

## DESARROLLO DE APLICACIONES WEB AVANZADO

### LABORATORIO N° 07

## Estilos con React y Bootstrap



<b>Alumno(s):</b>					<b>Nota</b>	
<b>Grupo:</b>			<b>Ciclo:V</b>			
<b>Criterio de Evaluación</b>	<b>Excelente (4pts)</b>	<b>Bueno (3pts)</b>	<b>Requiere mejora (2pts)</b>	<b>No accept. (0pts)</b>	<b>Puntaje Logrado</b>	
Entiende cómo funciona React-Bootstrap						
Utiliza Bootstrap para la maquetación						
Desarrolla aplicaciones web con express						
Realiza con éxito lo propuesto en la tarea.						
Es puntual y redacta el informe adecuadamente						

## Laboratorio 07: Estilos con React y Bootstrap

### Objetivos:

Al finalizar el laboratorio el estudiante será capaz de:

- Entender el funcionamiento de React-Bootstrap
- Desarrollar aplicaciones web con el framework Express
- Implementación correcta de estilos css

### Seguridad:

- Ubicar maletines y/o mochilas en el gabinete del aula de Laboratorio.
- No ingresar con líquidos, ni comida al aula de Laboratorio.
- Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

### Equipos y Materiales:

- Una computadora con:
  - Windows 7 o superior
  - VMware Workstation 10+ o VMware Player 7+
  - Conexión a la red del laboratorio
- Máquinas virtuales:
  - Windows 7 Pro 64bits Español - Plantilla
- Instalador de node.js

### Procedimiento:

#### Lab Setup

#### 1. Configuración inicial del proyecto

- 1.1. Copie el contenido del laboratorio 6 (la clase anterior) a excepción de la carpeta node\_modules en una nueva carpeta llamada lab07 y reinstale todas las dependencias:

```
>npm install
```

- 1.2. Instalaremos Bootstrap y su complemento React-Bootstrap

```
npm install react-bootstrap bootstrap
```

- 1.3. Modificaremos el archivo index.js para que agregue los estilos de Bootstrap

```
import React from 'react';
import ReactDOM from 'react-dom';
import 'bootstrap/dist/css/bootstrap.css';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

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

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

- 1.4. Modificamos App.js, para que incluya un nuevo componente llamado Layout que se encargará de envolver a nuestra aplicación (será la plantilla a utilizar)

```
import React, { Component } from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom';

import Home from './views/Home';
import Details from './views/Details';
import Login from './views/Login';
import Layout from './hoc/Layout/Layout';

class App extends Component {
  render() {
    return (
      <BrowserRouter>
        <Layout>
          <Switch>
            <Route path="/" exact component={Home} />
            <Route path="/details" component={Details} />
            <Route path="/login" component={Login} />
          </Switch>
        </Layout>
      </BrowserRouter>
    );
  }
}

export default App;
```

- 1.5. Crearemos la carpeta **hoc** (High Order Components) que recibirá este nombre por contener a los futuros componentes que sean invocados a nivel de raíz. Dentro de esta carpeta, crearemos la carpeta **Layout** y dentro de la misma, el archivo **Layout.js** con el siguiente contenido.

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

import Header from '../components/Header/Header';
import Footer from '../components/Footer/Footer';

const Layout = props => {
  return (
    <Fragment>
      <Header />
      <main id="main">{props.children}</main>
      <Footer />
    </Fragment>
  );
};

export default Layout;
```

**Fragment** es un componente de React que nos permite envolver componentes adyacentes. Si usted intenta retirar **Fragment**, le dará un error de sintaxis ya que el **return** de la función debe tener un componente envolviendo a los demás.

- 1.6. Cree el archivo **Header.js** dentro de la carpeta **Header**, que será creada en una carpeta llamada **components** dentro de **src**.

```
import React from 'react';
import {
  Nav,
  Navbar,
  NavDropdown,
  Form,
  FormControl,
  Button
} from 'react-bootstrap';

const Header = props => {
  return (
    <Navbar bg="dark" expand="lg" variant="dark">
      <div className="container">
        <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="mr-auto">
            <Nav.Link href="#home">Home</Nav.Link>
            <Nav.Link href="#link">Link</Nav.Link>
            <NavDropdown title="Dropdown" id="basic-nav-dropdown">
              <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>
              <NavDropdown.Item href="#action/3.2">
                Another action
              </NavDropdown.Item>
              <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>
              <NavDropdown.Divider />
              <NavDropdown.Item href="#action/3.4">
                Separated link
              </NavDropdown.Item>
            </NavDropdown>
          </Nav>
          <Form inline>
            <FormControl type="text" placeholder="Search" className="mr-sm-2" />
            <Button variant="outline-success">Search</Button>
          </Form>
        </Navbar.Collapse>
      </div>
    </Navbar>
  );
};

export default Header;
```

Puede copiar el contenido del **return** del primer ejemplo situado en el siguiente link (que es la documentación oficial de **React-Bootstrap**)

<https://react-bootstrap.github.io/components/navbar/#navbars-overview>

- 1.7. Así mismo, dentro de la carpeta **components** crearemos la carpeta **Footer**, que contendrá el archivo **Footer.js** con el siguiente contenido.

```
import React from 'react';

const Footer = props => {
  return (
    <footer id="footer" className="pb-4 pt-4">
      <div className="container">
        <div className="row text-center">
          <div className="col-12 col-lg">
            <a href="#">Preguntas frecuentes</a>
          </div>
          <div className="col-12 col-lg">
            <a href="#">Contáctanos</a>
          </div>
          <div className="col-12 col-lg">
            <a href="#">Términos y condiciones</a>
          </div>
          <div className="col-12 col-lg">
            <a href="#">Privacidad</a>
          </div>
          <div className="col-12 col-lg">
            <a href="#">Mi App</a>
          </div>
        </div>
      </div>
    </footer>
  );
};

export default Footer;
```

- 1.8. Adicionalmente, modificaremos el contenido de index.css para que contenga lo siguiente

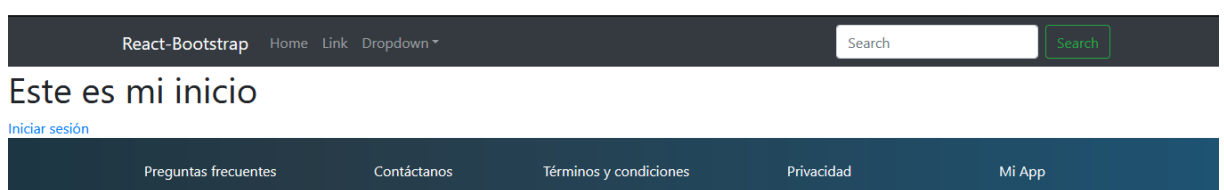
```
#header img {
  width: 170px;
}

#main .carousel-inner img {
  max-height: 70vh;
  object-fit: cover;
  filter: grayscale(70%);
}

#footer {
  background: linear-gradient(90deg, #1c3643, #273b47 25%, #1e5372);
}

#footer a {
  color: white;
}
```

- 1.9. Si todo ha ido bien, deberemos tener una vista como la siguiente.



React-Bootstrap
 Home Link Dropdown ▾

Usuario 
 Contraseña

Preguntas frecuentes
 Contáctanos
 Términos y condiciones
 Privacidad
 Mi App

## 2. Agregando estilos a las vistas existentes

- 2.1. Crearemos un componente de slides para darle una mejor presentación a nuestro sitio. Dentro de la carpeta **components** agregue la carpeta **Carousel** con el archivo **Carousel.js** que tendrá el siguiente contenido.

```
import React from 'react';
import { Carousel } from 'react-bootstrap';

import slide1 from '../../assets/img/slide1.jpg';
import slide2 from '../../assets/img/slide2.jpg';
import slide3 from '../../assets/img/slide3.jpg';
import slide4 from '../../assets/img/slide4.jpg';

const CarouselComponent = props => {
  return (
    <Carousel>
      <Carousel.Item>
        <img className="d-block w-100" src={slide1} alt="First slide" />
        <Carousel.Caption>
          <h3>First slide label</h3>
          <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
        </Carousel.Caption>
      </Carousel.Item>
      <Carousel.Item>
        <img className="d-block w-100" src={slide2} alt="Third slide" />

        <Carousel.Caption>
          <h3>Second slide label</h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        </Carousel.Caption>
      </Carousel.Item>
      <Carousel.Item>
        <img className="d-block w-100" src={slide3} alt="Third slide" />

        <Carousel.Caption>
          <h3>Third slide label</h3>
          <p>
            Praesent commodo cursus magna, vel scelerisque nisl consectetur.
          </p>
        </Carousel.Caption>
      </Carousel.Item>
    </Carousel>
  );
};
```

```

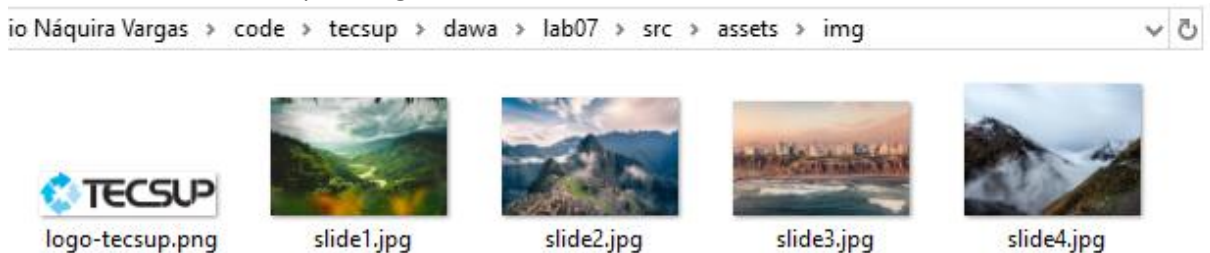
    <Carousel.Item>
      <img className="d-block w-100" src={slide4} alt="Fourth slide" />

      <Carousel.Caption>
        <h3>Fourth slide label</h3>
        <p>
          Praesent commodo cursus magna, vel scelerisque nisl consectetur.
        </p>
      </Carousel.Caption>
    </Carousel.Item>
  </Carousel>
);
};

export default CarouselComponent;

```

- 2.2. Descomprima el archivo **assets.zip** que fue entregada junto con este laboratorio y copie las fotos obtenidas en la carpeta **img** dentro de **assets**



- 2.3. Modifiquemos nuestro Home.js para que ahora incluya el **Carousel**

```

import React from 'react';
import { Link } from 'react-router-dom';

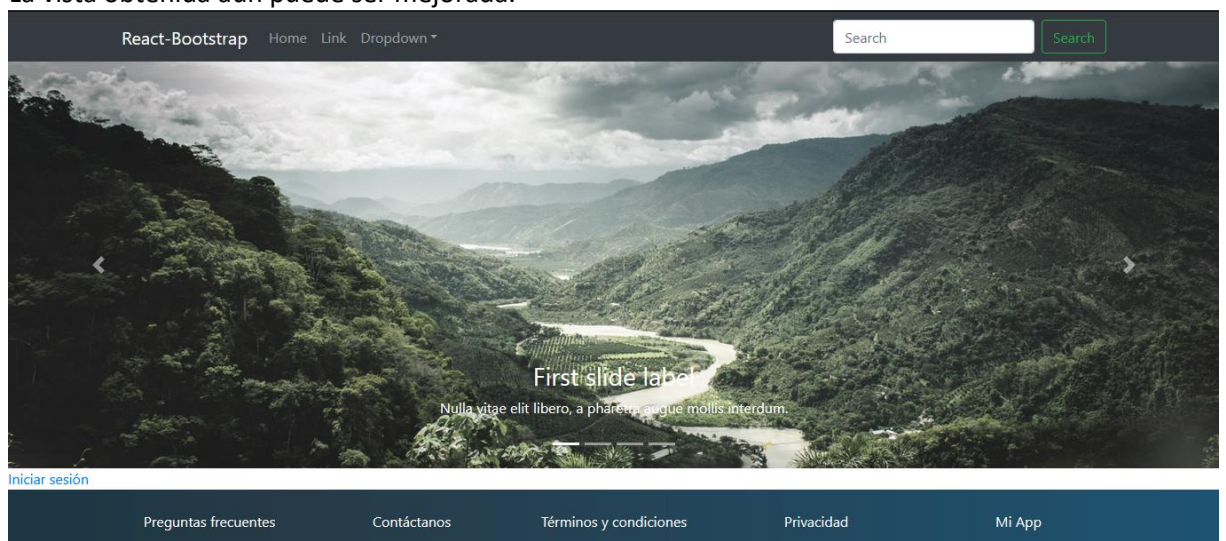
import Carousel from '../components/Carousel/Carousel';

const Home = () => {
  return (
    <div>
      <Carousel />
      <Link to="login">Iniciar sesión</Link>
    </div>
  );
};

export default Home;

```

- 2.4. La vista obtenida aún puede ser mejorada.



- 2.5. Modifique nuevamente el Home.js, ahora para agregar un pequeño mensaje de bienvenida y al mismo tiempo, mejorar los estilos del botón de inicia sesión.

```
import React, { Fragment } from 'react';
import { Link } from 'react-router-dom';

import Carousel from '../components/Carousel/Carousel';

const Home = () => {
  return (
    <Fragment>
      <Carousel />
      <div className="container">
        <div className="jumbotron">
          <h1>Bienvenido a mi aplicación!</h1>
          <p>
            <Link className="btn btn-primary btn-lg" to="/login">
              Inicie sesión
            </Link>
          </p>
        </div>
      </div>
    </Fragment>
  );
};

export default Home;
```

- 2.6. Ya que estamos modificando nuestra aplicación, modificaremos nuestro Header.js, ahora con enlaces reales y el logo de nuestra aplicación.

```
import React from 'react';
import { Link, NavLink } from 'react-router-dom';
import { Nav, Navbar } from 'react-bootstrap';

import imgLogo from '../../assets/img/logo-tecsup.png';

const Header = props => {
  return (
    <Navbar bg="dark" expand="lg" variant="dark" id="header">
      <div className="container">
        <Link className="navbar-brand" to="/">
          <img src={imgLogo} alt="Tecsups Logo" />
          Tecsups App
        </Link>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav" className="justify-content-end">
          <Nav>
            <NavLink to="/" exact className="nav-link">
              Inicio
            </NavLink>
            <NavLink to="/login" className="nav-link">
              Iniciar sesión
            </NavLink>
            <NavLink to="/logout" className="nav-link">
              Cerrar sesión
            </NavLink>
          </Nav>
        </Navbar.Collapse>
      </div>
    </Navbar>
  );
};

export default Header;
```



### 3. Agregando estilos al formulario de inicio

#### 3.1. Modificaremos la importación de Form y Button en Login.js

```
import React, { Component } from 'react';
import { Form, Button } from 'react-bootstrap';

import axios from '../utils/axios';
```

#### 3.2. También modificaremos el render de Login.js, para que el formulario ahora si luzca con los estilos de Bootstrap

```
render() {
  return (
    <div className="container">
      <div className="row justify-content-around">
        <div className="col-6 mt-2 mb-2">
          <div className="card">
            <div className="card-body">
              <Form onSubmit={this.submitHandler}>
                <Form.Group>
                  <Form.Label>Usuario</Form.Label>
                  <Form.Control
                    type="text"
                    placeholder="Usuario para sesión"
                    value={this.state.usuario}
                    onChange={this.usuarioHandler}
                  />
                </Form.Group>
                <Form.Group>
                  <Form.Label>Contraseña</Form.Label>
                  <Form.Control
                    type="password"
                    placeholder="123456"
                    value={this.state.password}
                    onChange={this.passwordHandler}
                  />
                </Form.Group>
                <Button variant="primary" type="submit">
                  Iniciar sesión
                </Button>
              </Form>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
}
```

#### 4. Vista por defecto y vista de bienvenida

- 4.1. Modificamos nuevamente App.js para tener dos nuevas vistas, una de bienvenida al usuario recién logueado, y otra de error para las rutas no definidas.

```
import Details from './views/Details';
import Login from './views/Login';
import Layout from './hoc/Layout/Layout';
import Welcome from './views/Welcome';
import NotFound from './views/NotFound';

class App extends Component {
  render() {
    return (
      <BrowserRouter>
        <Layout>
          <Switch>
            <Route path="/" exact component={Home} />
            <Route path="/details" component={Details} />
            <Route path="/login" component={Login} />
            <Route path="/welcome" component={Welcome} />
            <Route component={NotFound} />
          </Switch>
        </Layout>
      </BrowserRouter>
    );
  }
}
```

Note como la última ruta no tiene un path a donde apuntar. Esto significa que toda ruta que no caiga en ningún criterio anterior, utilizará por defecto esa vista de NotFound.

- 4.2. Crearemos la vista **NotFound.js** dentro de la carpeta **views**

```
import React from 'react';
import { Link } from 'react-router-dom';

const NotFound = () => {
  return (
    <div className="container">
      <div className="row justify-content-around">
        <div className="col-6 mt-2 mb-2">
          <div className="error-template">
            <h1>Oops!</h1>
            <h2>404 Not Found</h2>
            <div className="error-details">
              Lo sentimos, has solicitado una página que no existe
            </div>
            <div className="error-actions">
              <Link to="/" className="btn btn-primary btn-lg">
                <span className="glyphicon glyphicon-home" />
                Llévame al Inicio{' '}
              </Link>
              <Link to="/login" className="btn btn-default btn-lg">
                <span className="glyphicon glyphicon-envelope" /> Inicia sesión{' '}
              </Link>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

export default NotFound;
```

- 4.3. Al mismo tiempo, crearemos la vista **Welcome.js** dentro de la carpeta **views**. Note como hacemos uso de **localStorage** para obtener el nombre del usuario, pero dicho nombre aún no ha sido definido.

```
import React from 'react';

const Welcome = props => {
  const userName = localStorage.getItem('userName');
  return (
    <div name="container">
      <div className="jumbotron">
        <h1>Bienvenido {userName}!</h1>
      </div>
    </div>
  );
};

export default Welcome;
```

localStorage es el almacenamiento local del navegador, donde se pueden almacenar valores de tipo clave-valor.

- 4.4. Definamos entonces dicho nombre. Modificaremos la función **submitHandler** de **Login.js** para que, al encontrar un usuario válido, guarde en el **localStorage** las siguientes variables:

```
submitHandler = e => {
  e.preventDefault();
  axios({
    method: 'post',
    url: 'user/signin',
    data: {
      username: this.state.usuario,
      password: this.state.password
    }
  })
  .then(response => {
    console.log('mi respuesta', response);
    localStorage.setItem('userId', response.data.data._id);
    localStorage.setItem('userName', response.data.data.username);
    localStorage.setItem('userToken', response.data.token);
    this.props.history.push('/welcome');
  })
  .catch(error => {
    console.log('hubo un error', error);
  });
};

render() {
```

- 4.5. Finalmente, necesitaremos una ruta para eliminar nuestra sesión (cerrarla) por lo que llamaremos a una ruta logout. En este caso, no necesitamos una vista, simplemente eliminar nuestro localStorage y luego redirigirnos a algún lugar, por lo que modificaremos la importación de nuestro archivo **App.js** para que incluya **Redirect**, y agregaremos la ruta **logout**

```
import React, { Component } from 'react';
import { BrowserRouter, Route, Switch, Redirect } from 'react-router-dom';

import Home from './views/Home';
import Details from './views/Details';
import Login from './views/Login';
import Layout from './hoc/Layout/Layout';
import Welcome from './views/Welcome';
import NotFound from './views/NotFound';

class App extends Component {
  render() {
    return (
      <BrowserRouter>
        <Layout>
          <Switch>
            <Route path="/" exact component={Home} />
            <Route path="/details" component={Details} />
            <Route path="/login" component={Login} />
            <Route path="/welcome" component={Welcome} />
            <Route
              path="/logout"
              render={() => {
                localStorage.removeItem('userId');
                localStorage.removeItem('userName');
                localStorage.removeItem('userToken');
                return <Redirect to="/login" />;
              }}
            />
            <Route component={NotFound} />
          </Switch>
        </Layout>
      </BrowserRouter>
    );
  }
}
```

**6. Finalizar la sesión**

- 6.1. Apagar el equipo virtual
- 6.2. Apagar el equipo

**Conclusiones:**

Indicar las conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.