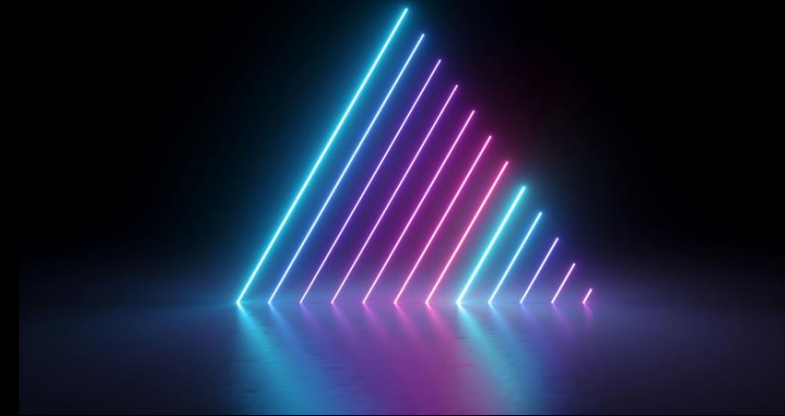
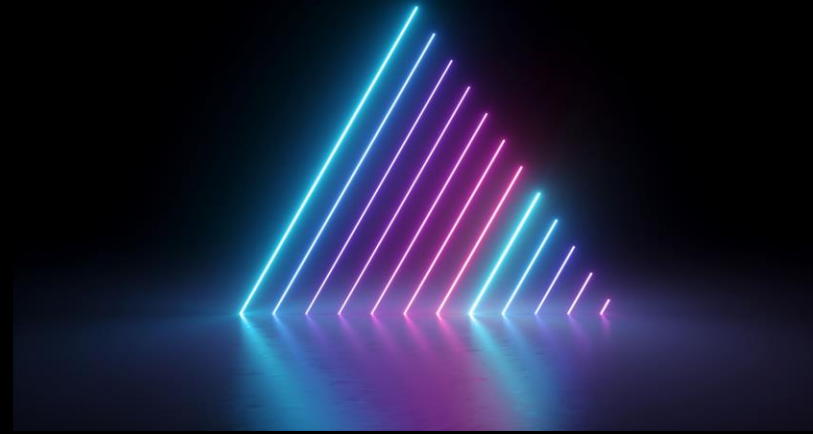
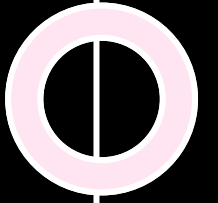


TUTORIAL REACT

Aurora Andreu Mateo,
Juan Fernando Gaviria Álvarez y
Adrián Federico Bartel Prieto





Contenido

- Introducción a React
- Configuración
- JSX: Sintaxis de React
- Componentes
- Estado y ciclo de vida en componentes de clase





Introducción a React

- Biblioteca de JavaScript para construir interfaces de usuario
- Son reutilizables e independientes
- Actualización automática
- Permite el desarrollo de aplicaciones web y móviles
- Eficiente y flexible



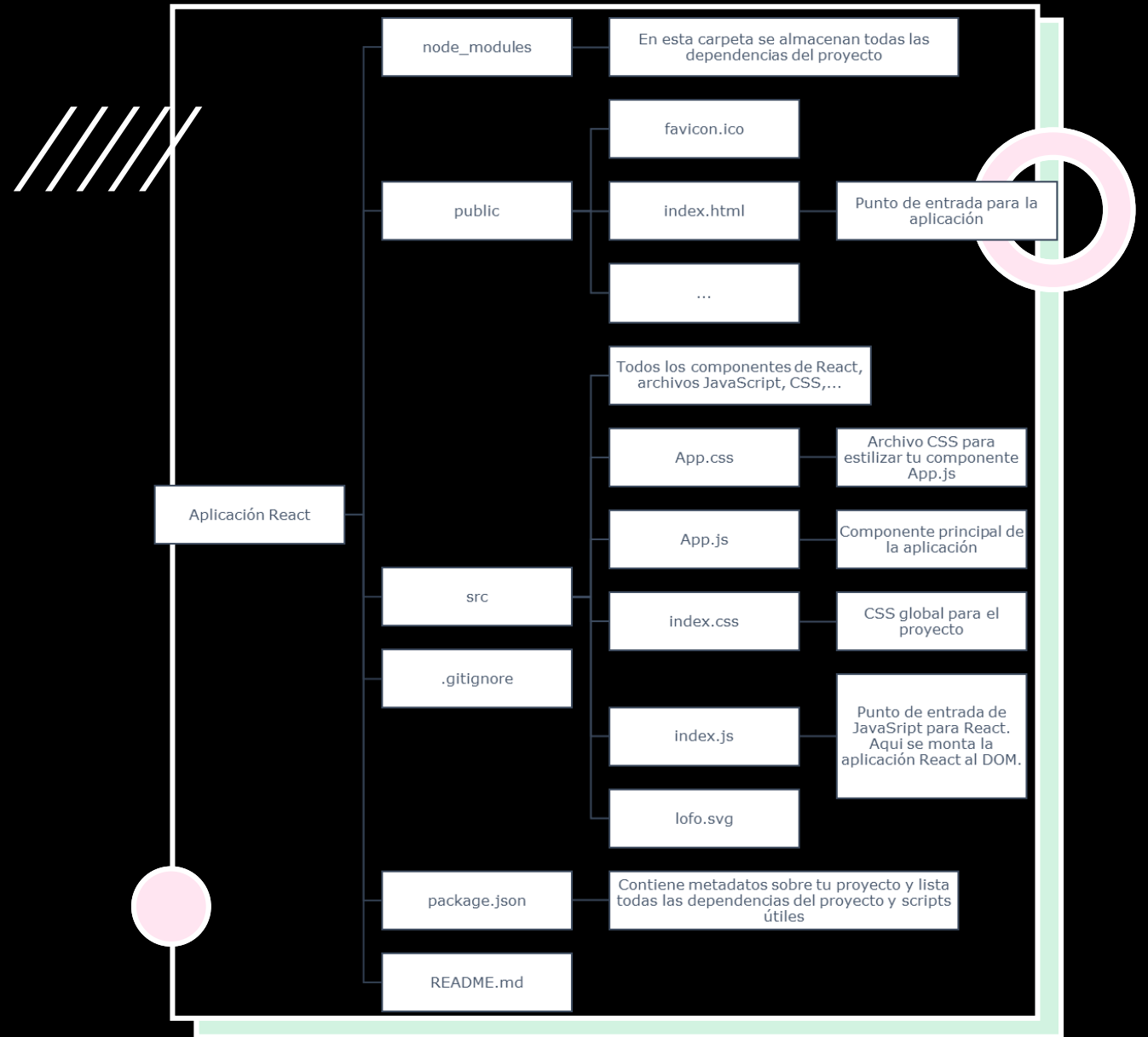


Configuración

- <https://nodejs.org/en>
- Para ver las versiones y comprobar que está instalado
 - Node -v
 - Npm -v
- Crear un nuevo proyecto con React
 - Npx create-react-app [nombre-de-la-aplicación]
- Ejecutar la aplicación:
 - Npm start



ESTRUCTURA



JSX: Sintaxis de React

```
const nombre = "Aurora"  
const saludo = <h1> Hola, {nombre} </h1>
```



JSX: Sintaxis de React

Componentes de clase

```
class Saludar extends React.Component {  
  render(){  
    return <h1>Hola, {this.props.nombre}</h1>;  
  }  
}
```

Componente funcional

```
function Saludar( props ) {  
  return <h1>Hola, {props.nombre}</h1>;  
}
```



JSX: Sintaxis de React

- Código HTML para ambos componentes

```
const App = () => {  
  return <Saludar nombre='Aurora' />;  
}
```



Componentes en React

```
function App() {  
  return (  
    <div>  
      <Saludar nombre='Aurora' />  
      <Saludar nombre='Juan' />  
    </div>  
  );  
}
```



Componentes en React

```
function Mensaje(props) {  
  return <p>{props.texto}</p>;  
}  
  
function App(){  
  return (  
    <div>  
      <Mensaje texto='Envia un mensaje importante' />  
    </div>  
  );  
}
```



Formularios

Datos

```
const Formulario = () => {  
  // Estado para almacenar los datos del formulario  
  const [formData, setFormData] = useState({  
    nombre: "",  
    email: "",  
  });  
};
```

handleChange y handleSubmit

```
// Manejador de cambios en los inputs  
const handleChange = (e) => {  
  const { name, value } = e.target;  
  setFormData({ ...formData, [name]: value });  
};  
  
// Manejador de envío del formulario  
const handleSubmit = (e) => {  
  e.preventDefault();  
  // Aquí se puede realizar acciones con los datos del formulario  
  console.log("Datos del formulario:", formData);  
};
```



● <form>

```
return (  
  <form onSubmit={handleSubmit}>  
    <label>  
      Nombre:  
      <input  
        type="text"  
        name="nombre"  
        value={formData.nombre}  
        onChange={handleChange}  
      />  
    </label>  
    <br />  
    <label>  
      Email:  
      <input  
        type="email"  
        name="email"  
        value={formData.email}  
        onChange={handleChange}  
      />  
    </label>  
    <br />  
    <button type="submit">Enviar</button>  
  </form>  
);
```



● Estilos y CSS

- Style

```
const MiComponente = () => {  
  return (  
    <div style={{ color: 'blue', fontSize: '16px', padding: '10px' }}>  
      Contenido del componente  
    </div>  
  );  
};
```

- CSS Modules

```
import React from 'react';  
import styles from './MiComponente.module.css';  
  
const MiComponente = () => {  
  return <div className={styles.miComponente}>Contenido del componente</div>;  
};
```



Rutas y navegación con React Router

- Creación de rutas

```
import React from 'react';
import { Route, Switch } from 'react-router-dom';
import Inicio from './Inicio';
import AcercaDe from './AcercaDe';
import Contacto from './Contacto';

const App = () => {
  return (
    <div>
      <Switch>
        <Route path="/acerca-de" component={AcercaDe} />
        <Route path="/contacto" component={Contacto} />
        <Route path="/" component={Inicio} />
      </Switch>
    </div>
  );
};
```

- Navegación entre páginas

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

const Header = () => {
  return (
    <nav>
      <ul>
        <li> <Link to="/"> Inicio </Link> </li>
        <li> <Link to="/acerca-de"> Acerca de </Link> </li>
        <li> <Link to="/contacto"> Contacto </Link> </li>
      </ul>
    </nav>
  );
};
```



● Llamadas a API's

- Fetch

```
useEffect(() => {  
  const fetchData = async () => {  
    try {  
      const response = await fetch('https://api.example.com/data');  
      const result = await response.json();  
      setData(result);  
    } catch (error) {  
      console.error('Error al obtener datos:', error);  
    }  
  }  
});
```

- Axios

```
useEffect(() => {  
  const fetchData = async () => {  
    try {  
      const response = await axios.get("https://api.example.com/data");  
      setData(response.data);  
    } catch (error) {  
      console.error("Error al obtener datos:", error);  
    }  
  }  
});
```

