



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

Departamento de Sistemas y Computación

EXAMEN

Carrera: Ingeniería En Informática Período: **Agosto- Dic 2020** **Materia:** Desarrollo de Aplicaciones para

Dispositivos Móviles **Grupo:** AEB-1011IF8A Salón: **Unidad (es) a evaluar:** Unidad 1 y Unidad 2 **Tipo de**

examen: Practico Fecha: 19/Nov/2020 **Catedrático:** Daniela A. Sánchez Vizcarra Firma del maestro: Calificación:

Alumno: Fierro López Luis Eduardo No. Control: 16212352

Elabore utilizando **React JS**, la siguiente aplicación para realizar la selección de productos que desea



comprar.

En componente **App**, se mandaran llamar los componentes: **Encabezado, Productos y**

Listado • Encabezado: solo mostrara el logo

• **Productos:** mostrara el **state** productosLista y el total

• **Listado:** mostrara el **state** carrito

El **state** inicial será el siguiente:

```
this.state = {
  carrito:[],
  total:0,
  productosLista:[
    {codigo:1,descripcion:"Huawei Matebook D 15", precio:15899},
    {codigo:2,descripcion:"Samsung Galaxy S10", precio:13999},
    {codigo:3,descripcion:"Samsung Galaxy A01", precio:1850},
    {codigo:4,descripcion:"Xiaomi Redmi Note 9s", precio:5949},
    {codigo:5,descripcion:"Mochila Xiaomi", precio:699},
    {codigo:6,descripcion:"Teclado Primus Gaming Ballista", precio:1999},
  ],
};
```

Para dar formato al precio puede utilizar:

`${(producto.precio).toFixed(2).replace(/\d(?=(\d{3})+\.)/g, '$&,')}` No olvide enviar su reporte, enlace de su repositorio y el deploy de la aplicación

Herramientas

Para la realización de esta práctica se utilizaron diversas herramientas para manejar el lenguaje JavaScript y alojar la página en un servidor web como lo es GH Pages.

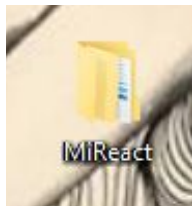
- Node js
- Visual Studio Code
- Lenguaje JavaScript
- GitHub
- GH-pages
- Bootstrap
- SweetAlert2

Desarrollo

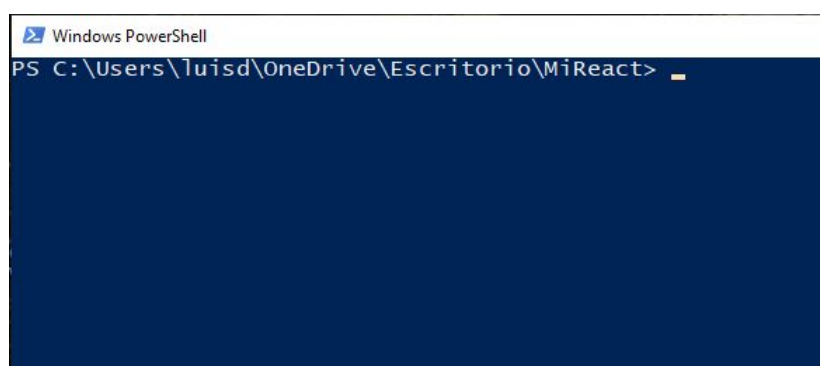
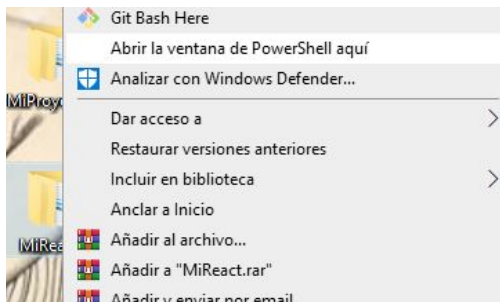
En este apartado se presentarán los pasos que fueron necesarios para realizar esta actividad

Como requisito haber instalado Node js y posteriormente crear una carpeta en la cual se descargar los archivos con los que trabajaremos.

Seleccionar carpeta para crear el proyecto:



En esa carpeta alojada en el escritorio vamos a abrir una consola de powershell presionando shift + click derecho



Crear proyecto

Para crear el proyecto se deberá usar el siguiente comando en la ventana de powershell que abrimos con anterioridad:

```
npx create-react-app nombre
```

Donde “nombre” es el nombre que le daremos a la app, procurar que esté en minúsculas y sea una sola palabra por si las moscas.

Cambiar directorio

Para acceder al directorio de la carpeta utilizaremos el comando cd y el nombre de la carpeta del proyecto de la siguiente manera:

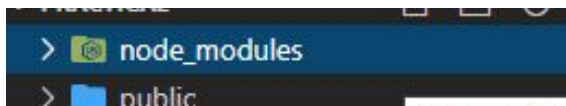
```
cd nombre
```

Ahora estaremos dentro de la carpeta.

Instalar carpeta node_modules

Para instalar dicha carpeta y que el proyecto pueda funcionar utilizaremos el siguiente comando en el mismo powershell al que cambiamos el directorio:

```
npm install
```



Instalar componente Bootstrap

Ahora deberemos instalar los componentes con los que vamos a trabajar para no tener problemas con las importaciones siendo uno de estos Bootstrap:

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

Ejemplo:



Instalar componente Sweetart 2

Este es para las alertas al ejecutar una acción:

```
npm install sweetalert2
```

Ejemplo:



Abrir proyecto

Estando en la misma ventana de powershell donde realizamos las instalaciones podemos abrir el proyecto en un editor de texto o iniciaron en localhost, pero si cerramos la ventana de powershell podemos abrir otra de la siguiente manera.

Seleccionamos la carpeta con shift y presionamos click derecho > abrir la ventana de powershell aquí



En la ventana de powershell agregamos el comando de **“code .”** para abrir la carpeta en vs code y el comando **“npm start”** para abrir el servidor local, a veces puede tardar una eternidad así que toca esperar. También se puede realizar desde la consola que ofrece VS code.

```
PS C:\Users\luisd\OneDrive\Escritorio\MiReact\practica2> code .
PS C:\Users\luisd\OneDrive\Escritorio\MiReact\practica2> npm start

> example17@0.1.0 start C:\Users\luisd\OneDrive\Escritorio\MiReact\practica2
> react-scripts start

D:\ws: Project is running at http://192.168.56.1/
D:\ws: webpack output is served from
D:\ws: Content not from webpack is served from C:\Users\luisd\OneDrive\Escritorio
D:\ws: 404s will fallback to /
Starting the development server...
Compiled successfully!

You can now view example17 in the browser.

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

Añadir el repositorio a git de manera local

Para esto debemos abrir una terminal, puede ser la de git, powershell o la que ofrece VS code por defecto y acto seguido, seguir la secuencia de comando en este orden.

1. `$ git init`
2. `$ git add .`
3. `$ git remote add origin git@gitorious.org:project/project.git`
4. `$ git commit -m "Mi primer commit"`
5. `$ git push origin maste`

Añadir el proyecto a gh-pages

1. Abrir el archivo `package.json` y agregar `homepage` al proyecto.
`"homepage": "https://myusername.github.io/my-app",`
2. Luego debemos instalar gh-pages utilizando el siguiente comando
`npm install --save gh-pages`
3. Ahora colocaremos un `predeploy` y un `deploy` en el archivo `package.json` en el apartado de scripts
`"scripts": {`
`+ "predeploy": "npm run build",`

```
+ "deploy": "gh-pages -d build",
  "start": "react-scripts start",
  "build": "react-scripts build",
```

4. Sigue hacer el deploy utilizando el siguiente comando

```
npm run deploy}
```
5. Ahora solo faltaría cambiar la rama main en el repositorio de github por la rama gh-pages

Aplicación desplegada en gh-pages

Productos

Codigo	Descripcion	Precio	Acción
1	Huawei Matebook D 15	\$15899	Agregar
2	Samsung Galaxy S10	\$13999	Agregar
3	Samsung Galaxy A01	\$1850	Agregar
4	Xiaomi Redmi Note 9s	\$5949	Agregar
5	Mochila Xiaomi	\$699	Agregar
6	Teclado Primus Gaming Ballista	\$1999	Agregar

Total: \$

Mi Carrito

No hay productos

Cuerpo de las tablas

```
const Productos = (props) => {
  return (
    <div className="Productos">
      <h1>Productos</h1>
      <Table striped bordered hover>
        <thead>
          <tr>
            <th>Codigo</th>
            <th>Descripcion</th>
            <th>Precio</th>
            <th>Acción</th>
          </tr>
        </thead>
        <tbody>
          {
            props.ProductosLista.map((a, index) =>
              <tr key={index}>
                <td>{a.codigo}</td>
                <td>{a.descripcion}</td>
                <td>${a.precio}</td>
                <td><Button onClick={()=>props.enviar(a)} variant="success">Agregar</Button></td>
              </tr>
            )
          }
        </tbody>
      </Table>
      <h4>Total: $ {props.total}</h4>
    </div>
  )
}
```

Agregar Producto

Para agregar un producto al carrito basta con presionar el botón de “agregar” que aparece en color verde, acto seguido se mandará un mensaje indicando si se registró correctamente..

Codigo	Descripcion	Precio	Acción
1	Huawei Matebook D 15	\$15899	<button>Agregar</button>



Producto agregado

Como se puede observar, el producto quedó añadido en el carrito.

The screenshot shows a web application interface. At the top is a red banner with a small product image. Below the banner, there are two main sections: 'Productos' and 'Mi Carrito'. The 'Productos' section contains a table with 6 rows of products, each with a green 'Agregar' button. The 'Mi Carrito' section contains a table with 1 row, showing the 'Huawei Matebook D 15' with a red 'eliminar' button. At the bottom, there is a 'Total: \$' label.

Codigo	Descripcion	Precio	Acción
1	Huawei Matebook D 15	\$15899	Agregar
2	Samsung Galaxy S10	\$13999	Agregar
3	Samsung Galaxy A01	\$1850	Agregar
4	Xiaomi Redmi Note 9s	\$5949	Agregar
5	Mochila Xiaomi	\$699	Agregar
6	Teclado Primus Gaming Ballista	\$1999	Agregar

Total: \$

Código reciclado y tuneado de la función enviar para el botón de agregar:

```
enviar=(objeto)=>{  
  
  Swal.fire({  
    position: 'center',  
    icon: 'success',  
    title: 'Producto agregado',  
    showConfirmButton: false,  
    timer: 4500  
  })  
  
  this.setState({  
    carrito:[...this.state.carrito,objeto]  
  })  
}
```

Eliminar

Para eliminar un producto del carrito basta con dar click en el botón de eliminar, esto lo borrará del carrito. Si la acción se realizó correctamente se mandará un mensaje confirmando dicha acción.

Codigo	Descripcion	Precio	Acción
1	Huawei Matebook D 15	\$15899	eliminar



Producto eliminado

Carrito vacío después de eliminar el retirar el producto que tenía.

Productos				Mi Carrito	
				No hay productos	
Codigo	Descripcion	Precio	Acción		
1	Huawei Matebook D 15	\$15899	Agregar		
2	Samsung Galaxy S10	\$13999	Agregar		
3	Samsung Galaxy A01	\$1850	Agregar		
4	Xiaomi Redmi Note 9s	\$5949	Agregar		
5	Mochila Xiaomi	\$699	Agregar		
6	Teclado Primus Gaming Ballista	\$1999	Agregar		

Código reciclado y tuneado de la función eliminar para el botón de eliminar.

```
eliminar=(a,index)=>{  
  Swal.fire({  
    position: 'center',  
    icon: 'warning',  
    title: 'Producto eliminado',  
    showConfirmButton: false,  
    timer: 4500  
  })  
  
  const temporal = this.state.carrito.filter((a,i)=>i!==index)  
  
  this.setState({  
    carrito:temporal  
  })  
}
```

Links (Hipervínculos)

Repositorio con las tareas:

<https://github.com/FILL16Z/Desarrollo-de-Aplicaciones>

Repositorio del examen:

<https://github.com/FILL16Z/DdA-ExamenU1yU2>

Página desplegada en gh-pages:

<https://fill16z.github.io/DdA-ExamenU1yU2/>