



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React 02



JSX

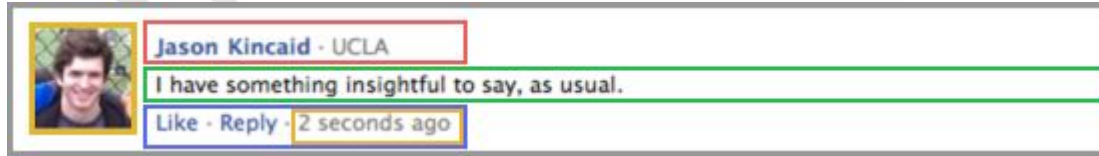
JSX sin etiquetas

```
render() {  
  return (  
    <React.Fragment>  
      <td>Hello</td>  
      <td>World</td>  
    </React.Fragment>  
  );  
}
```

Versión resumida

Versión con React.Fragment

```
render() {  
  return (  
    <>  
      <td>Hello</td>  
      <td>World</td>  
    </>  
  );  
}
```



Si tenemos los componentes arriba mostrados...

¿Como podemos hacer que la información que muestran sea dinámica?

Parámetros a componentes



Collapsible Content

Overview

Features

Hardware

Software



Collapsible Component with props

```
<Collapsible title="Overview" />
```

```
<Collapsible title="Features" />
```

```
<Collapsible title="Hardware" />
```

```
<Collapsible title="Software" />
```

```
<PrimerComponente name="Juancito"/>
```

```
export default class PrimerComponente extends React.Component {  
  render() {  
    const element = <h1>Hola {this.props.name}</h1>  
    return element;  
  }  
}
```


El paso de parámetros se realiza agregando un “atributo HTML” en el llamado al componente

El componente puede acceder al parámetro por medio de `this.props.<nombre del atributo>`



UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES



A los componentes realizados anteriormente, se le debe pasar como parámetro la información a mostrar...

- IMG: la dirección de la imagen (componente imagen)
- Title: el título a mostrar (componente titulo)
- Description: la descripción a mostrar (componente descripción)



Hola Juan Carlos

Hola Maria

Hola Gerardo



UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES

Podemos llamarlo así

```
<div className="App">
  <PrimerComponente name="Juan Carlos"/>
  <PrimerComponente name="Maria" />
  <PrimerComponente name="Gerardo" />
</div>
```



UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES

Mostrar un listado

Todo comienza por un vector...

```
var vectorNombres = [  
    'Juan Carlos',  
    'Maria',  
    'Gerardo'  
]
```

Y como lo mostramos

Opción 1: uso de array.forEach

```
var respuesta = [];  
vectorNombres.forEach(unItem => {  
    respuesta.push(<PrimerComponente name={unItem} />)  
})  
return (  
    <div className="App">  
        {respuesta}  
    </div>  
);
```

Opción 2: uso de array.map()

```
var vectorNombres = [  
  'Juan Carlos',  
  'Maria',  
  'Gerardo'  
]  
  
var respuesta = vectorNombres.map(unItem => {  
  return <PrimerComponente name={unItem} />  
})  
  
return (  
  <div className="App">  
    {respuesta}  
  </div>  
);
```







Ahora se debe mostrar el siguiente vector en el componente

```
vector = [  
  { url: 'https://placeimg.com/80/80/people?id=1', title: 'Titulo 1', description: 'Descripcion imagen 1'},  
  { url: 'https://placeimg.com/80/80/people?id=2', title: 'Titulo 2', description: 'Descripcion imagen 2'},  
  { url: 'https://placeimg.com/80/80/people?id=3', title: 'Titulo 3', description: 'Descripcion imagen 3'},  
  { url: 'https://placeimg.com/80/80/people?id=4', title: 'Titulo 4', description: 'Descripcion imagen 4'}  
]
```

Mostremos la hora actualizada





No funciona :-)



React no actualiza la interfaz
automáticamente

Entonces...



State al rescate

Todo aquello que deba modificar el estado de la interfaz, se debe incorporar en la propiedad state del componente

```
state = {  
  fecha: new Date()  
}
```

entonces...

```
this.state.fecha = new Date()
```



Todas las actualizaciones del estado se deben realizar por medio de...

```
this.setState({<propiedad>: <nuevo valor>});
```

Volvamos al ejercicio de la hora



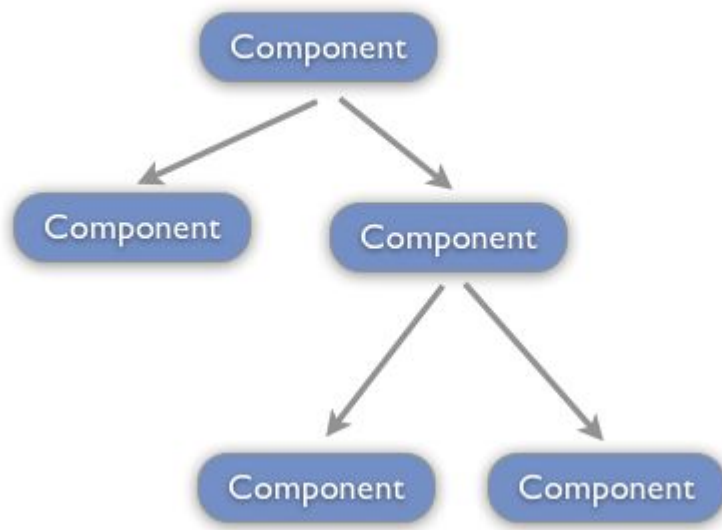
Eventos

Permite responder ante acciones del usuario

- Presiono un botón
- Completo un campo de un formulario
- Envió un formulario



Data



Events



UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES

Eventos más comunes en JS

- `onClick`: cuando se hace click sobre un elemento
- `onChange`: cuando se modifica el valor de un input

```
presionoBoton() {  
  console.log('Presino boton');  
}  
  
render() {  
  return <div>  
    {this.props.name}  
    <button onClick={this.presionoBoton}>Soy un botón</button>  
  </div>  
}
```

No se incluyen los () al indicar el método a utilizar al presionar el botón

Propiedad this

Cuando desde un método, deseamos acceder a una propiedad o método de la instancia utilizamos la propiedad this. Ejemplo

```
this.nombre = 'Victor';
```

```
this.llamarFuncionX();
```




Pero...
this no funciona en los eventos

You have my

undivided attention.

ICANHASCHEEZBURGER.COM



UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES

Cómo usamos this en los eventos

```
constructor(props) {  
  super(props);  
  this.agregar = this.agregar.bind(this);  
}  
  
agregar() {  
  this.setState({name: ''})  
}  
  
render() {  
  return (  
    <div>  
      <button onClick={this.agregar}>Agregar</button>  
    </div>  
  )  
}
```





UTN.BA

FACULTAD
REGIONAL
BUENOS AIRES

IMG	A Title
	<p>The description goes here.</p> <p>Current Date</p> <p>BTN</p>

Incorporar un botón al ejercicio, que cada vez que se presione muestre la fecha y hora actual dentro de la descripción