



SOMOS ARBUSTA|

**React Js**

01 **ReactJS**

02 **Conceptos basicos**

03 **JSX**

04 **Componentes Funcionales y  
de Clase**

# CONTENIDOS|



# ¿ Qué es React.js ?



**Biblioteca de JavaScript de código abierto diseñada para crear interfaces de usuario.**

**Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos.**



# Ventajas de React



- **Fácil de aprender y usar**
- **Componentes reutilizables**
- **Crear Aplicaciones dinámicas**
- **Buen desempeño**



# Componentes



**Parte de la interfaz de usuario que es independiente y reutilizable**



# JSX



**React utiliza una sintaxis basada en XML (similar al HTML) para construir sus componentes usando el Virtual DOM. Esta sintaxis se transcompila a JavaScript (Tal como lo hacen TypeScript o CoffeeScript), por lo que en el fondo sólo es JavaScript presentado de una forma un poco más "visual".**

**React plantea otra forma de construir vistas, diferente a los templates de toda la vida. Si bien no es la idea, los templates separan tecnologías, no preocupaciones (separation of concerns). Los componentes de React buscan enfocarse en la separación de preocupaciones propiamente dicha. Por lo tanto, en un principio esto puede parecer "mezclar JavaScript y HTML", o más genéricamente, "mezclar lógica y presentación", pero en realidad estamos agrupando la lógica y la presentación relacionadas en módulos independientes. Separación de preocupaciones.**



# Componentes



## Funcionales

**Son más concisos y  
más fáciles de escribir,  
leer .**

## De Clase

**Funcion JS/ES6 => JSX**



# Ejemplo de Funciones



```
function Saludo(props){  
  return <h1>i Hola, {props.nombre}!</h1>;  
}
```





# Características

- **Debe retornar un elemento JSX**
- **Debe comenzar con una letra mayúscula**
- **Puede recibir valores si es necesario**



# Props



**Argumentos que puede recibir un componente de React.  
Componente dentro de otro componente (enviamos valores con los props).**



# Importante

- **Los props solo pueden ser enviados de “padre a hijo”**



# Componente de Clase



Clase de ES6 (JavaScript moderno) que retorna un elemento JSX.

```
class Saludo extends React.Component{  
  render(){  
    return <h1>iHola,{this.props.nombre}! </h1>;  
  }  
}
```



## Característica

- Debe extender `React.Component`. - Nos permite utilizar las propiedades de `react`-
- Debe tener un método `render()` para retornar un elemento de `JSX`
- Puede recibir valores si es necesario - `props` -



## Representación en JavaScript del conjunto de propiedades de un componente y sus valores actuales

**\*Propiedades se refiere a la información sobre el componente**



# Porque hay 2 componentes

**Antes se utilizaban componentes de clase para poder trabajar con “estados” de nuestros componentes.**

Luego llegaron los Hooks



# Practica Componente funcional



```
import "bootstrap/dist/css/bootstrap"

function Cabecera(){
  return (<header className="alert alert-info text-center">
    <h1 className="display-1"> Bienvenidos a la
clase cinco de REACT!! </h1>
    </header>)
}
export default Cabecera;
```







**ARBUSTA**

[/arbustait](#)



**Gracias  
por participar**

[WWW.ARBUSTA.NET](http://WWW.ARBUSTA.NET)