



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

PROGRAMACIÓN INTEGRATIVA DE COMPONENTES WEB

NRC - ASIGNATURA: 19178

PROFESOR: Ing. Jose A. Sancho Arias

PERÍODO ACADÉMICO: 2024_50

TAREA N° 2.2

TÍTULO:

**Manual de manejo de Listas en frameworks de
web components + React**

ESTUDIANTE:

Betty Lizeth Rodriguez Salas

ENLACE VIDEO:

<https://youtu.be/BbwRCEuBHog>

FECHA DE ENTREGA:

2024 / 08 / 03

Interfaz visual de formulario con listas

I. Introducción

Las actividades incluyen definir una aplicación base, implementar funcionalidades y documentar el proceso. Los entregables son una aplicación en React, la funcionalidad implementada, un manual detallado y un video demostrativo.

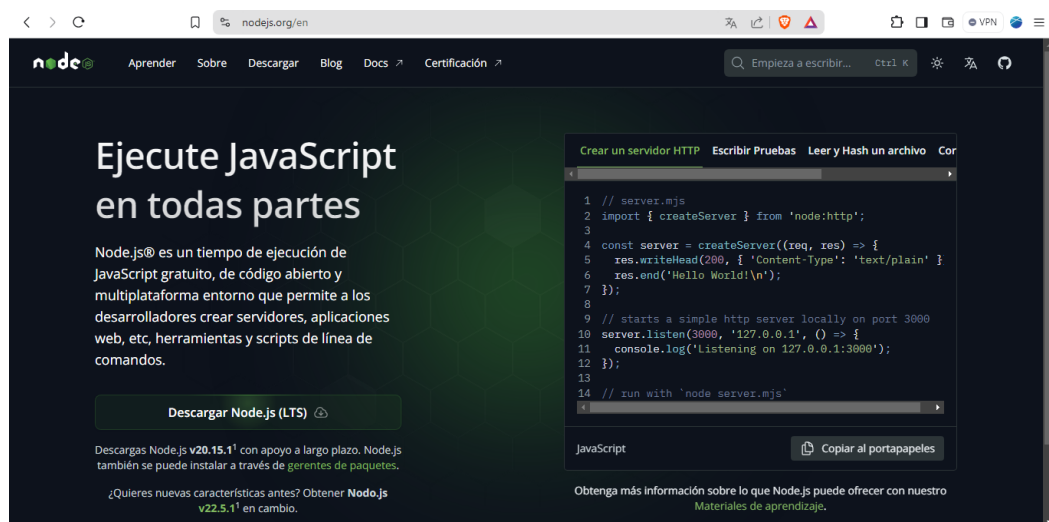
II. Objetivos

OBJETIVO: identificar el desarrollo de aplicaciones, implementando framework y librerías basadas en web componentes

OBJETO DE ESTUDIO: Listas con web components

III. Instalación de plataforma de desarrollo

Primero accedemos a <https://nodejs.org/en> descargamos e instalamos los componentes necesarios para la plataforma base node js.



Iniciamos el cmd y ejecutamos el comando `node -version` para ver la versión de node instalado en el dispositivo.

```
Windows PowerShell
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\lizet>node -v
v20.15.1
```

```
Windows PowerShell
C:\Users\lizet>npm -v
10.7.0
```

IV. Instalación framework React

Dentro de la línea de comando vamos a instalar React a nivel global con los comandos

```
C:\WINDOWS\system32\cmd.
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\lizet>npm install -g create-react-app
```

La versión instalada de React

```
C:\proyectos\react>cd listas
C:\proyectos\react\listas>npm list react
listas@0.1.0 C:\proyectos\react\listas
+-- @testing-library/react@13.4.0
|   |-- react@18.3.1 deduped
|   |-- react-dom@18.3.1
|   |-- react@18.3.1 deduped
|   |-- react-scripts@5.0.1
|   |-- react@18.3.1 deduped
|   |-- react@18.3.1
+-- react@18.3.1
```

V. Creación e inicialización de la tarea

Ingresamos en la carpeta de React agregando el comando `cd`

```
C:\proyectos>cd react
```

Creamos el nuevo proyecto llamado “listas”

```
C:\proyectos\react>npx create-react-app listas
Creating a new React app in C:\proyectos\react\listas.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1480 packages in 14m
261 packages are looking for funding
run 'npm fund' for details
```

Para inicializar agregamos el comando `npm start`

```
Windows PowerShell
Happy hacking!
C:\proyectos\react>cd listas
C:\proyectos\react\listas>npm start
> listas@0.1.0 start
> react-scripts start
```

Finalmente se activa el local host

```
Windows PowerShell
You can now view listas in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Visualizamos en el navegador el framework React.



VI. Código de lógica de negocio.

En términos de lógica de negocio, el componente Parking gestiona la visualización de los autos en el estacionamiento. La lógica relacionada con cómo se obtienen y actualizan los datos de los autos (si se conectan a una API o se manejan internamente) estaría en otros lugares del código o en componentes adicionales, Parking se encarga de presentar los datos y estructura visual para el usuario final.

Datos de Autos:

Se define un array cars con información sobre varios autos, incluyendo su id, brand (marca), model (modelo) y owner (propietario).

Renderización de la Interfaz:

Encabezado y Imagen: Dentro del primer div con la clase section, se muestra un encabezado (h1) y una imagen (img) que actúa como fondo para la sección.

Lista de Autos: Dentro del segundo div, se mapea el array cars a una lista (ul). Cada elemento de la lista representa un auto, y para cada auto se renderiza

un componente Car con las propiedades correspondientes (brand, model, y owner).

Estilos y Recursos:

Estilos: Se importan estilos desde el archivo parking.css.

Imagen: Se importa una imagen (parImd) para usarla como fondo o elemento decorativo.

Componente Car:

El componente Car se encarga de mostrar la información de cada auto. Aunque no está incluido en este fragmento, se asume que Car recibe las props (brand, model, owner) y las muestra de alguna manera

```
1 import React from "react";
2 import Car from "./Car";
3 import "./parking.css";
4 import parImd from "./assets/parking.jpg";
5
6 function Parking() {
7   const cars = [
8     { id: 1, brand: "Ford", model: "Focus", owner: "Alicia" },
9     { id: 2, brand: "BMW", model: "X3", owner: "Jhon" },
10    { id: 3, brand: "Audi", model: "A4", owner: "Charlie" },
11  ];
12  return (
13    <>
14      <div className="container">
15        <div className="section">
16          <h1 className="title">
17            Estacionamiento Universidad de las Fuerzas Armadas- ESPE
18          </h1>
19          <img src={parImd} alt="imagen_fondo" />
20        </div>
21        <div className="section">
22          <ul>
23            {cars.map((car) => (
24              <Car
25                key={car.id}
26                brand={car.brand}
27                model={car.model}
28                owner={car.owner}
29              />
30            ))}
31          </ul>
32        </div>
33      </div>
34    </>
35  );
36 }
37
38 export default Parking;
39
```

El componente Car muestra información de un auto (marca, modelo y propietario) en un elemento de lista () y utiliza un archivo de estilos para su formato.

El componente `Car` muestra información de un auto (marca, modelo y propietario) en un elemento de lista (``) y utiliza un archivo de estilos para su formato.

```
1 import React from 'react';
2 import './Car.css';
3
4 function Car(props) {
5   return (
6     <li>
7       <strong> Marca: </strong> {props.brand},
8       <strong> Modelo:</strong> {props.model},
9       <strong> Propietario:</strong> {props.owner}
10    </li>
11  );
12 }
13
14 export default Car;
```

Este código configura y renderiza una aplicación React, importando los módulos necesarios, el archivo de estilos, y el componente Parking. Crea un contenedor para la aplicación en el elemento del DOM con el id root y renderiza el componente Parking dentro de ese contenedor.

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import Parking from './Parking';
5
6 const root = ReactDOM.createRoot(document.getElementById("root"));
7 root.render(<Parking />);
8
```

Estilos:

Estiliza los elementos de lista con borde azul, fondo claro, bordes redondeados, y una sombra ligera, y cambia el fondo al pasar el ratón.

```
1 li {
2   padding: 15px;
3   margin: 10px 0;
4   border: 2px solid #007BFF; /* Resalta con un borde azul */
5   border-radius: 8px; /* Bordes redondeados */
6   background-color: #f9f9f9; /* Fondo claro */
7   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Sombra ligera para destacar */
8   font-size: 16px;
9   font-family: Arial, sans-serif;
10 }
11
12 strong {
13   color: #007BFF; /* Color azul para los labels */
14 }
15
16 li:hover {
17   background-color: #e9ecef; /* Cambio de fondo en hover */
18 }
```

Define fuentes globales, elimina márgenes y padding del body, y centra el contenido dentro del contenedor #root.

```
1 /* Estilos globales */
2 body {
3   margin: 0;
4   font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",
5     "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",
6     sans-serif;
7   -webkit-font-smoothing: antialiased;
8   -moz-osx-font-smoothing: grayscale;
9   padding: 0;
10  overflow: hidden;
11 }
12
13 code {
14   font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",
15     monospace;
16 }
17
18 /* Asegura que el contenedor principal tenga un padding adecuado */
19 #root {
20   display: flex;
21   justify-content: center; /* Centra el contenido horizontalmente */
22   align-items: center; /* Centra el contenido verticalmente */
23   min-height: 100vh; /* Ocupa toda la altura de la ventana */
24   padding: 0;
25 }
```

Estilos para:

h1: Títulos centrados, fuente moderna y color oscuro.

ul: Lista sin bullets, centrada y con ancho limitado.

.container: Contenido centrado, fondo claro, y altura completa.

img: Ancho máximo al 100%.

.section: Diseño de secciones en columna con padding.

```

1  h1 {
2    color: #333; /* Color de texto oscuro para contraste */
3    text-align: center; /* Centra el título */
4    margin-bottom: 20px;
5    font-size: 2em;
6    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif; /* Fuente moderna */
7  }
8
9  ul {
10   list-style-type: none; /* Elimina los bullets por defecto */
11   padding: 0;
12   margin: 0 auto;
13   max-width: 600px; /* Limita el ancho para mejor presentación */
14 }
15
16 .container {
17   display: flex;
18   align-items: center; /* Centra horizontalmente */
19   justify-content: space-between; /* Centra verticalmente */
20   min-height: 100vh; /* Ocupa toda la altura de la ventana */
21   width: 100%;
22   padding: 20px;
23   background-color: #f0f2f5; /* Fondo claro */
24 }
25 img {
26   max-width: 100%;
27 }
28 .section {
29   display: flex;
30   flex-direction: column;
31   padding: 1rem;
32   align-items: stretch;
33 }
34

```

VII. Navegación

Evidenciamos que la pagina tiene de lado derecho el listado de autos que se encuentran en el estacionamiento y funciona perfectamente.

