Clase 1 – ReactJS

# ¿Qué es?

Slide 2

Es una de las librerías, creada por Facebook, mas populares de JavaScript para el desarrollo de aplicaciones web y móviles. Se fundamenta en la creación de fragmentos o bloques reutilizables llamados Componentes, para crear interfaces.

Cabe recalcar que ReactJS no es un framework como si lo son otras alternativas para la creación de web como Angular o Vue. Estos últimos definen un marco de trabajo, estructura de carpetas e interacciones entre servicios y componentes bien definidas. Por otro lado, ReactJS solo se enfoca en la renderización de los Componentes creados por el desarrollador. Dejando de libre elección el patrón de diseño o arquitectura a usar para los demás aspectos de una aplicación web como lo son API’s, Validaciones, etc.

# ¿Por qué escogerlo?

Slide 3 - 6

* Gran comunidad. Al ser el más popular, esto conlleva a que tener una gran comunidad detrás de su desarrollo, mejora y creación de nuevas librerías para agilizar la implementación de componentes. Como puede ser el caso de librerías de Mapas, Gráficos 2D y 3D, manejo de formularios, manejo de Store Global y más.
* Fácil curva de aprendizaje. Al escoger a JSX (JavaScript XML) como sintaxis para la construcción de componentes, facilita la curva de aprendizaje. Debido a que JSX es una extensión para JavaScript que le permite crear componentes que luego serán compilados y renderizados como etiquetas HTML.
* Desarrollo web y móvil (nativo). Además de aplicaciones web, permite un acercamiento más sencillo al desarrollo de aplicaciones nativas. Esto se puede lograr usando el framework Cordova, el cual permite embeber nuestra aplicación web y empaquetarla para sistemas operativas Android o iOS. Sin embargo, también tiene un enfoque más nativo llamado React Native, este último tiene ciertas variaciones en la manera de declarar y usar ciertas etiquetas, pero reutiliza las bases principales de ReactJS como lo son los Hooks, JSX y el ciclo de vida de componentes que conoceremos más adelante.

Captura de HTML vs React Component

# Componentes

En la actualidad toda aplicación, ya sea web o móvil, se empieza a diseñar por secciones, páginas y bloques. Es aquí donde se empiezan a definir las bases de cada componente, sus funcionalidades, estilos e interacción con la UI o los datos de nuestra aplicación.

En ReactJS los bloques de nuestra aplicación son los Componentes, en ellos desarrollaremos, de manera declarativa, como se renderizará el componente, si hace uso o no de “props” y sus diferentes interacciones.

Diagrama de estructura básica de una página web  
Header, Navbar, Articles, Footer

# Primer componente

Crear un componente que renderice una imagen aleatoria desde la API <http://avatar.iran.lieara.run/public>. Utiliza el componente en el archivo de entrada de ViteJs (main.tsx)

Añadir props obligatorios al componente para enviar la url y el nombre. Finalmente, renderizar el componente desde un Array de objetos con las claves url y name.