UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Centro de e-Learning

Diplomatura en programación web full stack con React JS

Centro de e-Learning SCEU UTN - BA.
Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 5: React Inicial

Unidad 1: Fundamentos de React JS, Instalación, línea de comando de React, estructura de un proyecto, Introducción a JSX

Presentación:

Habiendo concluído con el servidor, ahora vamos a aprender como hacer una aplicación del lado del cliente que consuma los recursos de nuestro servidor. NodeJS. ReactJS es una herramienta muy popular actualmente para construir vistosas y potentes interfaces de usuario, así que comencemos!

Objetivos:

Que los participantes:

- Comprendan los fundamentos de ReactJS.
- Sean capaces de realizar la instalación de la herramienta.
- Obtengan nociones básicas sobre el lenguaje JSX.

Bloques temáticos:

1. Fundamentos de ReactJS
2. Instalación
3. Línea de comandos de React
4. Estructura de un proyecto React
5. Introducción a JSX
6. Ejemplo
7. Trabajo Práctico

Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

**** El MEC es el modelo de E-learning colaborativo de nuestro Centro.***

Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

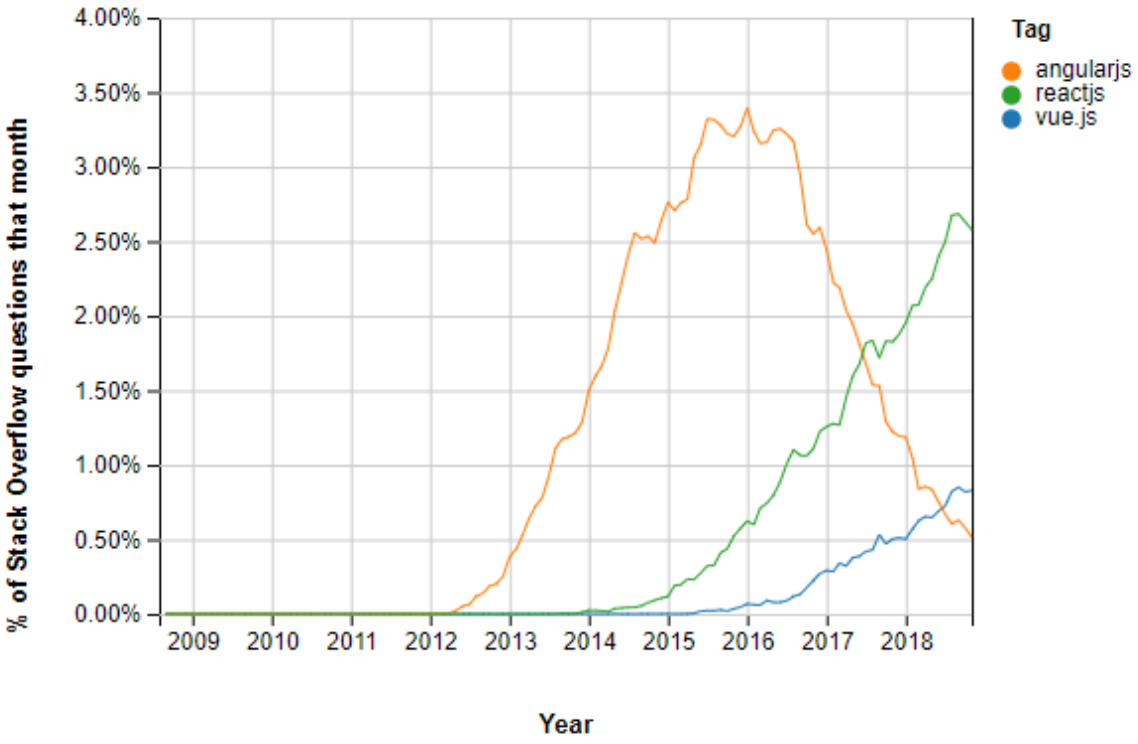
1. Fundamentos de ReactJS

Definición

Es una biblioteca JavaScript para construir interfaces de usuario, creada por Facebook para su plataforma, y actualmente utilizada por grandes empresas y proyectos como Instagram, New York Times, Netflix (para algunas de sus versiones), WhatsApp web, y Dropbox.

Beneficios

React es un framework muy popular el cual compite con Angular (Google) y Vue. La popularidad de React se puede ver reflejada en la cantidad de consultas que se han realizado en Stack Overflow sobre la tecnología durante los últimos años.



El ser una herramienta popular, implica que existe mayor cantidad de usuarios y por ende es utilizado en proyectos diversos permitiendo con mayor seguridad poder encontrar la respuesta a un problema que se nos presente en una implementación.

Además de la popularidad, React es un framework muy rápido, sobrepasando en los tests a Angular. Esto hace que sea una herramienta muy interesante para poder ofrecer una mejor experiencia de uso al usuario.

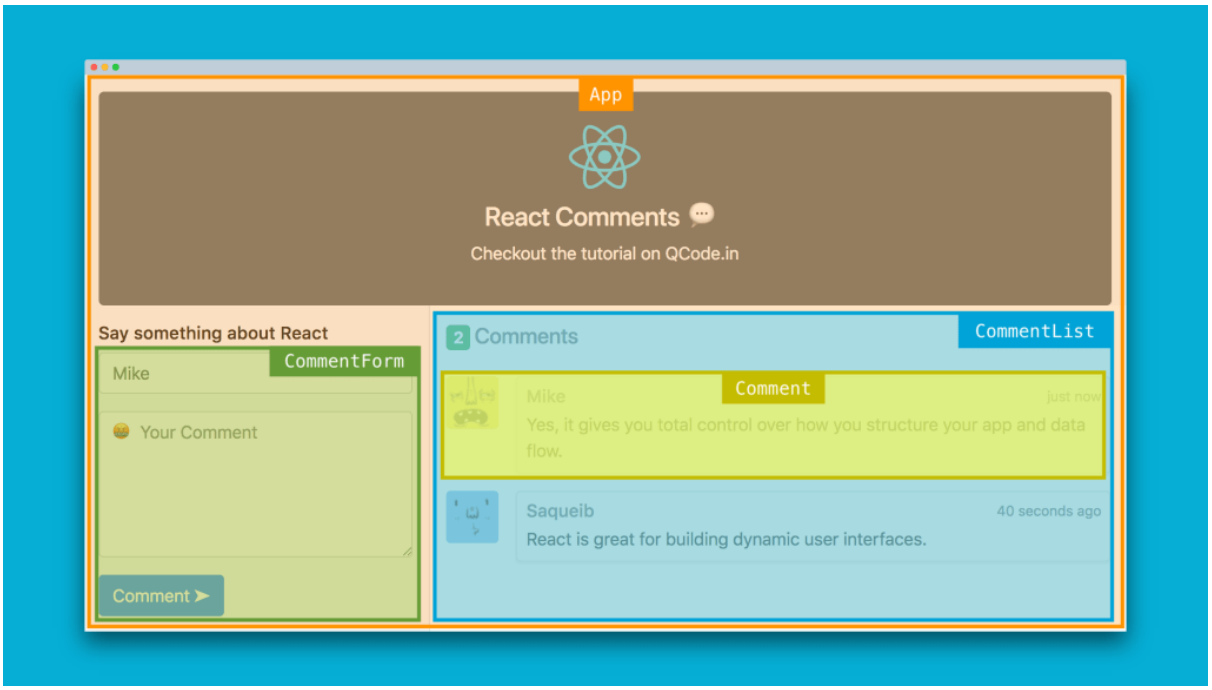
React se basa en HTML y JavaScript, por lo que con conocimiento de esas dos tecnologías hay mucho camino que ya tenemos recorrido en el proceso de aprendizaje en React.

Fundamentos del lenguaje

Componentes

React está basado en componentes. Entendemos componentes como pequeños fragmentos visuales, que la suma de los mismos integran una aplicación.

Ejemplo



En la imagen podemos apreciar diferentes componentes. Un componente principal App, y varios componentes que lo integran. Realicemos un árbol/estructura de cómo se componen los elementos.

- App
  - CommentForm
  - CommentList
    - Comment

En este caso, tenemos un componente principal App, el cual tiene 2 componentes hijos (CommentForm y CommentList).

A su vez, el componente CommentList, tiene un componente hijo (Comment)

Cada componente se encarga de como se mostrará un pequeño fragmento de la página.

- El componente Comment se encarga de como se verá un único comentario.
- El componente CommentList se encarga de como se mostrará la lista de comentarios, y delega la visualización de cada comentario al componente Comment.
- El componente CommentForm se encarga de ofrecer al usuario un formulario de ingreso de comentarios.

Componentes en la vida real

Ejemplo

Aquí podemos identificar algunos posibles componentes:

- La tarjeta (card) que contiene toda la información
- La imágen principal
- Botones (me gusta, comentar, compartir)
- Comentarios
    - Un comentario
- Input (campo de texto) para agregar un comentario

2. Instalación

Para instalar React previamente debemos tener instalado NodeJS. En caso que no tengas NodeJS instalado, lo podés descargar desde la página oficial de NodeJS.

Una vez que nos aseguramos de tener NodeJS instalado, debemos abrir una terminal/consola en nuestra computadora (Windows símbolo de sistema, MAC y Linux una terminal). Ya en la terminal, debemos escribir el siguiente comando

```
npm install -g create-react-app
```

El cual instalará una aplicación que nos permite crear un nuevo proyecto en React. Esta operación puede demorar unos minutos y necesita de una conexión a internet.

3. Línea de comandos de React

Una vez instalada la aplicación de generación de proyectos en React (create-react-app), y manteniéndonos en la consola, debemos dirigirnos a la carpeta de nuestra computadora en al cual deseamos crear el proyecto React. Y en dicha carpeta escribir

```
create-react-app <nombre de la aplicación>
```

Reemplazaremos <nombre de la aplicación> por el nombre que deseemos asignarle a nuestro proyecto. Esta operación puede demorar unos minutos.

Ejemplo

```
create-react-app mi-primer-proyecto
```

Ingresamos a la carpeta del proyecto recién generado

```
cd <nombre de la aplicación>
```

Y por último podemos ver el proyecto en funcionamiento con la siguiente línea en la consola

```
npm start
```

Con esta simple línea de comando se abrirá un navegador y podremos ver la aplicación inicial en funcionamiento.

Cada vez que deseemos ver el proyecto, debemos ejecutar la sentencia **npm start** para iniciar el proyecto. Mientras el proyecto se encuentra corriendo, actualiza automáticamente en el navegador, todos los cambios que realicemos en el código fuente. Esta forma de trabajo es muy útil para poder ver en el momento el impacto de nuestros cambios.

4. Estructura de un proyecto React

El proyecto creado por la aplicación create-react-app es el siguiente

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   └── manifest.json
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    ├── logo.svg
    └── serviceWorker.js
```

En el cual podemos encontrar 2 carpetas principales

- public: en la cual se encuentran los archivos que no son de programación (html, imágenes)
- src: en la cual se encuentra todo nuestro proyecto. Todo aquello que programaremos en nuestro proyecto (archivos JS con los componentes, archivos de estilos)

Archivo public/index.html

Es el archivo sobre el cual corre el proyecto React, tiene toda la estructura básica de un archivo HTML y debemos incorporar todas aquellas etiquetas que NO deseamos que sean modificadas en nuestro proyecto. El elemento principal que tiene este archivo es un div cuyo id es root, que es el punto de partida del proyecto React (dentro de este div es donde se mostrarán todos los componentes React)

```
<div id="root"></div>
```

Archivo src/index.js

Es el punto de entrada de JavaScript para el proyecto React. Este archivo se encarga de mostrar el componente **App.js** en el ** ** del archivo HTML principal

```
import App from './App';
ReactDOM.render(<App />, document.getElementById('root'));
```

La llamada a **ReactDOM.render** recibe como primer parámetro el componente a mostrar (**App**) y como segundo parámetro en qué lugar mostrar dicho elemento (en la etiqueta html cuyo id sea **'root'**)

Archivo src/App.js

Este es el archivo que contiene el componente principal de la aplicación. A partir de este componente se genera la interfaz de la aplicación. El "return" de la función es el encargado de generar la Interfaz del componente.

Ejemplo

```
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Mi primer aplicación React</h1>
    </div>
  );
}

export default App;
```

Los () en el return sirven para indicar que vamos a estar retornando un fragmento de código.

Este archivo generará la interfaz de la aplicación, generando el siguiente código HTML que será interpretado por el navegador y most

```
<div id="App">
    <div class="App">
        <h1>Mi primer aplicación React</h1>
    </div>
</div>
```

5. Introducción a JSX

Es una extensión de la sintaxis de JavaScript con la cual se generan las interfaces gráficas en React.

Es similar a un lenguaje de plantillas (templates) pero que tiene todo el potencial de JavaScript.

JSX permite “mezclar” código HTML con código JavaScript. Por ejemplo, permite asignar un fragmento HTML a una variable JavaScript y luego mostrar dicha variable.

Ejemplo

```
const variableAMostrar = <h1>Título de la página</h1>;
```

A pesar de parecer código HTML puro lo que está del lado derecho de la asignación, es código JSX, el cual tiene mucho más potencial que el simple código HTML.

También podemos mostrar variables JavaScript dentro del código HTML

Ejemplo

```
const curso = 'React inicial - 101';
const variableAMostrar = <h1>Bienvenido al curso {curso}</h1>
```

Este código generará el HTML

```
<h1>Bienvenido al curso React inicial - 101</h1>
```

Para mostrar el componente, debemos incluir el mismo dentro del return de la función, la que debe retornar el JSX que se mostrará

Ejemplo

```
function Prueba() {
    const curso = 'React inicial - 101';
    const variableAMostrar = <h1>Bienvenido al curso {curso}</h1>;
    return variableAMostrar;
}
```

Y el ejemplo completo del componente es

```
import React from 'react';

export default function PrimerComponente() {
        const curso = 'React inicial - 101';
        const variableAMostrar = <h1>Bienvenido al curso {curso}</h1>
        return variableAMostrar;
}
```

En donde el nombre del componente es PrimerComponente y el mismo mostrará por pantalla (al momento de llamarse)

```
<h1>Bienvenido al curso React inicial - 101</h1>
```

¿Sabías que?

Todo el código JSX debe estar contenido dentro de una etiqueta, entonces si disponemos del siguiente código

```
<div>
      <p>Uno</p>
</div>
<div>
      <p>Dos</p>
</div>
```

El mismo no es válido en JSX, porque no tenemos un único elemento “padre” sino que tenemos 2 elementos que agrupan el contenido, lo cual no es válido en JSX.

Por lo que debemos modificar el código a la siguiente forma

```
<div>
      <div>
            <p>Uno</p>
      </div>
      <div>
            <p>Dos</p>
      </div>
</div>
```

Aquí sí tenemos un único elemento padre, un único

6. Ejemplo

1. Instalación

   npm install -g create-react-app

2. Creamos el proyecto

   create-react-app <nombre de la app>

3. Ingresamos a la carpeta del proyecto

   cd <nombre de la app>

4. Vemos la siguiente estructura

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   └── manifest.json
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    ├── logo.svg
    └── serviceWorker.js
```

7. Trabajo Práctico

Realizar la instalación de React y la creación de un proyecto que muestre como mínimo por pantalla

```
<h1>Bienvenido</h1>
<p>A la diplomatura en React</p>
```

Bibliografía utilizada y sugerida

Node JS (n.d.) Recuperado de: https://nodejs.org/es/

React JS (n.d.) Recuperado de: https://es.reactjs.org/

Wikipedia (n.d.) Recuperado de: https://es.wikipedia.org/wiki/React

Lo que vimos:

Introducción a la herramienta ReactJS.

Lo que viene:

Componentes, estados y eventos.