educación IT
CODE YOUR FUTURE

ReactJS Developer

Módulo 2

educación IT
CODE YOUR FUTURE

Renderizando elementos

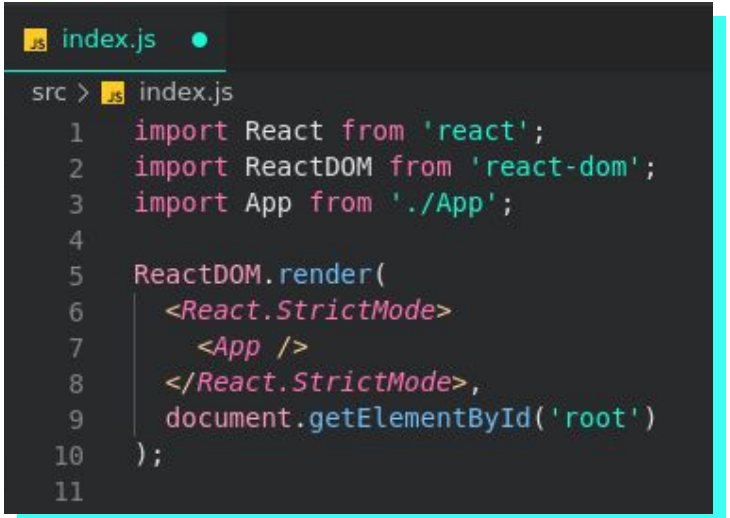
Mostrando componentes

Todo empieza en un solo lugar, no importa cuál sea el alcance de tu aplicación React, si estás haciendo un home banking, tu portfolio personal o una lista de compras. Toda aplicación React empieza en **un solo y único componente.**

En el archivo *index.js*, en el punto de entrada del programa, vemos la llamada a la función `render` de `ReactDOM`. Lo que hace esta función es montar **un componente en un elemento *HTML***.

Ese componente montado es **el componente raíz**. Observa que hay un solo nodo raíz (en este caso es `React.StrictMode`).

Como toda aplicación React se monta desde un único componente, la forma de mostrar elementos será añadiéndolos al árbol de componentes.



Paso a paso para añadir un componente al árbol de componentes

1. Define el **elemento** que que quieres mostrar. Con React podemos renderizar componentes personalizados así como cualquier otra etiqueta *HTML*.
2. Define **dónde** quieres insertarlo, en qué parte del árbol: ¿Se trata de un componente que deba envolver a todos? ¿Es un ícono específico de un menú? ¿Dónde se mostrará esto?
3. Ve hacia el archivo **donde quieres insertar el elemento.**
4. Si quieres insertar un componente personalizado, asegúrate de primero **importarlo.**
5. Ve hacia la línea **donde quieres insertar el elemento.**
6. Escribe el nombre del componente **como si fuese una etiqueta *HTML*.**
7. Puedes pasarle **props como si fuesen atributos *HTML*.**
8. Un componente puede tener componentes hijos y serán pasados todos a través de la `prop children`.

```
function App() {
  return (
    <div className="fondo">
        <div>
            <h1>Mi aplicación</h1>
            <p>
                Hola mundo
            </p>
        </div>
    </div>
  );
}
```

```
function App() {
  return (
    <div className="fondo">
        <main>
            <h1>Mi aplicación</h1>
            <section>
                Hola mundo
            </section>
        </main>
    </div>
  );
}
```

```
import Contacto from "./components/contacto/contacto.component";
import Divider from "./components/divider/divider.component";
import "./fondo.css";

function App() {
  return (
    <div className="fondo">
        <Divider></Divider>
        <Contacto></Contacto>
        <Divider></Divider>
    </div>
  );
}
```

Una herramienta muy útil de *JSX* son **las llaves simples ({ y })**. Dentro de ellas podemos renderizar cualquier cosa (componentes, variables, etc.) **en una sola línea.** En otras palabras: podemos usar cualquier código *JS* / *JSX* dentro de estas llaves siempre y cuando nos ocupe **una sola sentencia de código** (no podemos hacer un `if-else`, pero sí un ternario, por ejemplo).

```
function App(props) {
  return (
    <div className="fondo">
        { props.isAuth ? "logueado" : "no logueado" }
    </div>
  );
}
```

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!