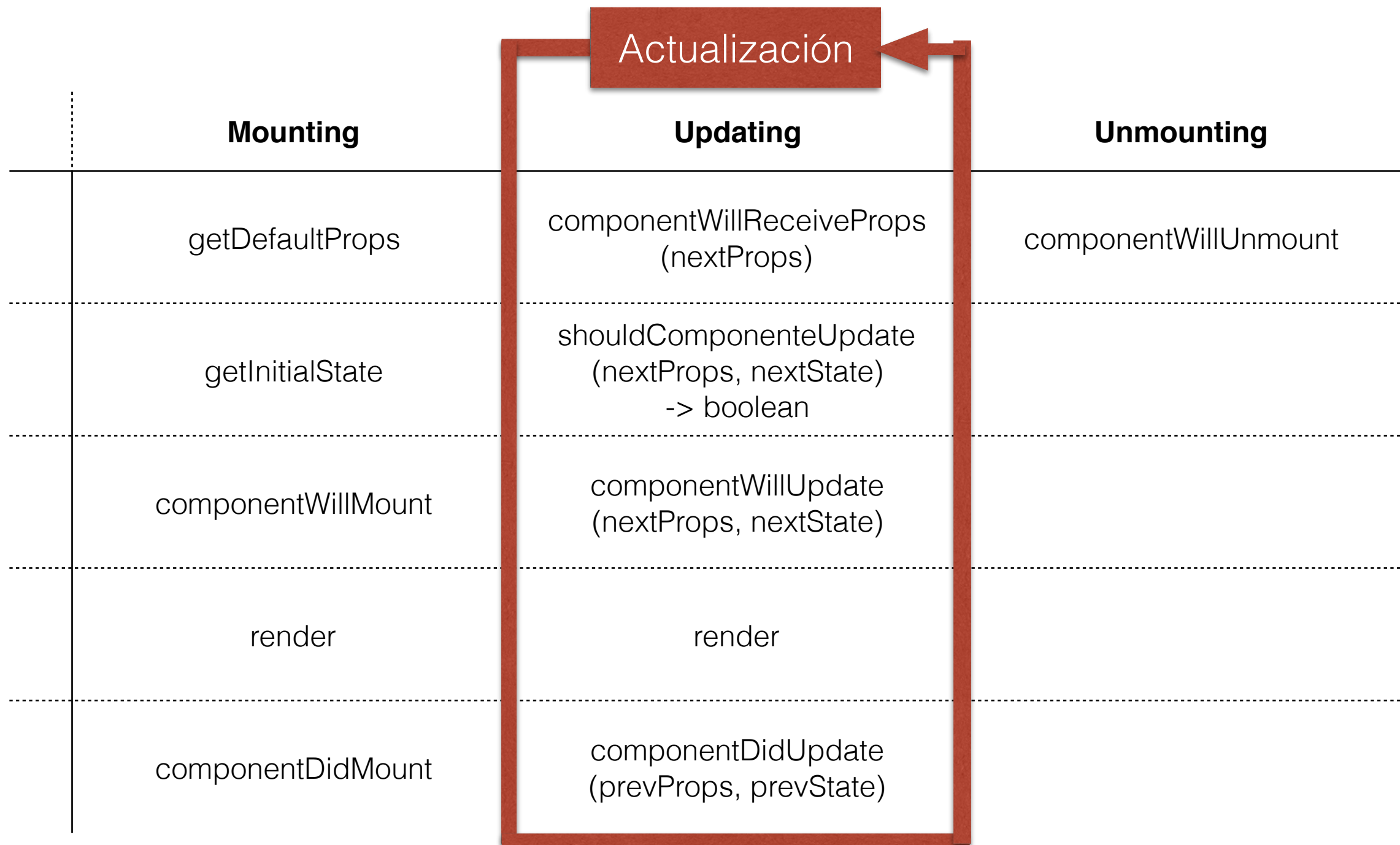
Ciclo de vida de un componente

	Mounting	Updating	Unmounting
	getDefaultProps	componentWillReceiveProps (nextProps)	componentWillUnmount
	getInitialState	shouldComponentUpdate (nextProps, nextState) -> boolean	
	componentWillMount	componentWillUpdate (nextProps, nextState)	
	render	render	
	componentDidMount	componentDidUpdate (prevProps, prevState)	

Ciclo de vida de un componente



Ciclo de vida de un componente



Ciclo de vida de un componente

			Destrucción
	Mounting	Updating	Unmounting
	getDefaultProps	componentWillReceiveProps (nextProps)	componentWillUnmount
	getInitialState	shouldComponentUpdate (nextProps, nextState) -> boolean	
	componentWillMount	componentWillUpdate (nextProps, nextState)	
	render	render	
	componentDidMount	componentDidUpdate (prevProps, prevState)	

Ciclo de vida de un componente

- El flujo en React siempre es unidireccional como se ve en la tabla anterior, en la etapa de actualización
- Dentro de **render** no modificamos ni propiedades ni estado, lo hacemos en los manejadores de eventos
- **render** es una función pura: dadas las mismas props y mismo estado devuelve exactamente lo mismo

Optimización

- Los métodos del ciclo de vida nos permiten influir en el mismo o conocer el momento actual de nuestro componente (¿estoy ya visible? ¿me he actualizado?)
- **shouldComponentUpdate** es el método con el que podemos *cancelar* una llamada a render devolviendo *false*. Por defecto devuelve *true*
- Tenemos acceso a las próximas *props* y próximo *state* por lo que podemos decidir que no necesitamos otro render
- De esta forma ahorramos a React que ejecute **render** para al final no cambiar nada

Mixins

- Utilizar herencia nos aporta cosas: comportamientos comunes, validaciones, etc
- Al usar composición, ¿perdemos estos beneficios?
- No, si usamos Mixins!

Mixins


- Un Mixin es simplemente un objeto JS con métodos, o una función que devuelve un objeto JS
- Este objeto será **fundido** por React con el código de nuestro componente, por lo que podemos utilizar métodos estándar como **getInitialState**, **setState**, etc. además de añadir nuestros propios métodos
- Dentro de **React.createClass** podemos incluir un Array de mixins bajo la clave **mixins** en la definición del componente

Mixins - definición

```
const notifyUpdateMixin = {  
  componentDidUpdate: function(prevProps, prevState) {  
    console.log('Component updated!');  
  },  
  customMethod: function(text) {  
    alert(text)  
  }  
}
```

```
const mixinWithOptions = function(n) {  
  return {  
    componentDidMount: function() {  
      console.log('Your component mixin has value', n);  
    }  
  }  
}
```

Mixins - uso



```
const MixinComponent = React.createClass({
  mixins: [notifyUpdateMixin, mixinWithOptions(25)],
  getInitialState: function() {
    return {
      clicks : 0
    };
  },
  handleClick: function(e) {
    this.customMethod('Boo!');
    this.setState({ clicks: this.state.clicks+1 });
  },
  render: function() {
    return (
      <div>
        <button onClick={ this.handleClick }>Click me</button>
        <p>Has hecho click { this.state.clicks } veces</p>
      </div>
    )
  }
});

export default MixinComponent;
```


Componentes de orden superior

- Otra forma de reutilizar código y separar responsabilidades
- Un componente de orden superior es una función que devuelve un componente generado “on the fly” que envuelve a otro, parametrizando sus **propiedades**
- Algo parecido al patrón Decorator “al estilo React”

Componentes de orden superior

- Se pueden utilizar como sustitutos de Mixin, especialmente para acceder a datos externos, etc.
- Veremos ejemplos en el siguiente tema

Componentes sin estado

- Nuevo en React 0.14+
- Definir componentes con una simple función de Javascript que equivale a **render** y cuyo argumento es el objeto **props**
- `componente = function(props)`

Componentes sin estado

ES5

```
function RandomNumber (props) {  
  return (<li>{ props.number }</li>)  
}
```

ES6 con arrow functions

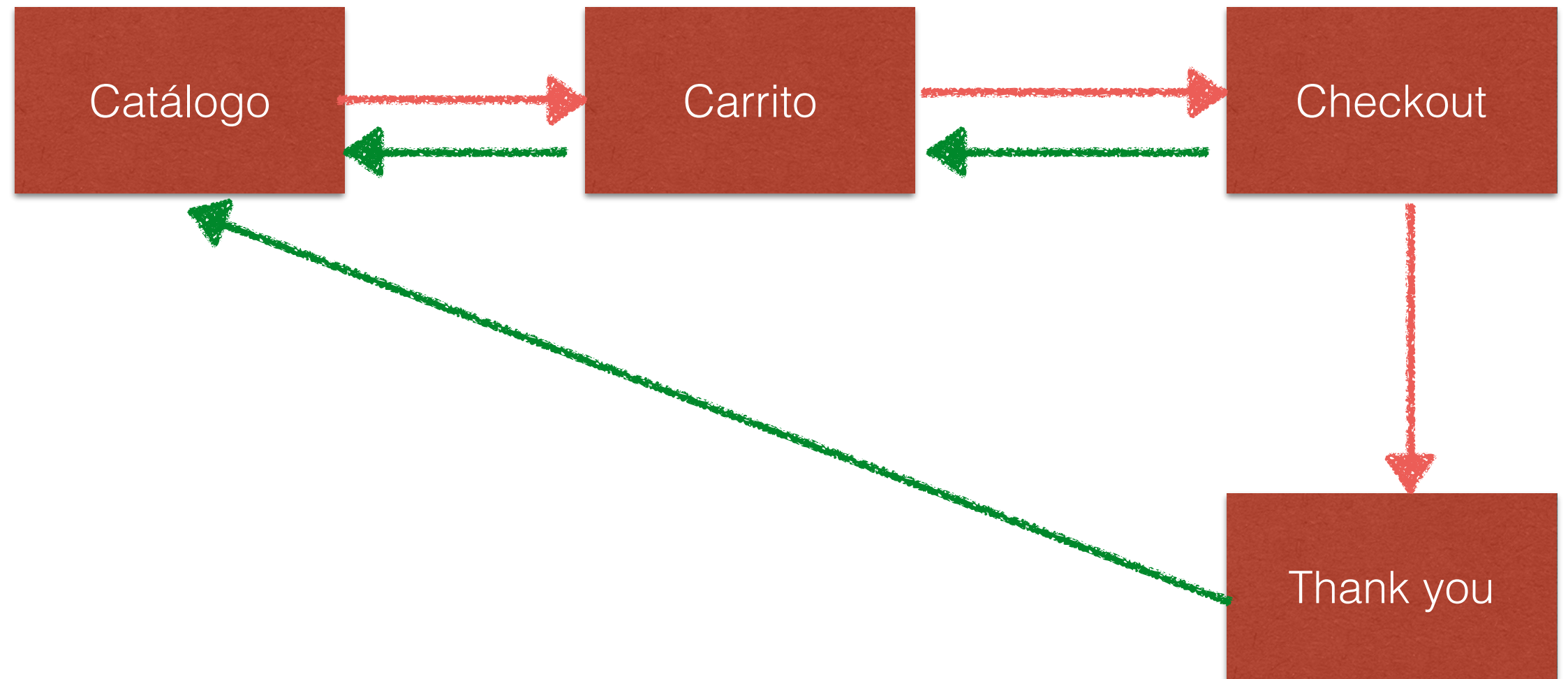
```
const RandomNumber = (props) => <li>{props.number}</li>;
```

Ejemplo en **components/random_list.js**

Ejercicio - Ecommerce

- Vamos a implementar una micro tienda que contiene diferentes pantallas:
 - Catálogo - se muestran los productos y se pueden añadir al carrito
 - Carrito - se muestran los productos escogidos, se manipula su cantidad y se vuelve al catálogo o se va al checkout
 - Checkout - se piden datos del usuario, se **validan** y, si es correcto, se va a la página de gracias
 - Confirmación - se muestra un mensaje de confirmación y se puede volver al Catálogo.

Ejercicio - Ecommerce



Ejercicio - Ecommerce

- Tendremos que mostrar la pantalla adecuada según el estado de nuestra tienda
- Podemos mantener una clave **page** en el estado del componente raíz
- La modificamos con **setState({ page: XXX })** cuando queramos navegar entre páginas
- La utilizamos para decidir qué componente pintar

Ejercicio - Ecommerce

```
render: function() {  
  return (  
    <div className="shopping-cart">  
      { this.getPageComponent(this.state.page) }  
    </div>  
  );  
}
```


Ejercicio - Ecommerce

```
getPageComponent: function (page) {  
  switch (page) {  
    case 'catalog':  
      return <Catalog  
        products={this.state.catalog}  
        onProductAdd={this.addProductToCart} />;  
    case 'cart':  
      return <Cart  
        products={this.state.cart}  
        onNavigate={this.setPage}  
        onItemQtyChange={this.changeCartItemQuantity}  
        onItemRemove={this.removeCartItem} />  
    case 'checkout':  
      return <Checkout onNavigate={this.setPage} onOrderPlaced={this.completeCheckout} />;  
    case 'thank-you':  
      return <ThankYou onNavigate={this.setPage} order={this.state.customerDetails} />;  
  }  
},
```

Ejercicio - Ecommerce

- Plantilla HTML disponible en **/ejercicios/tema3/src/plantillas/shoppingcart.html**
- Datos del catálogo en **/ejercicios/tema3/src/data/catalog.js**