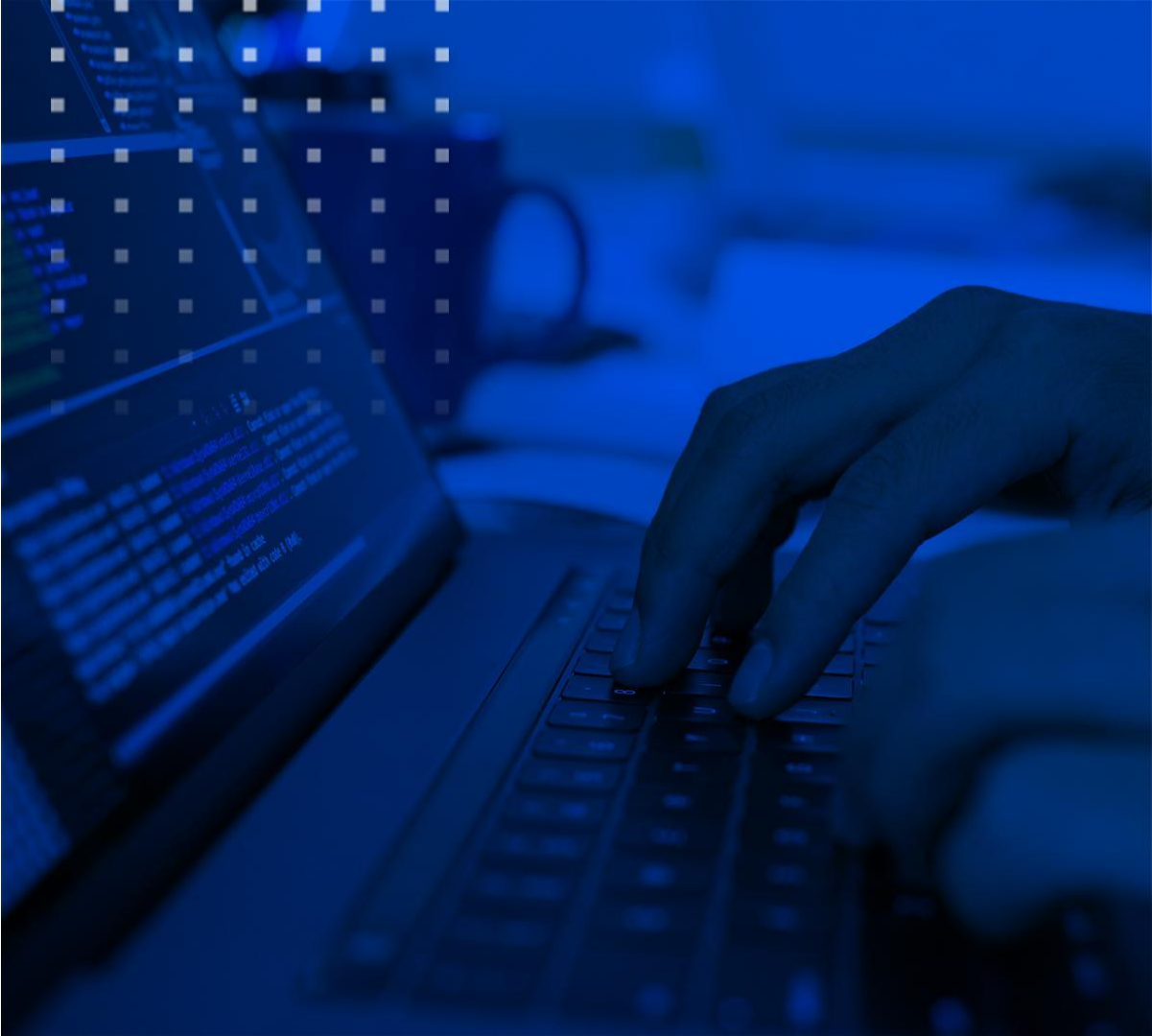


CURSO

FULL STACK DEVELOPER NIVEL INICIAL

UNIDAD 19
JavaScript
Componentes y Props



JSX

JSX

Seguramente notaron que el código 'html' que aparece en nuestra app recién creada es algo raro...

JSX

Seguramente notaron que el código 'html' que aparece en nuestra app recién creada es algo raro...

Eso es porque **no es HTML**, sino JSX.

JSX

JSX es una extensión de la sintaxis de Javascript que se **parece a HTML**. No es ni una cadena de caracteres, ni HTML.

Es una extensión que permite hacer llamadas a funciones y a construcción de objetos.

JSX

JSX es una **extensión de Javascript**, no de React.

Esto significa que **no hay obligación** de utilizarlo, pero es

recomendado en el sitio web oficial de React.

Al final, JSX **se transforma en código JavaScript**.

JSX

Esto nos da algunas **ventajas** como:

- ★ Ver errores en tiempo de compilación
- ★ Asignar variables
- ★ Utilizar métodos

COMPONENTES

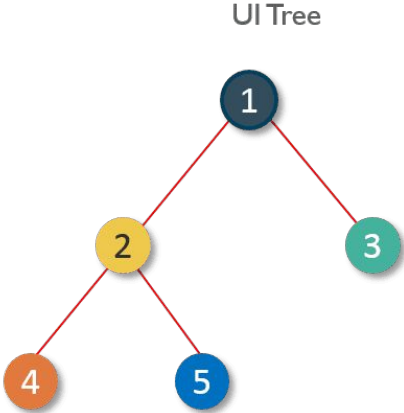
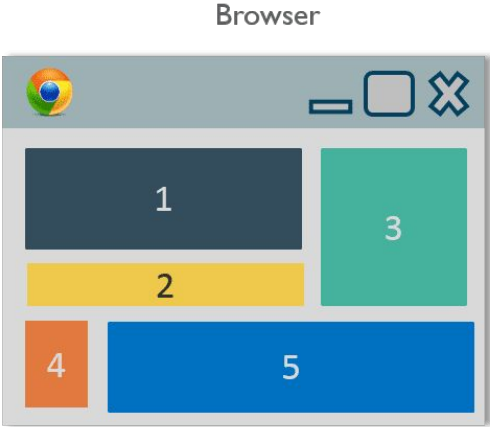
COMPONENTES

Las aplicaciones en React básicamente **se construyen mediante componentes.**

Así, podemos crear aplicaciones completas de una manera modular y de fácil mantenimiento.

COMPONENTES

Seria algo asi...



COMPONENTES

Los componentes permiten separar la interfaz de usuario en piezas **independientes, reutilizables** y pensar en cada pieza de forma **aislada**. La ventaja más evidente de los componentes es que podemos reutilizarlos siempre que queramos (¡como cuando creamos funciones JS!)

COMPONENTES

Vamos a trabajar con **componentes basados en funciones**. Se ven más o menos así:

```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <img src={logo} className="App-logo" alt="logo" />  
        <p>  
          Hello, World!  
        </p>  
      </header>  
    </div>  
  );  
}
```

```
import React from 'react';  
import './Intro.css';
```

```
const Intro = () => {  
  return (  
    <p  
      className="texto-intro">  
      ¡Hola!  
    </p>  
  );  
}  
export default Intro;
```

COMPONENTES

Puntos a tener en cuenta:

- ★ **Importar React** al principio del componente
- ★ Se pueden usar **funciones flecha**
- ★ Dentro del **return** va nuestro código **JSX**
- ★ Hay que **exportar** el componente para que pueda ser reutilizado

¡A Practicar!

- Crear una carpeta llamada **/components** y dentro una subcarpeta llamada **/intro** con un archivo **intro.js** e **intro.css**
- Deberá contener un párrafo con una introducción a tu sitio web.
- Importar el componente **intro** en tu `app.js`

COMPONENTES > PROPS

COMPONENTES

Propiedades

Las propiedades son la forma que tiene

React para pasar parámetros de un

componente superior a sus **children**. Como

vemos, es un flujo de datos unidireccional

COMPONENTES

Propiedades

```
const Intro = ({texto}) => {  
  return (  
    <p className="texto-intro">  
      {texto}  
    </p>  
  );  
}  
export default Intro;
```

Se reciben como **parámetro** de la función del componente.

COMPONENTES

Propiedades

```
const Intro = ({texto}) => {  
  return (  
    <p className="texto-intro">  
      {texto}  
    </p>  
  );  
}  
export default Intro;
```

Se reciben como **parámetro** de la función del componente.

Es decir, un componente superior se lo envía al children. Por ejemplo: app le podría enviar la prop “texto” a intro.

COMPONENTES

Propiedades

```
function App() {  
  return (  
    <Intro texto="¡¡Hola!!" />  
  );  
}
```

Entonces el componente app le estaría enviando el valor del parámetro (prop) texto a su componente children intro.

COMPONENTES

Las **props** se reciben del **componente padre** y son sólo de lectura



¡A Practicar!

Con el componente intro que creamos previamente:
convertir ese texto de bienvenida en una prop que
reciba desde app.js

¡A Practicar!

1. Dentro de la carpeta /components vamos a crear cuatro subcarpetas:
 - ★ navbar
 - ★ footer
 - ★ contacto
 - ★ cv
2. Cada una de estas carpetas tiene que tener su respectivo archivo JS y CSS. Teniendo en cuenta que el CSS no es obligatorio en todas.
3. Recordá que cada archivo JS es un componente por lo que deberá tener una **función** que **devuelva** un JSX y un **export** de esa función.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES