



# Herramientas de Programación 2

Tema N°8:  
DISEÑO WEB CON BOOTSTRAP

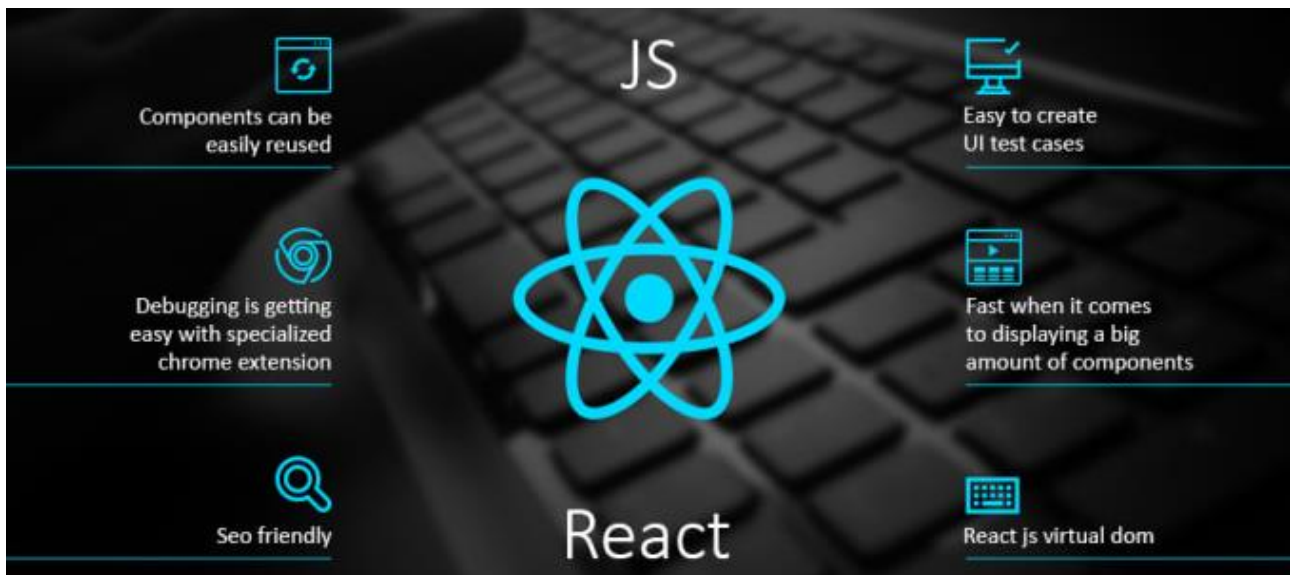
Indicador de logro N°8:  
Elabora diseño de páginas web utilizando Bootstrap y el estándar HTML5 (HTML, CSS y JS)

## TEMA Nº8: DISEÑO WEB CON BOOTSTRAP

### Subtema 8.1: REACT

React es una librería Javascript focalizada en el desarrollo de interfaces de usuario. Así se define la propia librería y evidentemente, esa es su principal área de trabajo. Sin embargo, lo cierto es que en React encontramos un excelente aliado para hacer todo tipo de aplicaciones web, SPA (Single Page Application) o incluso aplicaciones para móviles. Para ello, alrededor de React existe un completo ecosistema de módulos, herramientas y componentes capaces de ayudar al desarrollador a cubrir objetivos avanzados con relativamente poco esfuerzo.

Por tanto, React representa una base sólida sobre la cual se puede construir casi cualquier cosa con Javascript. Además facilita mucho el desarrollo, ya que nos ofrece muchas cosas ya listas, en las que no necesitamos invertir tiempo de trabajo.



Paso 1. Crear un Proyecto React.

- a. Verificamos la versión de *NodeJs* que tenemos instalado, para ello abrimos la consola de comandos de Windows (o en nuestro caso Windows Powershell) y ejecutamos el comando:  
*node --version*

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\QA> node --version
v12.13.1
PS C:\Users\QA>
```

- b. Verificamos la versión de *npm* que tenemos instalado, para ello ejecutamos el comando:  
*npm --version*

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\QA> node --version
v12.13.1
PS C:\Users\QA> npm --version
6.12.1
PS C:\Users\QA>
```

- c. Creamos una carpeta donde estará el código fuente de nuestras aplicaciones React.  
d. En la consola de comandos nos ubicamos dentro de la carpeta que acabamos de crear, por ejemplo:  
*cd D:\Personal\IDAT\IDAT\_HPII\_Presencial\Proyectos\S12\_React16*

```
PS C:\Users\QA> cd D:\Personal\IDAT\IDAT_HPII_Presencial\Proyectos\S12_React16
PS D:\Personal\IDAT\IDAT_HPII_Presencial\Proyectos\S12_React16>
```

- e. Para crear nuestra primera aplicación “mitiendavirtual\_frontend” ejecutamos el comando:  
*npx create-react-app mitiendavirtual\_frontend*

```
Windows PowerShell
PS D:\Personal\IDAT\IDAT_HPII_Presencial\Proyectos\S12_React16> npx create-react-app mitiendavirtual_frontend
```

- f. Culminado la creación del proyecto obtenemos:

```

Windows PowerShell

Starts the development server.

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

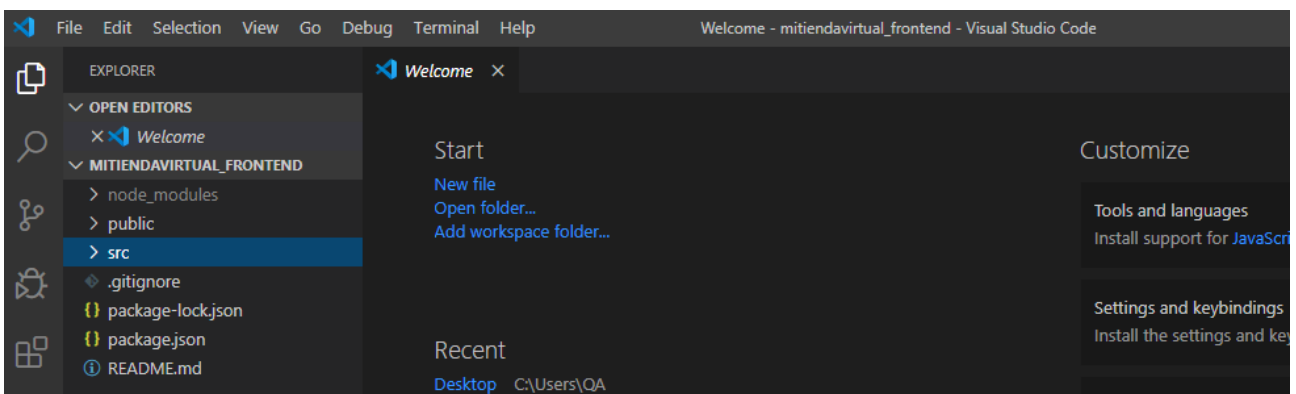
cd mitiendavirtual_frontend
npm start

Happy hacking!
PS D:\Personal\IDAT\IDAT_HPII_Presencial\Proyectos\S12_React16>
  
```

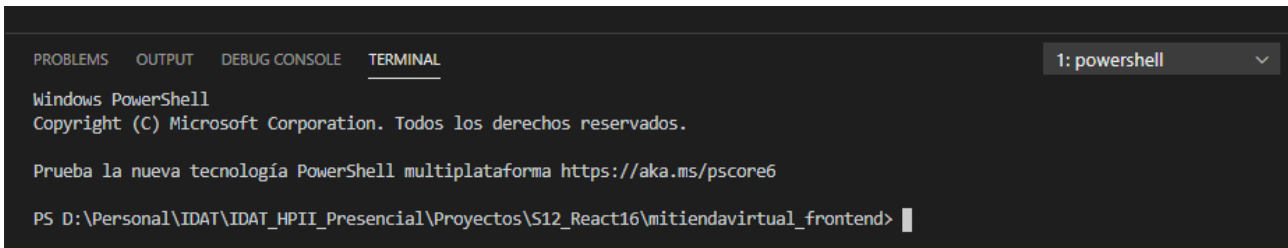
- g. Cerramos la ventana de la consola de comandos.
- h. Revisamos la carpeta y los archivos que se han creado.

| Proyectos > S12_React16 > mitiendavirtual_frontend > |                       |                     |        | Buscar en mitiendavirtual_fr... |  |
|--|-----------------------|---------------------|--------|---------------------------------|--|
| Nombre   | Fecha de modificación | Tipo                | Tamaño |                                 |  |
| .git   | 18/02/2020 16:39      | Carpeta de archivos |        |                                 |  |
| node_modules   | 18/02/2020 16:39      | Carpeta de archivos |        |                                 |  |
| public   | 18/02/2020 16:38      | Carpeta de archivos |        |                                 |  |
| src  | 18/02/2020 16:38      | Carpeta de archivos |        |                                 |  |
| .gitignore   | 26/10/1985 03:15      | Documento de te...  | 1 KB   |                                 |  |
| package.json   | 18/02/2020 16:39      | Archivo JSON        | 1 KB   |                                 |  |
| package-lock.json                                    | 18/02/2020 16:39      | Archivo JSON        | 565 KB |                                 |  |
| README.md  | 26/10/1985 03:15      | Archivo MD          | 3 KB   |                                 |  |

- i. Para editar el código de nuestra aplicación utilizaremos Visual Studio Code.
- j. En Visual Studio Code seleccionamos la opción: *File > Open Folder...*
- k. Seleccionamos la carpeta *mitiendavirtual\_frontend* y lo abrimos.



- l. Para ejecutar nuestra aplicación abrimos la consola de comandos que viene integrado con Visual Studio Code, para ello desde el menú seleccionamos la opción: *Terminal* > *New Terminal*.
- m. En la parte inferior de Visual Studio Code se muestra el panel *TERMINAL* desde donde podemos ejecutar diversos comandos.

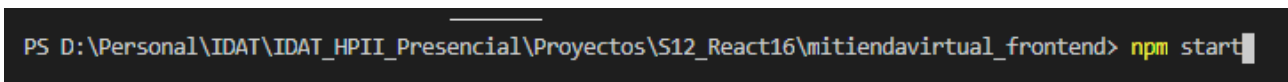


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

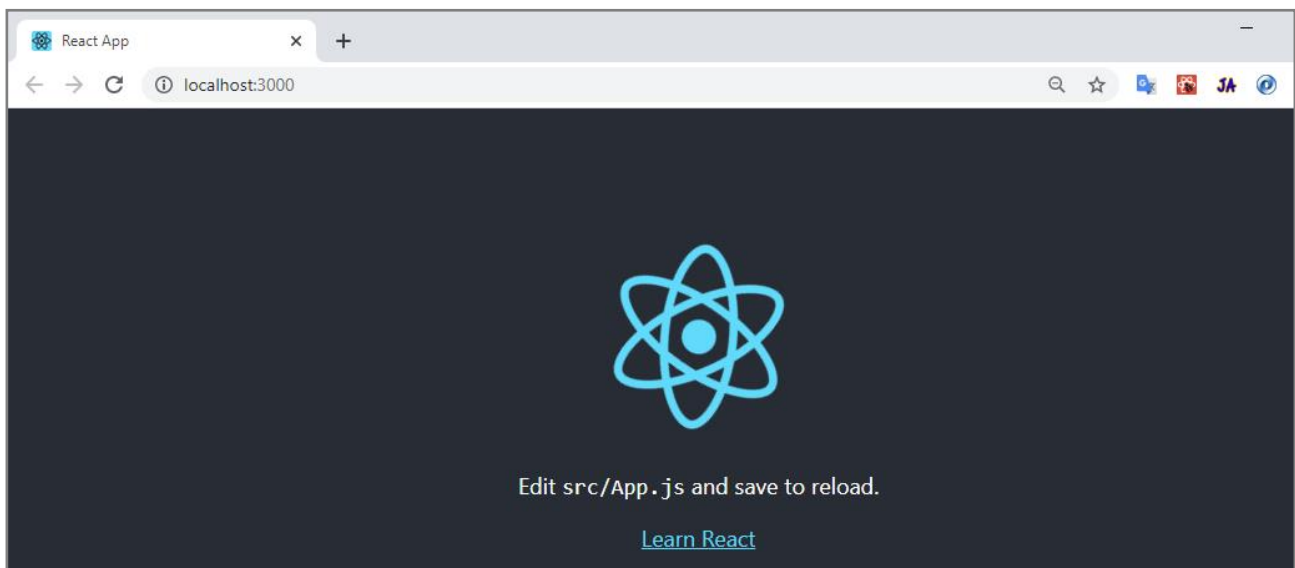
PS D:\Personal\IDAT\IDAT_HP II_Presencial\Proyectos\S12_React16\mitiendavirtual_frontend>
```

- n. Ejecutamos el comando:  
*npm start*



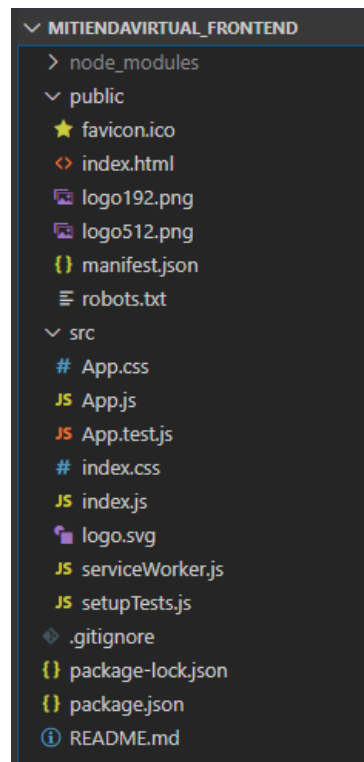
```
PS D:\Personal\IDAT\IDAT_HP II_Presencial\Proyectos\S12_React16\mitiendavirtual_frontend> npm start
```

- o. Se levanta el servicio que se encarga de gestionar nuestra aplicación React y nos muestra la página inicial.



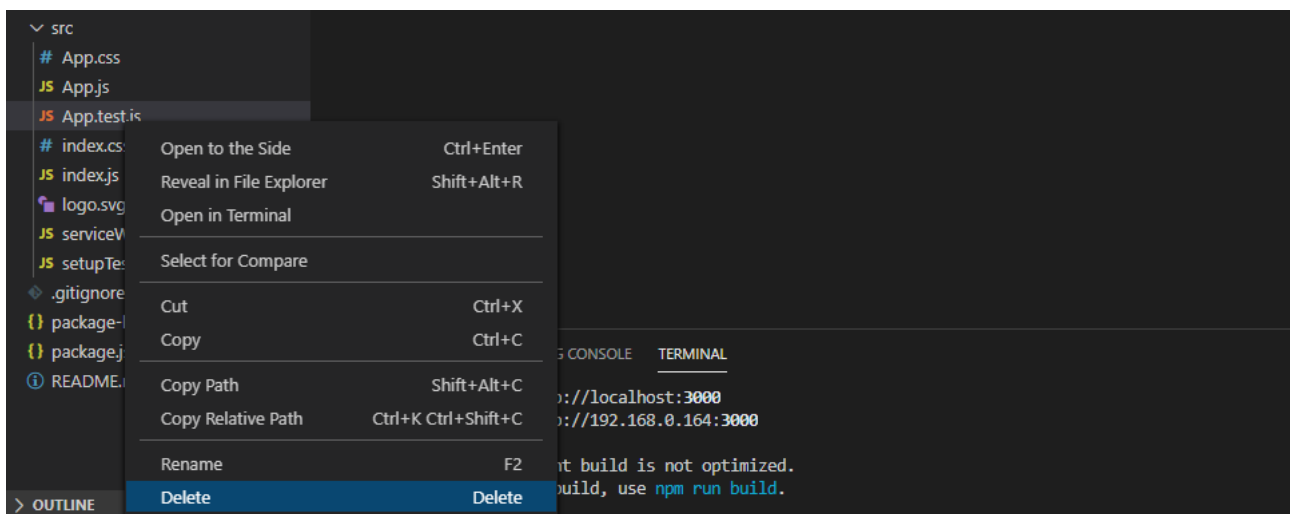
- p. Cada vez que actualicemos el código de nuestra aplicación el resultado se verá reflejado inmediatamente en esta ventana del explorador, por lo que no debemos cerrarla inmediatamente.

Paso 2. Estructura del proyecto.



Paso 3. Acondicionamos el proyecto.

- a. Dentro de la carpeta *src* eliminamos los siguientes archivos:
  - a. App.test.js
  - b. Logo.svg
  - c. serviceWorker.js
  - d. setupTest.js



- b. Abrimos el archivo *index.js* y actualizamos su código de acuerdo a:

```
JS index.js ×
src > JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5
6 ReactDOM.render(<App />, document.getElementById('root'));
```

c. Abrimos el archivo *index.css* y eliminamos todo su contenido:

```
JS index.js # index.css ×
src > # index.css
1 |
```

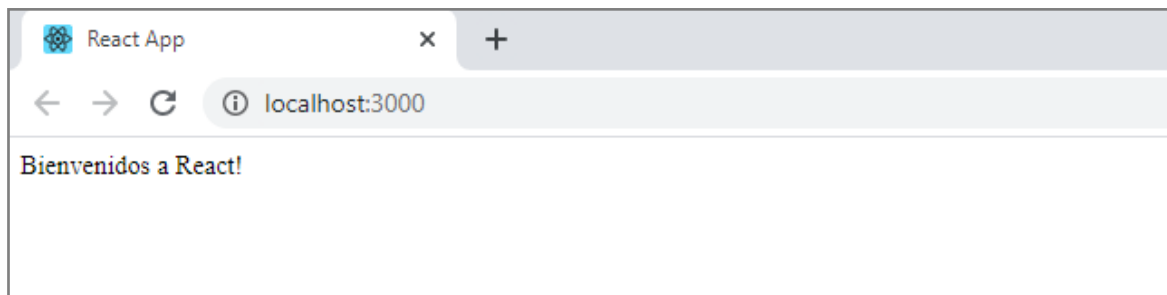
d. Abrimos el archivo *App.js* y actualizamos su código de acuerdo a:

```
JS index.js # index.css JS App.js ×
src > JS App.js > ...
1 import React from 'react';
2 import './App.css';
3
4 function App() {
5   return (
6     <div>Bienvenidos a React!</div>
7   );
8 }
9
10 export default App;
```

e. Abrimos el archivo *App.css* y eliminamos todo su contenido:

```
JS index.js # index.css JS App.js # App.css ×
src > # App.css
1 |
```

f. Verificamos que no se muestren errores y visualizamos el resultado de los cambios en el explorador web.



Paso 4. Agregar Bootstrap al proyecto React.

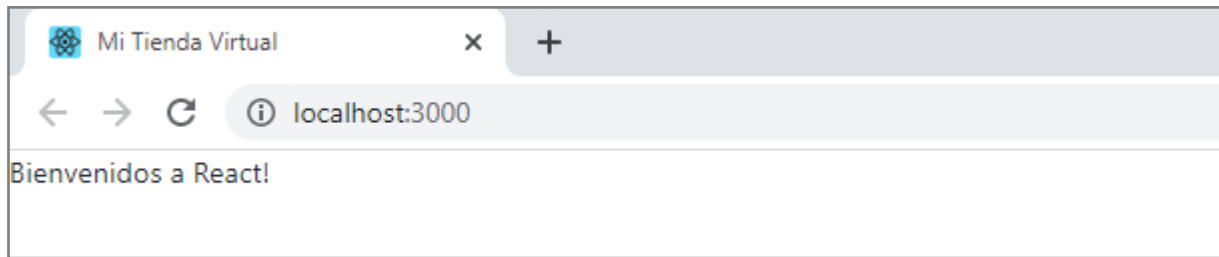
- a. Abrimos el archivo *index.html* (que está dentro de la carpeta *public*) y modificamos su contenido de acuerdo a:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap
.min.css" integrity="sha384-
Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <title>Mi Tienda Virtual</title>
  </head>
  <body>
    <noscript>Usted necesita habilitar JavaScript para ejecutar esta aplicación.</noscript>
    <div id="root"></div>

    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+pot4WnyKhv5vZF5SrPo0iEjwBvK7U7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></scrip
t>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity=
"sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></scrip
t>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrit
y="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6" crossorigin="anonymous"></scrip
t>
  </body>
</html>
```

- b. Verificamos que no se muestren errores y visualizamos el resultado de los cambios en el explorador web.





#### Paso 5. Componentes: Ejemplo 1.

- a. Abrimos el archivo *App.js* (que está dentro de la carpeta *src*) y modificamos su contenido de acuerdo a:

```
import React, {Component} from 'react';
import './App.css';

class App extends Component {

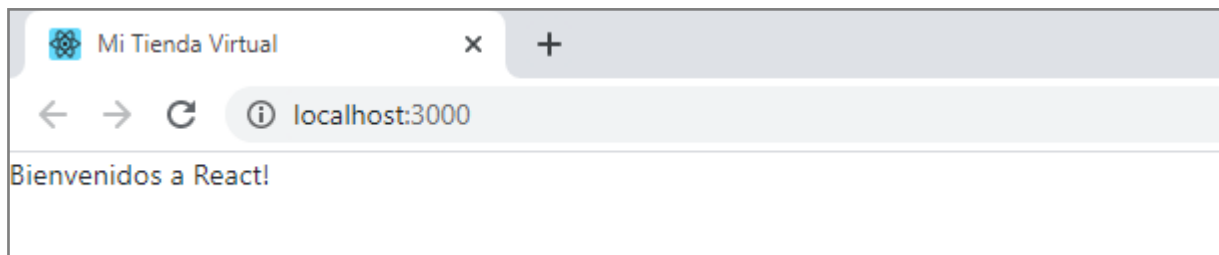
  render(){
    return <div >
      Bienvenidos a React!
    </div>
  }
}

export default App;
```

Con esto hemos convertido la función *App* en un componente (clase que hereda de *Component*).

En adelante cada sección de una página o una página en sí será un componente, cabe señalar que un componente puede contener otros componentes.

- b. Verificamos en el explorador web que no existen cambios.

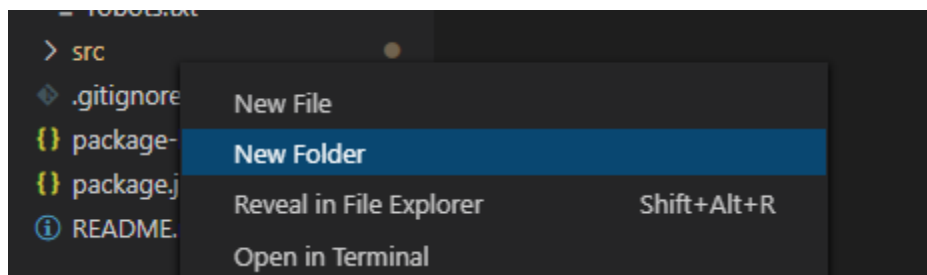


#### Paso 6. Componentes: Ejemplo 2.

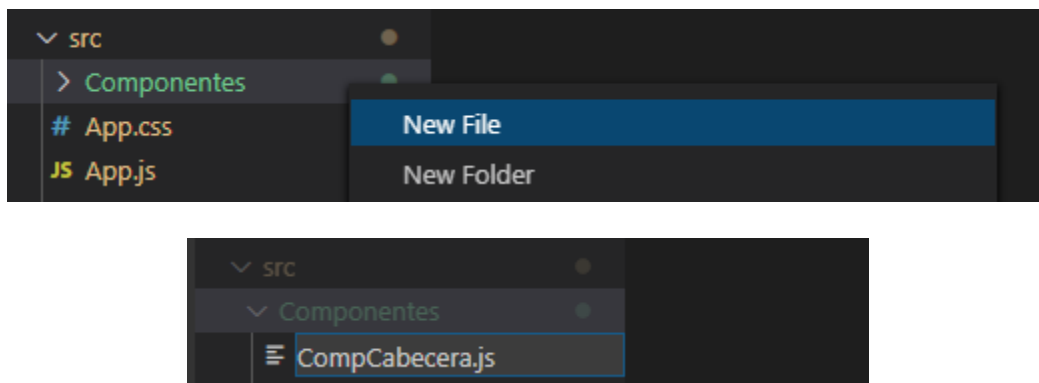
- a. Vamos a crear cinco componentes que van a conformar una pagina web, la página web en sí también será un componente.



- b. Primero creamos dentro de la carpeta *src* la carpeta *Componentes*.



- c. Vamos a crear nuestro primer componente (que será un archivo Javascript) tendrá el nombre *CompCabecera.js*



- d. Agregamos código jsx al componente *CompCabecera.js* de acuerdo a:

```
import React, {Component} from 'react';

class CompCabecera extends Component {

  render(){
    return <nav className="navbar navbar-expand-md navbar-dark bg-dark mb-4">
      <div className="container">
        <span className="float-left">
          <a className="navbar-brand" href="#">
            
          </a>
        </span>
        <span className="float-right">
          <div className="collapse navbar-collapse" id="navbarText">
            
            <a className="text-danger" href="#">&nbsp;3 producto(s)</a>
          </div>
        </span>
      </div>
    </nav>
  }
}

export default CompCabecera;
```

- e. Creamos el componente *CompPiePagina.js*. Agregamos código jsx al componente de acuerdo a:

```
import React, {Component} from 'react';

class CompPiePagina extends Component {

  render(){
    return <nav className="navbar fixed-bottom navbar-expand-sm navbar-dark bg-dark">
      <div className="container">
        <a className="navbar-brand" href="#">Mi Tienda Virtual</a>
        <p className="text-secondary">&copy;2020</p>
      </div>
    </nav>
  }
}

export default CompPiePagina;
```

- f. Creamos el componente *CompListaCategorias.js*. Agregamos código jsx al componente de acuerdo a:

```
import React, {Component} from 'react';

class CompListaCategorias extends Component {

  render(){
```

```

    return <div className="col-3">
      <p className="h5 text-secondary">Categorías</p>
      <div className="card">
        <ul className="list-group list-group-flush">
          <li className="list-group-item"><a href="#" className="text-
secondary">Destacados</a></li>
          <li className="list-group-item"><a href="#" className="text-
secondary">Ropa</a></li>
          <li className="list-group-item"><a href="#" className="text-
secondary">Zapatillas</a></li>
          <li className="list-group-item"><a href="#" className="text-
secondary">Accesorios</a></li>
        </ul>
      </div>
    </div>
  }
}

export default CompListaCategorias;

```

- g. Creamos el componente *CompProductoDetalle.js*. Agregamos código jsx al componente de acuerdo a:

```

import React, {Component} from 'react';

class CompProductoDetalle extends Component {

  render(){
    return <div className="col-9">
      <p className="h5 text-secondary">Producto</p>
      <div className="card mb-3">
        <div className="row no-gutters">
          <div className="col-md-4">
            
          </div>
          <div className="col-md-8">
            <div className="card-body">
              <h5 className="card-title text-primary">CASACA VARILITE</h5>
              <p className="card-
text">Cuando cae la temperatura, está casaca ofrece abrigo ligero y plegable contra el frio. Inco
rpora cámaras tejidas con aislamiento para un óptimo abrigo y una mayor resistencia al viento y e
l mal tiempo. Presenta un diseño suave al tacto con salidas para el cable de los audífonos en los
bolsillos.</p>
              <p className="h4 text-primary">S/ 379.00</p>
              <p className="text-right">
                <a className="btn btn-lg btn-
primary" href="#" role="button">Agregar al carrito</a>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  }
}

```

```
}  
}  
  
export default CompProductoDetalle;
```

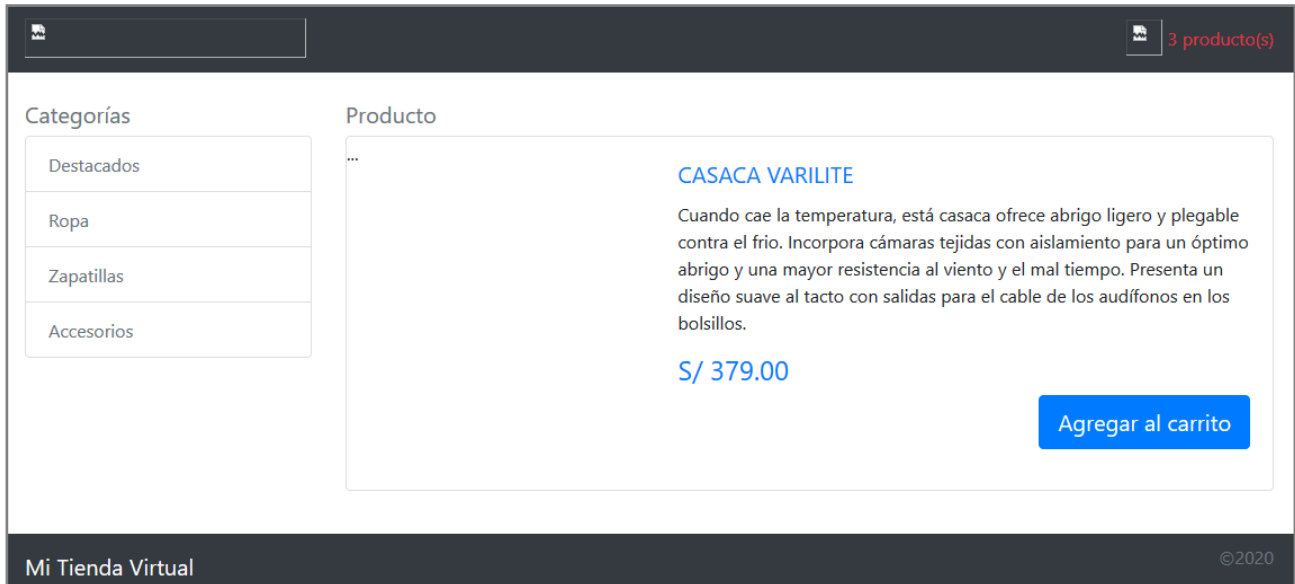
- h. Finalmente creamos la pagina que contendra los componentes creados anteriormente Creamos el componente *PageProducto.js*. Agregamos código jsx al componente de acuerdo a:

```
import React, {Component} from 'react';  
import CompCabecera from './CompCabecera';  
import CompListaCategorias from './CompListaCategorias';  
import CompProductoDetalle from './CompProductoDetalle';  
import CompPiePagina from './CompPiePagina';  
  
class PageProducto extends Component {  
  
  render(){  
    return <div >  
      <CompCabecera />  
      <main role="main" className="container">  
        <div className="row">  
          <CompListaCategorias />  
          <CompProductoDetalle />  
        </div>  
      </main>  
      <CompPiePagina />  
    </div>  
  }  
}  
  
export default PageProducto;
```

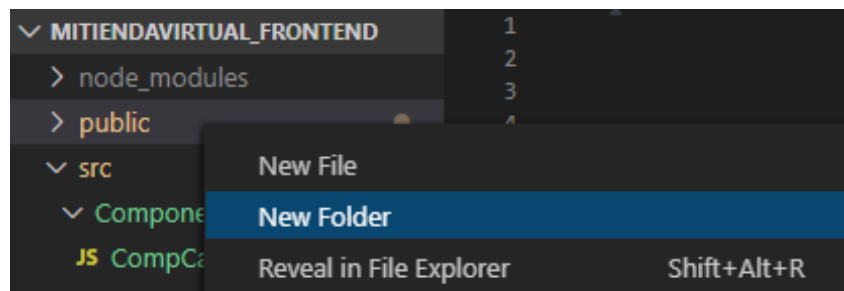
- i. Abrimos el archivo App.js y actualizamos su codigo de acuerdo a:

```
import React, {Component} from 'react';  
import './App.css';  
import PageProducto from './Componentes/PageProducto';  
  
class App extends Component {  
  
  render(){  
    return <PageProducto />  
  }  
}  
  
export default App;
```

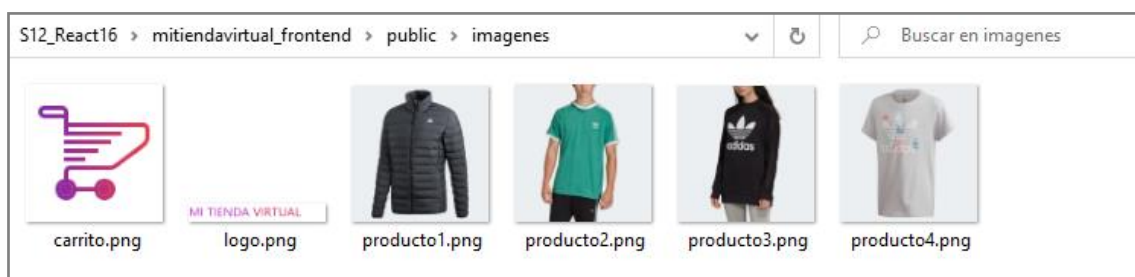
- j. Guardamos los cambios.
- k. Verificamos que no se muestran errores.
- l. Visualizamos los cambios en el explorador web.



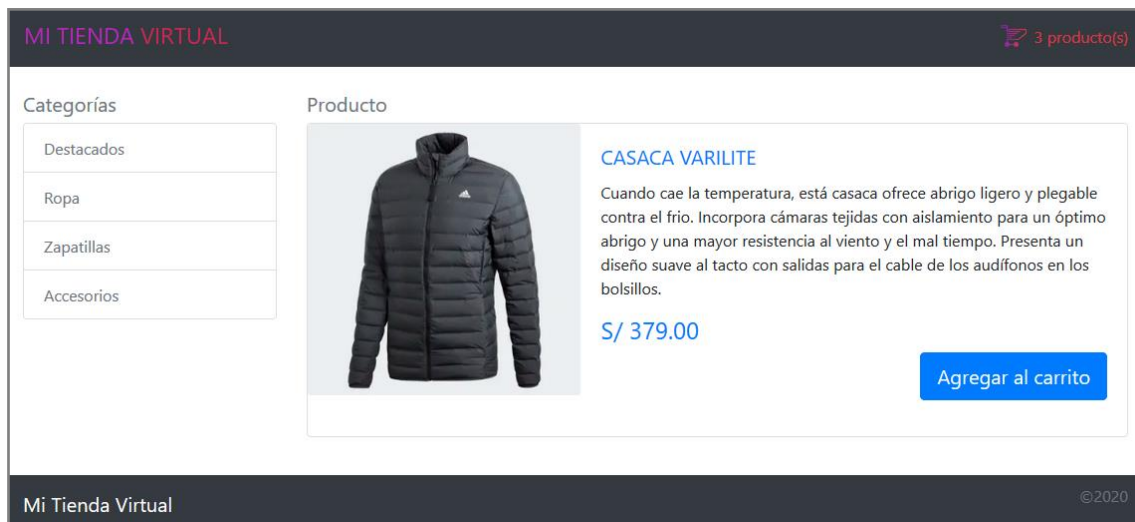
- m. Observamos que las imágenes no se están mostrando correctamente, para corregir esto primero creamos la carpeta *imagenes* dentro de la carpeta *public*.



- n. Dentro de la carpeta *imagenes* con el explorador de archivos agregamos los archivos de imágenes que trabajamos en el taller anterior.



- o. Hecho esto visualizamos los cambios en el explorador web (F5).

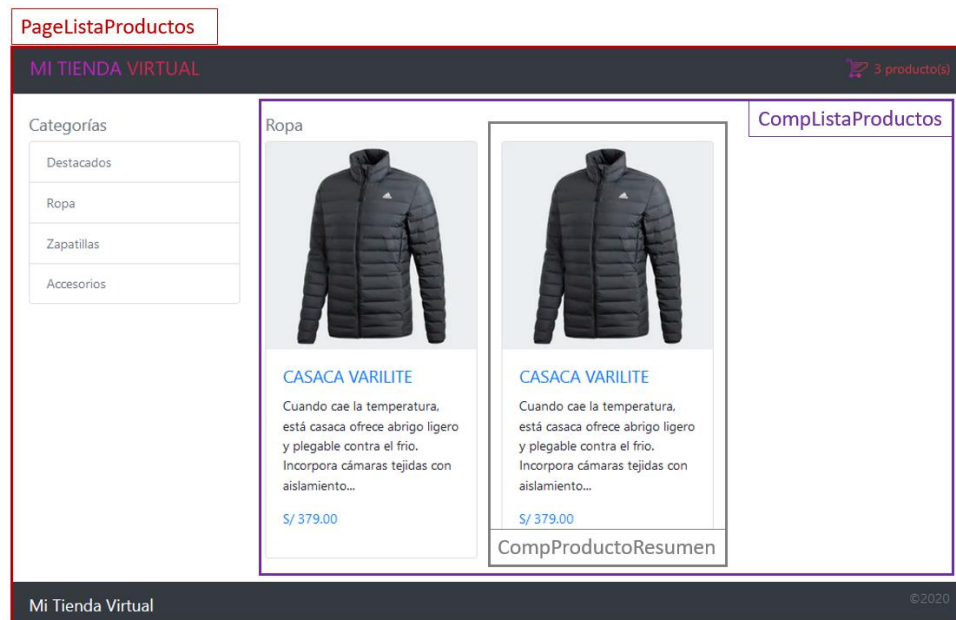


## ACTIVIDAD VIRTUAL

Ingresa a la plataforma virtual, luego desarrolla la siguiente actividad propuesta:

### a) CUESTIONARIO TÉCNICO

- i) Crear la página y los componentes necesarios para el siguiente diseño (debe reutilizar los componentes de la Cabecera, Pie de página y la Lista de categorías):



- ii) Crear el componente *CompProductoResumen.js*

- iii) Crear el componente *CompListaProductos.js*

- iv) Crear el componente *PageListaProductos.js*.

- v) Investigar la forma de enviar información de un componente a un componente interno.

b) CONCLUSIONES DE LA EXPERIENCIA

---

---

---

---

---

---

---

**ACTIVIDAD VIRTUAL:**

Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

c) CUESTIONARIO TÉCNICO

- vi) Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):

- Marca
- Categoría
- Producto

- vii) Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):

- Cliente
- Tarjeta

Estas dos entidades deben ser de solo lectura, esto es, no debe permitir crear, actualizar ni eliminar registros.

- viii) Para los nuevos mantenimientos, desarrollar las personalizaciones de forma análoga al CRUD de Usuarios.

- ix) Agregar en el menú superior links que permitan acceder a los nuevos mantenimientos. Además, alinear el menú a la derecha.



MI TIENDA VIRTUAL

[Pedidos](#)
[Productos](#)
[Categorías](#)
[Marcas](#)
[Clientes](#)
[Tarjetas](#)
[Usuarios](#)

# Welcome

Learn about [building Web apps with ASP.NET Core](#).

- x) Investigar la forma de agregar la funcionalidad de búsqueda en la lista de Usuarios, debe buscar por Nombres y Apellidos.

Usuarios

[Nuevo Usuario](#)

| Nombres  | Apellidos        | Dni      | Correo             | Activo                              |   |
|----------|------------------|----------|--------------------|-------------------------------------|---|
| Fernando | Vargas Camacho   | 10203040 | luis@hotmail.com   | <input checked="" type="checkbox"/> | <a href="#">Actualizar</a>   <a href="#">Eliminar</a> |
| Roberto  | Tubino Fernandez | 09584747 | guti0412@gmail.com | <input type="checkbox"/>            | <a href="#">Actualizar</a>   <a href="#">Eliminar</a> |

d) CONCLUSIONES DE LA EXPERIENCIA

---

---

---

---

---

---

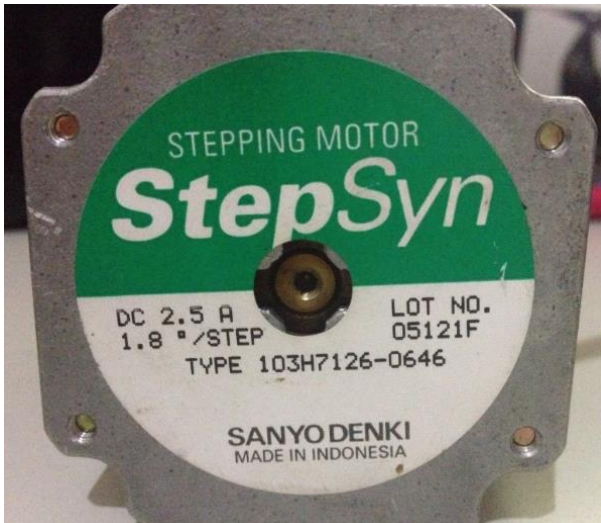
---

---

---

---

Un motor paso a paso siempre debemos considerar su corriente de operación y los grados, como indica el ejemplo de la siguiente placa característica de la imagen mostrada:

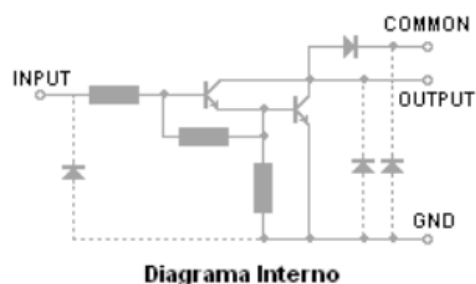
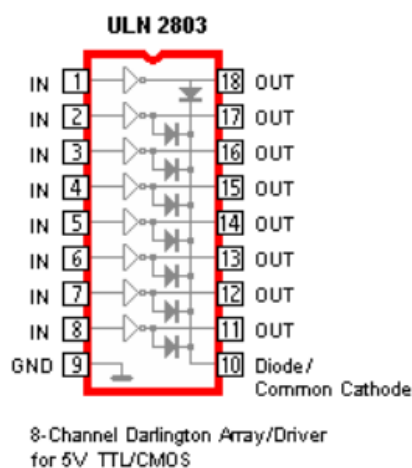


### Subtema 1.2:

## Transistores de potencia

Para operaciones con motores de bajo consumo tenemos circuitos integrados como el circuito ULN2803, el cual es un driver con transistores darlington, con soporte de hasta 500 mA.

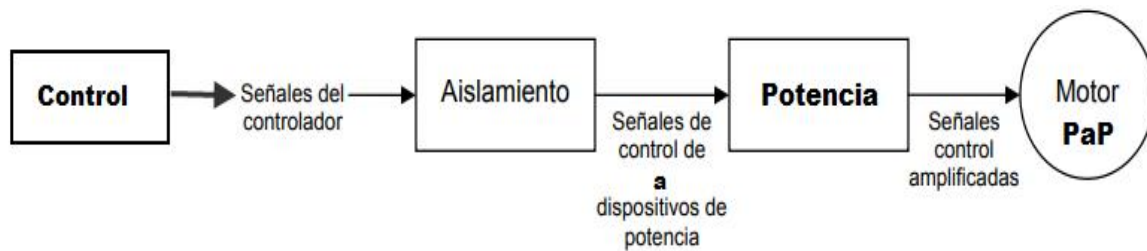
Por lo tanto si deseamos mayores corrientes debemos utilizar otros drivers o implementar uno con componentes discretos, es decir con transistores de potencia.



### Subtema 1.3:

## Control del Motor

## Las etapas para controlar un motor Paso a Paso



**Sistema de Control** + *Esquema básico del circuito excitador*

### Actividad:

Los estudiantes desarrollarán ejercicios de Teoría de exponentes y radicales, propuestos por el docente. según lo aprendido en clase. a) CUESTIONARIO TÉCNICO

- ¿Qué diferencia hay entre motores paso a paso unipolares y bipolares?
- ¿Por qué es importante usar driver para motores paso a paso?
- ¿Qué características posee un motor paso a paso?
- ¿Para qué me sirve el puente H?
- Investigue sobre cómo controlar los motores paso a paso con PWM y LOGO! 8.