



Bootcamp de Desarrollo Web

Clase 11

**Herramientas y Frameworks
populares para el Desarrollo Web**



React

- React es una biblioteca de JavaScript de código abierto utilizada para construir aplicaciones web con interfaces de usuario ricas.
- React ofrece a los desarrolladores la flexibilidad de crear componentes personalizados reutilizables que fomenta procesos de desarrollo rápidos. Su capacidad para permitir una rápida representación de páginas web también lo hace más amigable para los motores de búsqueda. React es una excelente biblioteca para desarrollar aplicaciones empresariales ligeras y complejas.



React

React crea un DOM VIRTUAL en la memoria. En lugar de manipular el DOM del navegador directamente, React crea un DOM virtual en la memoria, donde realiza todas las manipulaciones necesarias, antes de realizar los cambios en el DOM del navegador.

React solo cambia lo que se necesita cambiar, React descubre qué cambios se han realizado y cambia solo lo que se necesita cambiar.



React – Estructura de código

```
my-project
├── node_modules
└── public
    ├── favicon.ico
    ├── index.html
    ├── manifest.json
    ├── robots.txt
    └── sitemap.xml
└── src
    ├── actions
    │   ├── auth.js
    │   ├── post.js
    │   └── types.js
    ├── components
    │   ├── auth
    │   ├── common
    │   ├── feed
    │   ├── layout
    │   ├── navbar
    │   ├── post
    │   └── profile
    └── middleware
        └── api.js
```

```
profile
└── middleware
    └── api.js
└── reducers
    ├── auth.js
    ├── index.js
    ├── post.js
    └── types.js
└── styles
    ├── base.css
    ├── layout.css
    ├── modules.css
    ├── reset.css
    ├── theme.css
    └── variables.css
└── utils
    ├── api.js
    ├── auth.js
    └── helpers.js
```

```
views
├── auth
├── feed
├── layout
├── post
└── profile
    ├── Apps.js
    ├── index.css
    └── index.js
    ├── setupTests.js
    └── store.js
    └── serviceWorker.js
└── index.js
    ├── .env
    ├── .eslintrc.json
    ├── .gitignore
    ├── package.json
    └── README.md
```



React - Estructura de código

- **src:** Directorio que contiene el código fuente de la aplicación de react.
- **actions and reduces:** Ejemplo de directorios para almacenar contenido específico de Redux
- **components:** Contiene componentes reutilizables de la aplicación de react.
- **styles:** La carpeta de estilos contiene estilos CSS.
- **utils:** La carpeta utils contiene funciones de utilidad.
- **views:** La carpeta de vistas contiene componentes de nivel superior responsables de representar páginas o secciones específicas de la aplicación.
- **app.js:** El archivo contiene el componente raíz de la aplicación.
- **index.js:** El archivo es responsable de representar el componente raíz y montarlo en el DOM.
- **package.json:** El archivo package.json enumera las dependencias del proyecto.
- **README.md:** El archivo proporciona documentación para el proyecto.



React – componentes

- Los componentes de React son los scripts básicos de cualquier aplicación React. Encapsulan la estructura, la apariencia y el comportamiento de una parte de la interfaz de usuario. Los componentes son reutilizables, componibles y pueden administrar su propio estado o recibir datos de los componentes principales, lo que los convierte en herramientas poderosas para crear aplicaciones web dinámicas e interactivas.
- Los componentes en React se pueden definir de dos maneras: como componentes de clase o componentes de función. Los componentes de clase son clases ES6 que se extienden desde `React.Component` y deben implementar un método `render()`, que devuelve el JSX que se va a renderizar. Los componentes de función, por otro lado, son más simples y se pueden escribir como funciones que devuelven JSX directamente.



React – componentes

Clase del componente

```
class Car extends React.Component {  
  render() {  
    return <h2>Hi, I am a Car!</h2>;  
  }  
}
```

Función del componente

```
function Car() {  
  return <h2>Hi, I am a Car!</h2>;  
}
```

Render del componente

```
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Car />);
```

Props

```
function Car(props) {  
  return <h2>I am a {props.color} Car!</h2>;  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Car color="red"/>);
```



React – JSX

- ¿Qué es JSX? JSX significa JavaScript XML. JSX nos permite escribir HTML en React. JSX facilita la escritura y la incorporación de HTML en React. Codificación de JSX JSX nos permite escribir elementos HTML en JavaScript y colocarlos en el DOM sin ningún método `createElement()` o `appendChild()`. JSX convierte las etiquetas HTML en elementos de React. No es necesario que uses JSX, pero JSX facilita la escritura de aplicaciones de React.



React – Node Modules

Cada aplicación Node.js tiene módulos. Estos módulos forman parte de los componentes básicos de la aplicación. Ayudan a los desarrolladores a trabajar más rápido y a escribir código más estructurado.

En términos simples, un módulo es un fragmento de código JavaScript reutilizable. Puede ser un archivo .js o un directorio que contiene archivos .js. Puede exportar el contenido de estos archivos y usarlos en otros archivos. Los módulos ayudan a los desarrolladores a cumplir con el principio DRY (Don't Repeat Yourself) en programación. También ayudan a dividir la lógica compleja en fragmentos pequeños, simples y manejables.

Tipos de módulos de nodo Hay tres tipos principales de módulos de nodo con los que trabajará como desarrollador de Node.js. Incluyen lo siguiente: Módulos integrados Módulos locales Módulos de terceros.



React – Axios

Axios es una biblioteca JavaScript de código abierto muy popular que se utiliza para realizar solicitudes HTTP desde navegadores web o entornos Node.js. Simplifica el proceso de envío de solicitudes HTTP asíncronas a puntos finales REST, el manejo de respuestas y la realización de diversas tareas relacionadas con la red. Axios, que se basa en XMLHttpRequest nativo de JavaScript y la API de búsqueda, ofrece una API más fácil de usar con funciones como interceptores, transformación automática de datos JSON, manejo de errores y compatibilidad con navegadores más antiguos.



React – Casos de uso

React se utiliza en una amplia gama de aplicaciones, incluidas aplicaciones web, plataformas de transmisión de video, sitios de medios, herramientas de software como servicio, aplicaciones móviles (React Native), aplicaciones de escritorio (React + Electron.js), aplicaciones web progresivas (React + Ionic) y sitios JAMstack (Gatsby.js). Algunas aplicaciones populares construidas con React incluyen PayPal, Facebook y Netflix.



React – Casos de uso

Ventajas de React

Páginas web eficientes

Mejor experiencia de usuario

Reutilización de componentes

Alto rendimiento

Gran comunidad

Desventajas de React

Contras de React:

Actualizaciones continuas

Aumento de la complejidad del código

Se centra solo en la capa de vista



Bootstrap

Bootstrap es un popular framework de frontend para desarrollar sitios web responsivos y con diseño first-mobile y fue creado por Twitter, proporciona componentes y utilidades prediseñadas para acelerar el desarrollo web.

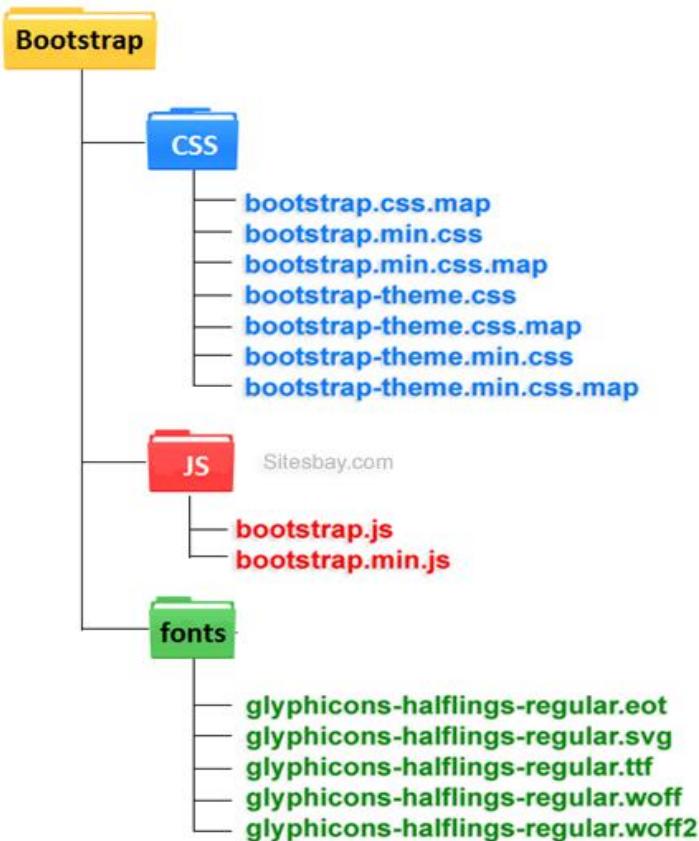
En esencia, Bootstrap es una herramienta robusta y versátil para crear sitios web modernos de forma rápida y eficiente.

Componentes Principales:

- **Sistema de Cuadrícula:** Sistema de diseño flexible y responsive.
- **Botones:** Varios estilos y tamaños.
- **Formularios:** Controles de formulario fáciles de estilizar y personalizar.
- **Navegación:** Barras de navegación, pestañas y pildoras preconstruidas.
- **Utilidades:** Clases auxiliares para margen, padding, alineación de texto, etc.



Bootstrap





Bootstrap

Ventajas:

- **Diseño Responsivo:** Asegura que tu sitio se vea bien en todos los dispositivos.
- **Componentes Preestilizados:** Ofrece botones, formularios, modales, barras de navegación, etc., que puedes usar fácilmente.
- **Consistencia:** Garantiza una apariencia uniforme en diferentes navegadores y dispositivos.
- **Personalización:** Fácilmente personalizable a través de variables y mixins de Sass.
- **Gran Comunidad y Documentación:** Muchos recursos, plugins y soporte.



Bootstrap

Casos de Uso:

- Prototipado rápido de interfaces web.
- Desarrollo de front-end listo para producción para aplicaciones web.
- Creación de diseños responsivos sin escribir CSS personalizado desde cero.
- Proyectos a gran escala que necesitan consistencia y eficiencia.