

PROGRAMACIÓN EN NUEVAS TECNOLOGÍAS 2

React: Components, Props, States, Styles

UNIDAD PREVIA

- ES6
- Funciones Closures
- Funciones de primer orden
- Pila de Ejecucion
- Asincronicidad
- Callbacks
- Promesas Async/Await

REACT

- Nos permite escribir vistas que “reaccionan” a cambios en el modelo de datos.
- Nos permite abstraer problemas complejos en componentes pequeños.
- Nos permite escribir código más simple que mantenga un patrón de diseño y siga siendo performante a nivel FrontEnd.

CÓDIGO IMPERATIVO VS DECLARATIVO

- Como vs Que.
- La programación Imperativa, describe una serie de pasos para llegar a lo que se quiere.
- La programación Declarativa, se declara directamente lo que se quiere.

REACT ES CÓDIGO DECLARATIVO

- Código imperativo vs declarativo.
- Las APIs de los navegadores pueden llegar a ser tediosas de trabajar de forma imperativa.
- React nos permite escribir lo que queremos y la librería se encargará de realizar la manipulación del Objeto DOM.

COMPONETIZACION

- React es fácilmente componetizado.
- Se puede dividir problemas complejos en componentes aislados y autónomos.
- Se puede reutilizar estos componentes:
 - Consistencia
 - Velocidad de Iteración
- Esta naturaleza declarativa permite la customización de componentes ya existentes.

PERFORMANCE

- Escribimos lo que queremos hacer, y react se encargará del trabajo tedioso.
- Renderización: Es el proceso por el cual React sincroniza los cambios en el **estado** de la aplicación al DOM.
 - Reconstruye el Virtual DOM
 - Solo aplica el cambio del estado del componente.

CODIFICANDO EN REACT

- JSX.
 - Sintaxis extendida de Javascript parecida al XML.
 - Transpila instrucciones a Javascript.
 - Etiquetas en minúsculas son tratados como HTML/SVG, Mientras que etiquetas con mayúscula son tratados como Componentes.
- Funciones como componentes.
 - Retorna un nodo (Puede ser una renderización, <div>).
 - Recibe un objeto de las propiedades que son pasadas al elemento HTML.

PROPS

- Son propiedades pasados como un objeto a un componente y utilizado para renderizado o acción dentro de mi componente.
- Los cambios en estos props causarán una ejecución nueva para devolver el nodo. Es decir, una nueva renderización.
- A diferencia de HTML, estos pueden ser cualquier valor Javascript

STATES

- Añade un manejo interno de la configuración de un componente.
- Un state, es una propiedad de identificación del componente.
- Solo puede ser actualizado invocando su método hook.
 - Implementado en el componente `useState()`
 - Las llamadas a métodos modificadores son encoladas y llamadas asincrónicamente.
 - Pasa un objeto a ser mergeado, o una función del estado anterior (`previous state`).
- Cambios en los **states** también causan un renderizado del componente.



Educando para la vida

todoApp.js

¿Solo nos limitamos al
ámbito WEB?

REACT NATIVE

- Framework basado en React.
- Nos permite construir aplicaciones móviles usando solo código Javascript.
 - “Learn once, write anywhere”
- Soporta iOS y Android.