

# Creando una aplicación con React

## Práctica Integradora

### Objetivo

Para la práctica de estos temas de React consideramos que lo mejor es tener la mayor experiencia real posible, por lo cual vamos a estar trabajando con un proyecto localmente.

Aquí dejamos el [código resuelto](#) de todas las ejercitaciones de esta práctica. Esta solución está para ayudarnos en el caso de que lo necesitemos. Si resolvemos la ejercitación consultando la solución, recomendamos volver a realizarla, pero esta vez sin consultarla, con el fin de comprobar que logramos internalizar los conceptos vistos en las clases.

Recordemos que luego de descargar de descargar y descomprimir el archivo con el código, debemos ingresar a la carpeta `>mi-primer-proyecto`, ejecutar el comando `>npm install` y finalmente `>npm start`.



## Micro desafíos - Paso 1:

Vamos a trabajar sobre este [proyecto](#) que no puede renderizar bien la página. Recordemos que al descargarlo debemos ejecutar el comando `>npm install` para instalar las dependencias y luego `>npm start` para ejecutar el proyecto.

¿Cuál puede ser la razón de la falla? Tratemos de analizar en unos cinco minutos cómo se levanta el proyecto de React, cuáles son los primeros componentes que se ejecutan y observemos si eso se cumple en el proyecto, ¿Qué hace el componente 'Web.js'?

**TIP:** React tiene un componente llamado ReactDOM que es el encargado de renderizar los componentes. ¿Se está renderizando el componente correcto?

Si logramos identificar donde está la falla y compilar todo de manera exitosa, podremos ingresar a nuestro navegador de preferencia y ejecutar el comando: **localhost:3000**.

Deberías ver la siguiente pantalla:



## Micro desafíos - Paso 2:

Como ya sabemos, cuando trabajamos con React, debemos mantener un orden en toda nuestra estructura. Es por ello que ahora debemos crear, dentro de la carpeta `>src`, una nueva carpeta llamada `>components` y dentro de ella colocar o mover el componente llamado `Web`.

Este es el primer componente que está renderizando TODO el sitio web. Por lo cual sería una buena práctica modularizar TODO el sitio en varios componentes, pero antes de

comenzar con eso, tratemos de crear el primer componente nuestro desde cero.

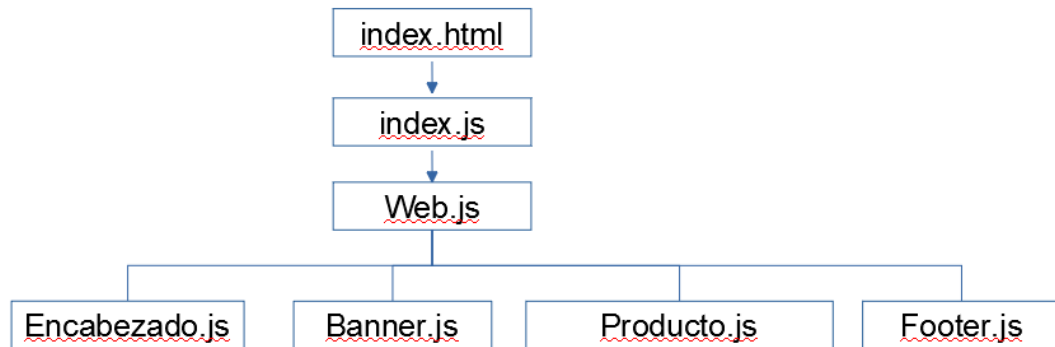
**TIP:** El código para crear un componente es el siguiente:

```
function NombreComponente() {  
  return (  
    <h1>Web</h1>  
  );  
}  
  
export default NombreComponente;
```

**TIP:** El código para poder utilizar un componente es el siguiente:

```
import React from 'react';  
import ComponenteHijo from './components/Componente'  
function ComponentePadre() {  
  return (  
    <ComponenteHijo/>  
  );  
}  
  
export default ComponentePadre;
```

Aquí dejamos una propuesta de todos y cada uno de los componentes que vamos a crear, partiendo del componente **Web** como podemos observar en la siguiente estructura:



Nuestro componente **Web** ahora va a renderizar el código que se muestra a continuación, por lo tanto, debemos ejecutar en el mismo los cambios necesarios. Nunca debemos olvidar que, cuando se trabaja con **JSX** en nuestros componentes, los atributos de las clases se deben llamar diferente, cerrar las etiquetas correctamente y las imágenes deben ser importadas, para luego imprimirlas en nuestras vistas:

```
<div class="container">
  <Encabezado />
</div>
```

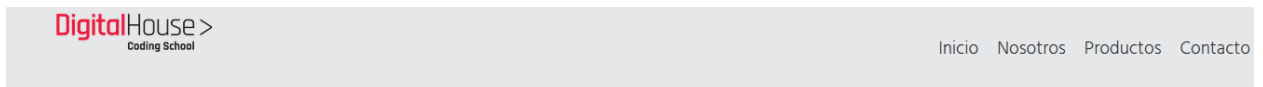
Como podemos ver, ahora es necesario crear el componente **Encabezado**. Este tendrá la responsabilidad de mostrar el **header de nuestra página**: [aquí podemos descargar el código HTML](#), el cual debemos transformar en base a las directivas de JSX. No olvidemos que debemos importar las imágenes utilizadas, tal como se muestra a continuación:

```
import logo from '../assets/img/logo-DH.png';
```

Para luego poderla imprimir en nuestra vista, de la siguiente manera:

```
<a href="#"><img src={logo} alt="Logo Digital House"/></a>
```

Si realizamos todos los ajustes necesarios para la creación de componentes, entendiendo que estamos trabajando con JSX, deberíamos ver la siguiente imagen:





## Micro desafíos - Paso 3:

Continuemos creando nuestros siguientes componentes. Ahora es el turno del componente **Banner**. Este tendrá la responsabilidad de mostrar en una sección de nuestra página **un Banner informativo**.

[Aquí podemos descargar el código HTML](#), el cual debemos transformar en base a las directivas de JSX. No olvidemos que cada vez que creamos un nuevo componente, debemos importarlo en nuestro componente **Web**, de la siguiente forma:

```
import Encabezado from './Encabezado';  
  
import Banner from './Banner';
```

Además, en este mismo componente es donde vamos a mandar a renderizar los componentes, de la siguiente manera:

```
return (  
  <div className="container">  
    <Encabezado />  
    <Banner />  
  </div>  
)
```

Si realizamos todos los ajustes necesarios para la creación de componentes, entendiendo que estamos trabajando con JSX, deberíamos ver la siguiente imagen:





## Micro desafíos - Paso 4:

Vamos muy bien, ya casi terminamos nuestro pequeño proyecto en React. A continuación debemos crear el siguiente componente **Producto**. Este tendrá la responsabilidad de mostrar en una sección de nuestra página **todos nuestros productos**.

[Aquí podemos descargar el código HTML](#), el cual debemos transformar en base a las directivas de JSX. No olvidemos que cada vez que creamos un nuevo componente, debemos importarlo en nuestro componente **Web**.

Al culminar este componente, deberíamos ver que se acaba de agregar la visualización de nuestros productos en la página, tal como se muestra a continuación:

### Nuestros productos



**Lorem, ipsum.**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni mollitia blanditiis natus dicta error, cum alias ipsam libero odio impedit, fuga at dolore voluptatem illum pariatur cupiditate fugit, ullam amet?

**Precio: \$450**

[Ver más](#)



**Lorem, ipsum.**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni mollitia blanditiis natus dicta error, cum alias ipsam libero odio impedit, fuga at dolore voluptatem illum pariatur cupiditate fugit, ullam amet?

**Precio: \$650**

[Ver más](#)



**Lorem, ipsum.**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni mollitia blanditiis natus dicta error, cum alias ipsam libero odio impedit, fuga at dolore voluptatem illum pariatur cupiditate fugit, ullam amet?

**Precio: \$945**

[Ver más](#)



**Lorem, ipsum.**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni mollitia blanditiis natus dicta error, cum alias ipsam libero odio impedit, fuga at dolore voluptatem illum pariatur cupiditate fugit, ullam amet?

**Precio: \$999**

[Ver más](#)

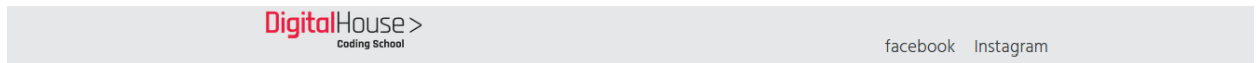


## Micro desafíos - Paso 5:

Para culminar debemos crear el componente **Footer**. Este tendrá la responsabilidad de mostrar en una sección de nuestra página **el pie de página**.

[Aquí podemos descargar el código HTML](#), el cual debemos transformar en base a las directivas de JSX. No olvidemos que cada vez que creamos un nuevo componente, debemos importarlo en nuestro componente **Web**.

Si todo el código del componente está correcto, deberíamos ver que se acaba de agregar la visualización de nuestro **footer** en la página, tal como se muestra a continuación:



## Conclusión

Felicitaciones, hemos logrado crear nuestro primer proyecto con React. [Aquí podemos descargar la solución de la práctica.](#)

**¡Hasta la próxima!**