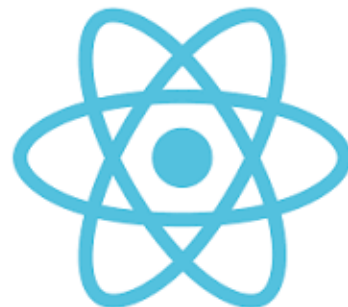


Agencia de  
Aprendizaje  
a lo largo  
de la vida

# ESPECIALIZACIÓN REACT





Agencia de  
Aprendizaje  
a lo largo  
de la vida

## Clase 04. REACT JS

### *COMPONENTES 1*

# COMPONENTES

# COMO CREAR UN COMPONENTE EN REACT



Los componentes en React son funciones o clases que devuelven un elemento de React. Hoy en día lo más recomendado es usar funciones:

```
function HolaMundo() {  
  return <h1>Hola Mundo!</h1>  
}
```

Pero también puedes usar una clase para crear un componente React:

Lo importante es que el nombre de la función o clase empiece con una letra mayúscula. Esto es necesario para que React pueda distinguir entre componentes y elementos HTML.

```
import { Component } from 'react'  
  
class HelloWorld extends Component {  
  render() {  
    return <h1>Hello World!</h1>  
  }  
}
```

# ¿QUE SON LAS PROPS?

Las props son las propiedades de un componente. Son datos que se pasan de un componente a otro. Por ejemplo, si tienes un componente Button que muestra un botón, puedes pasarle una prop text para que el botón muestre ese texto:

```
function Button(props) {  
  return <button>{props.text}</button>  
}
```

Podríamos entender que el componente Button es un botón genérico, y que la prop text es el texto que se muestra en el botón. Así estamos creando un componente reutilizable.

Debe considerarse además que al usar cualquier expresión JavaScript dentro de JSX debe envolverlos con {}, en este caso el objeto props, de otra forma JSX lo considerará como texto plano.

Para usarlo, indicamos el nombre del componente y le pasamos las props que queremos:

```
<Button text="Haz clic aquí" />  
<Button text="Seguir a @messi" />
```

# ¿QUE SON LAS PROPS?

Las props son una forma de parametrizar nuestros componentes igual que hacemos con las funciones. Podemos pasarle cualquier tipo de dato a un componente, incluso otros componentes.

## ¿Qué diferencia hay entre props y state?

Las props son un objeto que se pasan como argumentos de un componente padre a un componente hijo. Son inmutables y no se pueden modificar desde el componente hijo.

El state es un valor que se define dentro de un componente. Su valor es inmutable (no se puede modificar directamente) pero se puede establecer un valor nuevo del estado para que React vuelva a renderizar el componente.

Así que mientras que tanto props como state afectan al renderizado del componente, su gestión es diferente.

# RENDERIZADO CONDICIONAL

# ¿Qué es el renderizado condicional?

El renderizado condicional es la forma de mostrar un componente u otro dependiendo de una condición.

Para hacer renderizado condicional en React usamos el operador ternario:

```
function Button({ text }) {  
  return text  
    ? <button>{text}</button>  
    : null  
}
```

En este caso, si la prop text existe, se renderiza el botón. Si no existe, no se renderiza nada.



# ESTILOS EN REACT

## ¿Cómo puedes aplicar clases CSS a un componente en React?

Para aplicar clases CSS a un componente en React usamos la prop `className`:

Para hacer renderizado condicional en React usamos el operador ternario:

```
function Button({ text }) {  
  return (  
    <button className="button">  
      {text}  
    </button>  
  )  
}
```

La razón por la que se llama `className` es porque `class` es una palabra reservada en JavaScript. Por eso, en JSX, tenemos que usar `className` para aplicar clases CSS.

## ¿Cómo puedes aplicar estilos en línea a un componente en React?

Para aplicar estilos CSS en línea a un componente en React usamos la prop `style`. La diferencia de cómo lo haríamos con HTML, es que en React los estilos se pasan como un objeto y no como una cadena de texto (esto puede verse más claro con los dobles corchetes, los primeros para indicar que es una expresión JavaScript, y los segundos para crear el objeto):

```
function Button({ text }) {  
  return (  
    <button style={{ color: 'red', borderRadius: '2px' }}>  
      {text}  
    </button>  
  )  
}
```

Fíjate que, además, los nombres de las propiedades CSS están en camelCase.

## ¿Cómo puedo aplicar estilos de forma condicional a un componente en React?

Puedes aplicar estilos de forma condicional a un componente en React usando la prop `style` y un operador ternario:

```
function Button({ text, primary }) {  
  return (  
    <button style={{ color: primary ? 'red' : 'blue' }}>  
      {text}  
    </button>  
  )  
}
```

## ¿Cómo puedo aplicar estilos de forma condicional a un componente en React?

En el caso anterior, si la prop `primary` es `true`, el botón tendrá el color rojo. Si no, tendrá el color azul.

También puedes seguir la misma mecánica usando clases. En este caso, usamos el operador ternario para decidir si añadir o no la clase:

```
function Button({ text, primary }) {  
  return (  
    <button className={primary ? 'button-primary' : ''}>  
      {text}  
    </button>  
  )  
}
```