

# React.JS

## Introducción

Módulo 02

# Introducción Componentes

# Introducción

*"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 devuelven a React elementos que describen lo que debe aparecer en la pantalla."*

(Documentación Oficial)

# Componentes funcionales y de clase

La forma más sencilla de definir un componente es definir una función que reciba un objeto (props) y que retorne un elemento de React.

```
import React from 'react';

export default function Logo(props) {
  return (
    <div>MiEmpresa.Com</div>
  )
}
```

También podemos extender la clase `React.Component` para definir un componente basado en clases.

```
import React from 'react';

export default class Logo extends React.Component {

  constructor(props) {
    super(props);
  }

  render() {
    return(
      <div>MiEmpresa.Com</div>
    )
  }
}
```



Se pueden renderizar un componente como cualquier Elemento JSX

```
export class Nav extends React.Component {  
  constructor(props) {  
    super(props);  
  }  
  
  render() {  
    <nav>  
      <Logo></Logo>  
      <ul>  
        <li><a href="#">Item</a></li>  
      </ul>  
    </nav>  
  }  
}
```

# Buenas prácticas

## Composición

Consiste en agrupar funcionalidades comunes a varios componentes en un componente superior que las agrupe. Se usa cuando necesito reutilizar funcionalidades o estilos entre varios componentes.

## Extracción

Consiste en extraer una funcionalidad de un componente y, con ella, crear un nuevo componente. Se usa cuando un componente tiene más de una tarea o responsabilidad, entonces se divide en componentes más chicos.



# Comparación

	Componentes de clase	Componentes funcionales
<b>Props</b>	Se pasan como parámetro al constructor	Son el parámetro de la función
<b>State</b>	Es una variable ya provista, heredada de <b>React.Component</b>	Se debe acceder a través de la función 'hook' llamada <b>'useState()'</b>
<b>Ciclo de Vida</b>	Son los métodos de la clase, heredados de <b>React.Component</b>	Se pueden especificar a través de la función 'hook' llamada <b>'useEffect()'</b>
<b>Render</b>	Es un método de la clase	Es el retorno de la función



# Info y Links de interés

- <https://es.reactjs.org/docs/introducing-jsx.html>
- <https://es.reactjs.org/docs/components-and-props.html>
- <https://es.reactjs.org/docs/state-and-lifecycle.html>
- Los hooks son funciones que permiten a un componente funcional aprovechar las características de los componentes de clases. Si bien son un tema más avanzado, puedes echar un vistazo a <https://es.reactjs.org/docs/hooks-intro.html>  
También, en el [Curso de React Avanzado](#) se ven en detalle estas herramientas

# ¡Muchas gracias!

¡Sigamos trabajando!