

## 

## [S35 - EA2: ReactJs](https://iudigital.instructure.com/courses/8653/assignments/112889)

Presentado por:

Camilo Andres Olea Aguirre

Maria Catalina Alvarez Henao

Juan Camilo Jimenez Vasquez

PRESENTADO A

Jorge Armando Julio Cruz

Docente Ingeniería Web II

Grupo 140

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA

Medellín - Septiembre 12 de 2022

Ya hemos construido el backend con Node js que se conecta con la base de datos Mongo db alojada en mongodb atlas, en este documento se construirá la aplicación desde el frontend construida con React.

React es una de las herramientas de Javascript para el desarrollo Frontend que nos permite construir las interfaces con las que interactúa el usuario o parte visual de nuestra aplicación. React nos facilita la forma de escribir HTML y CSS desde JavaScript, dando orden para mejorar nuestra experiencia como desarrolladores frontend, generando una excelente experiencia de usuario. En esta aplicación usaremos el React desde el lado del cliente.

En el siguiente link <https://es.reactjs.org/docs/create-a-new-react-app.html> encontramos los comandos a ingresar en la terminal para instalar el React dentro de nuestro proyecto y crear nuestra aplicacion: Para empezar creamos una carpeta llamada desarrollo-react, desde el visual studio code abro la carpeta y en la terminal copio el siguiente comando npx create-react-app inventarios-frontend, la terminal debera enviarnos un mensaje que dice Happy hacking !. Luego ingresamos el comando cd inventarios-frontend, luego de estar en esta carpeta inicializamos la aplicación con el comando npm start.

Dentro de nuestro IDE, abrimos la carpeta creada y limpiamos los directorios que no utilizaremos de la carpeta SRC, solo dejamos el archivo index.js. En este archivo importamos React desde React, React Dom desde React posicion cliente, La estructura de un componente en React, la cabecera se importa React, segundo el cuerpo de la función y por último se exporta la función o componente, dentro de la función se definen las variables que utilizaremos y mostraremos en nuestros componentes y se finaliza con la palabra return que determinara lo que se verá en pantalla.

En la carpeta SRC creamos un archivo App.js y allí importamos React desde React, definimos la variable App en modo flecha definiendo sus etiquetas de html y css, para luego exportarla. Luego en Index.js importamos el componente App y le solicitamos al root que renderice App.

Luego creamos los directorios ***component*** (se crean 6 módulos: inventarios, usuarios, marcas, tipos, ui), ***services*** (llamados a la API mediante el mapeado de los endpoint y se contará con un archivo por cada módulo excepto UI) y ***helpers*** (en el cual se creará un archivo transversal para ser utilizado por cualquier módulo se configura axius para que conecte a partir de una ip base para conectarte a los servicios de la aplicación) de nuestra aplicaciones, sobre la carpeta SRC.

En el módulo UI se crea un archivo llamado header.js o componente de navegación y mediante bootstrap construiremos el menú y los demás componentes; para ello mediante CDN copio el link de instalación en index.html para que descargue en nuestra aplicación todos los estilos de bootstrap y se puedan implementar. Se comienza con el navbar y se depura. En el archivo App.js se importa React from React, el BrowserRouter, router, switch, route y redirect de react router dom, header del componente UI, EstadoView del componente estado, InventarioView del componente inventario, asi mismo MarcaView, UsuarioView y TipoView.

Se agrega el NavLink desde el react router dom al header

Se crea un botón con la función de crear el inventario, este botón se crea con Bootstrap y se configura el código CSS para darle estilo y color. Luego al botón del inventario se le realiza una configuración para el modal, para su configuración completa se crean varios formularios para la captación de la información y a continuación se configura cada campo, con una lista de opciones, cabe recordar que el formulario se realiza con Bootstrap. Estos formularios son grid (columnas y filas), también se realiza una configuración para que el modal tome el tamaño completo de la página principal.

Se crea una alerta para cuando el usuario después de llenar el formulario y la de clic en guardar aparezca en la pantalla un cuadro de diálogo dando a indicar que la información se guardó.

Se configura la opción de la página del activo, el botón *ver más* de cada uno de los productos del inventario, se copia el código del formulario que realizamos anteriormente y se cambian ciertos campos.

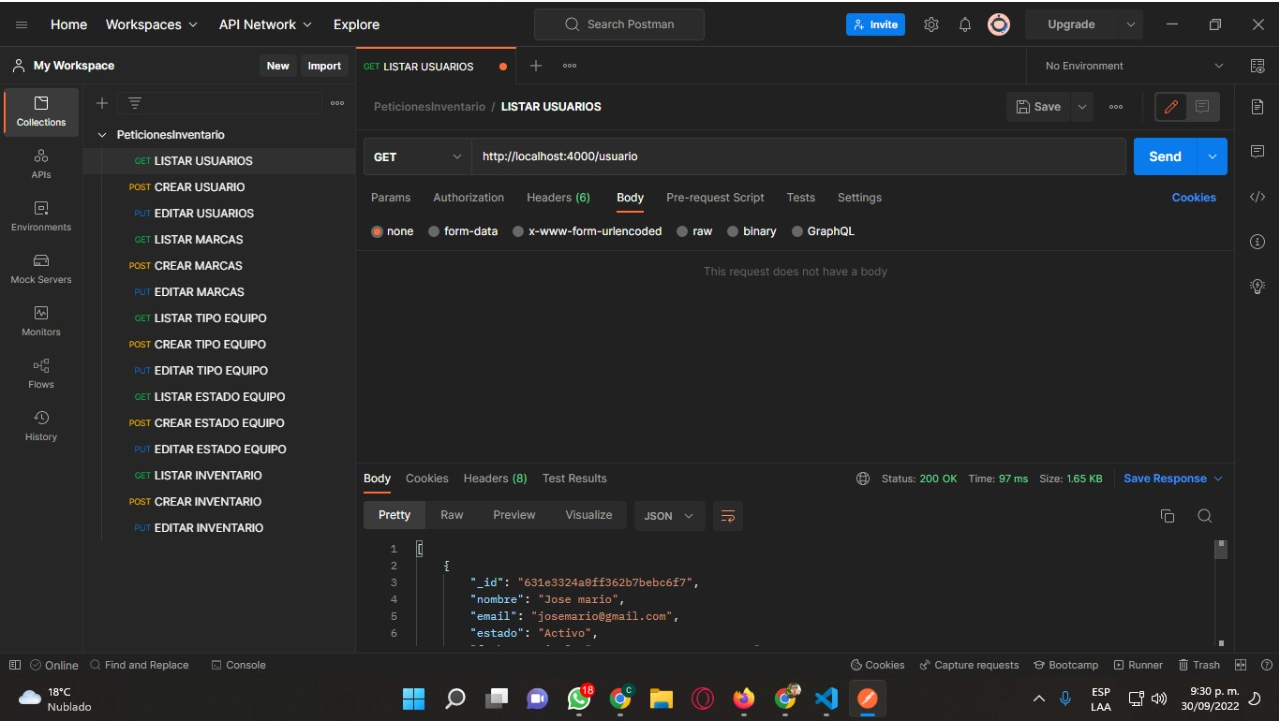
Ahora vamos a implementar el Docker. El Docker es una plataforma la cual permite crear, probar e implementar aplicaciones creadas de una forma rápida. Docker consiste en empaquetar en contenedores todo lo necesario para que se ejecute el software, se deben incluir todas las bibliotecas, las herramientas del sistema y el código para poder realizar las pruebas.

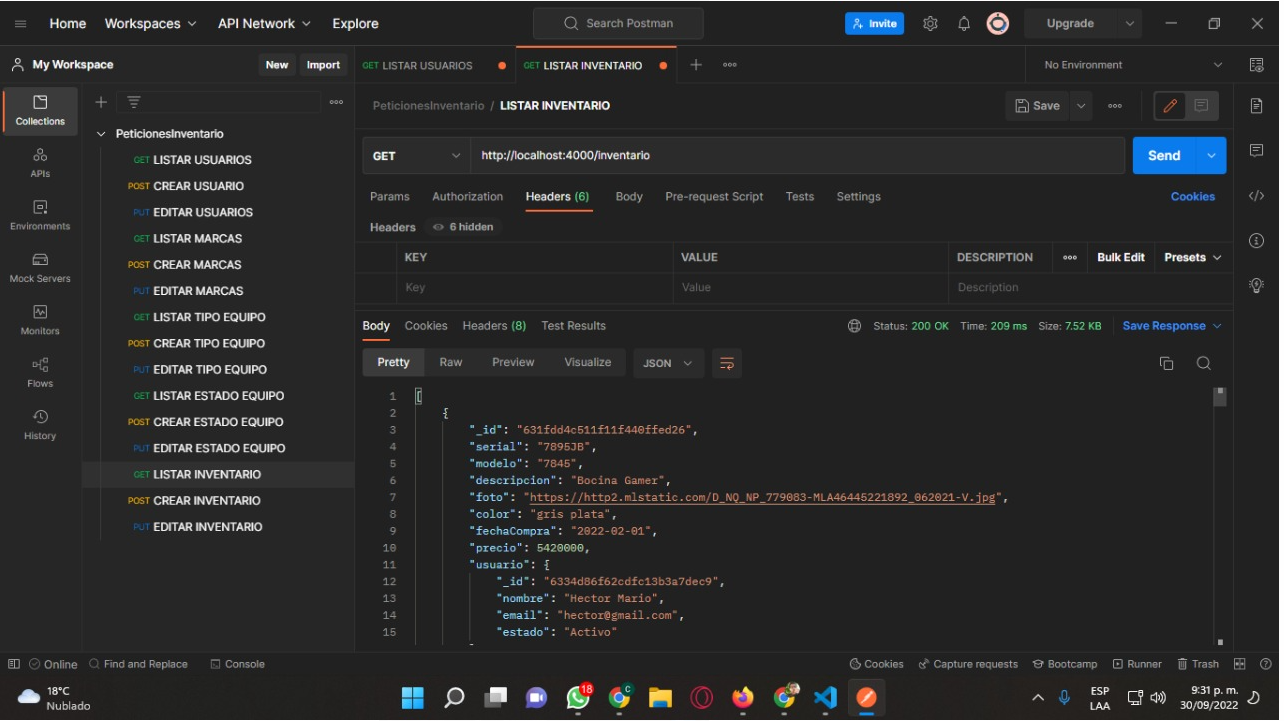
Se realiza el despliegue de Docker para realizar la prueba del backend, se sube el servicio de docker y dentro del visual code se verifica el archivo de Dockerflie, hay que tener presente que la aplicación correrá en el puerto 4000, además de estar pendiente de tener configurada la red que creamos para que tanto el backend como el frontend se puedan comunicar tiene que estar en la misma red.

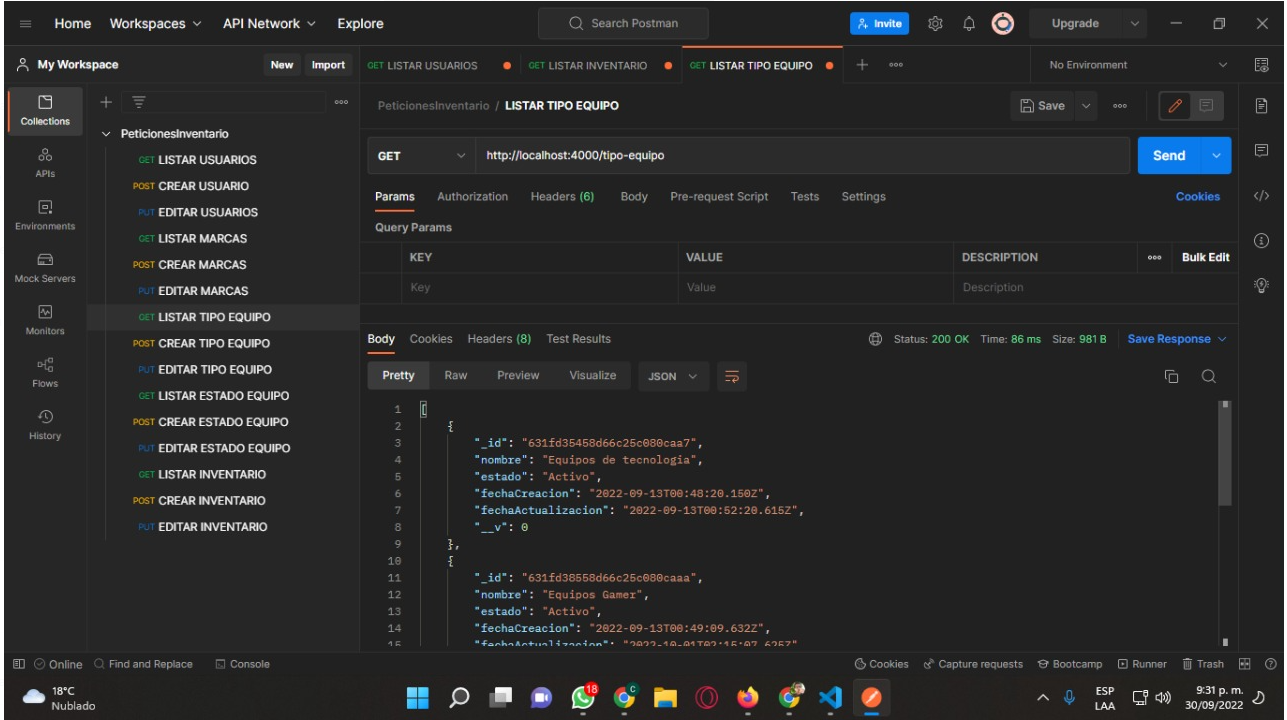
Nginx es un servidor web, este lo vamos a utilizar para poder desplegar la aplicación creada en react, realizamos la configuración en el archivo de Dockerfile con los datos del servidor web Nginx.

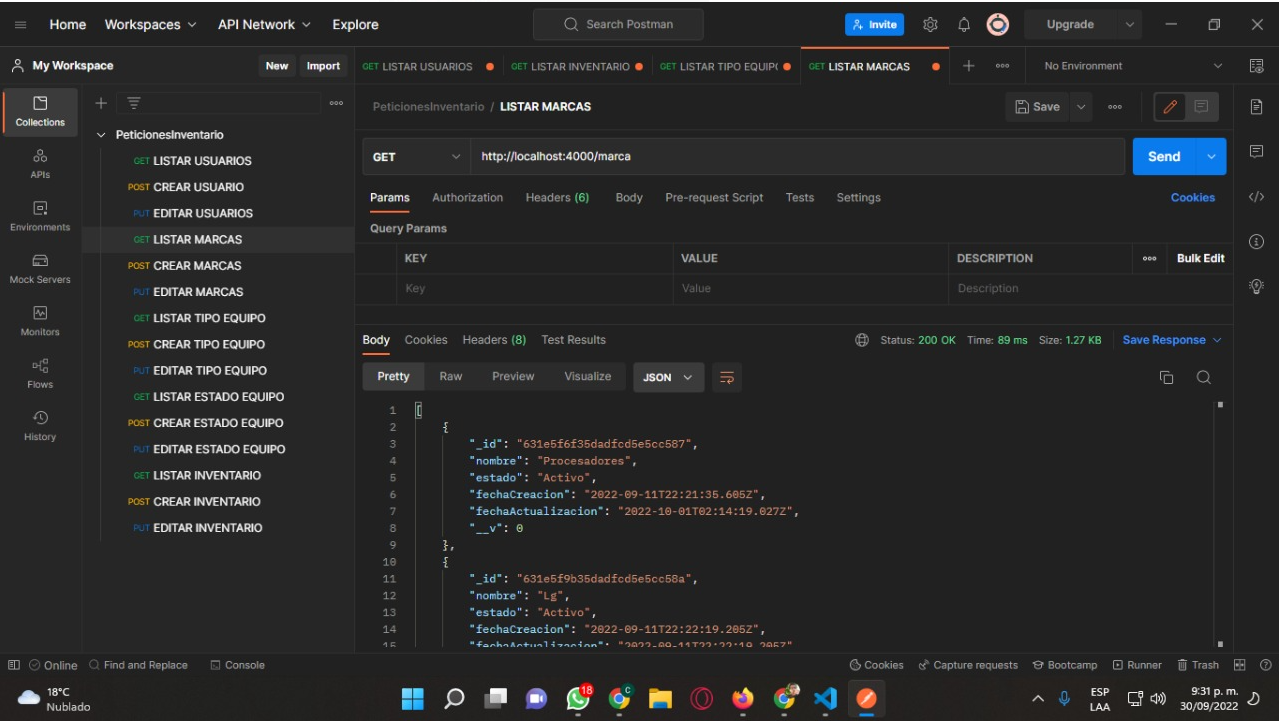
Se realiza la prueba en el navegador con la dirección de localhost:80 para verificar que la aplicación está en la red y funcionando correctamente y se realiza una prueba de crear un nuevo inventario para verificar que los campos quedan almacenados y se puede visualizar un nuevo activo en la página principal del inventario.

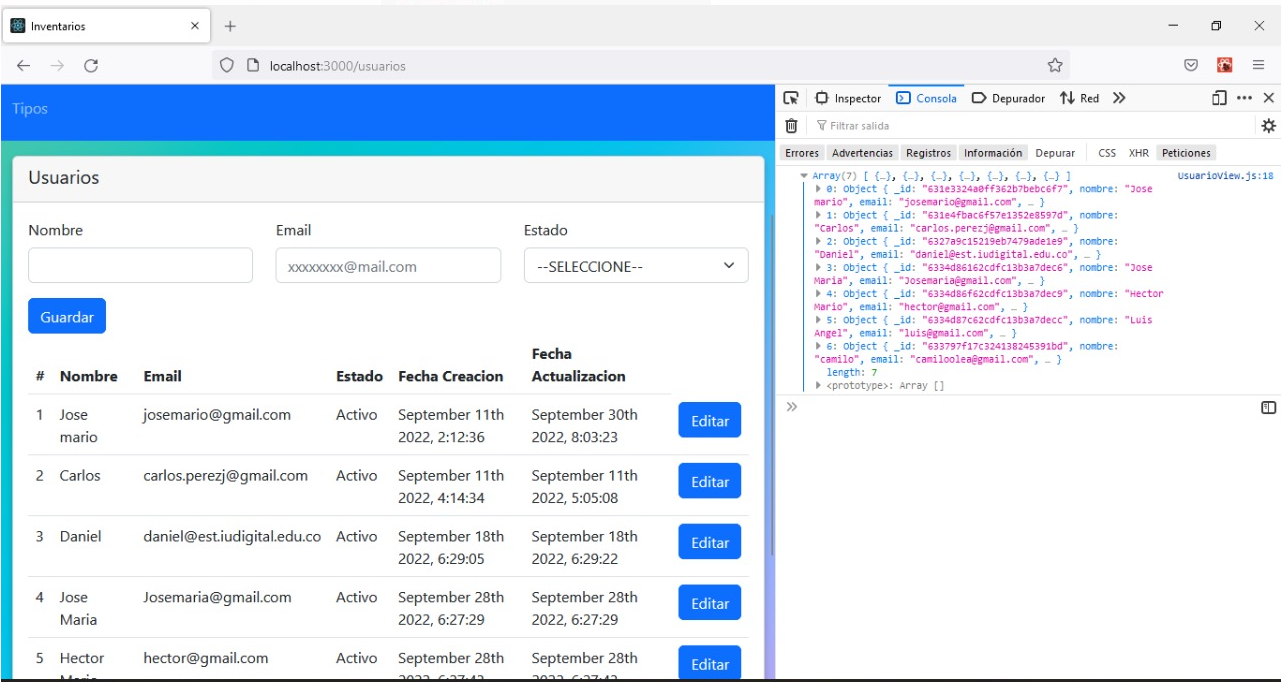
**VALIDACIONES**

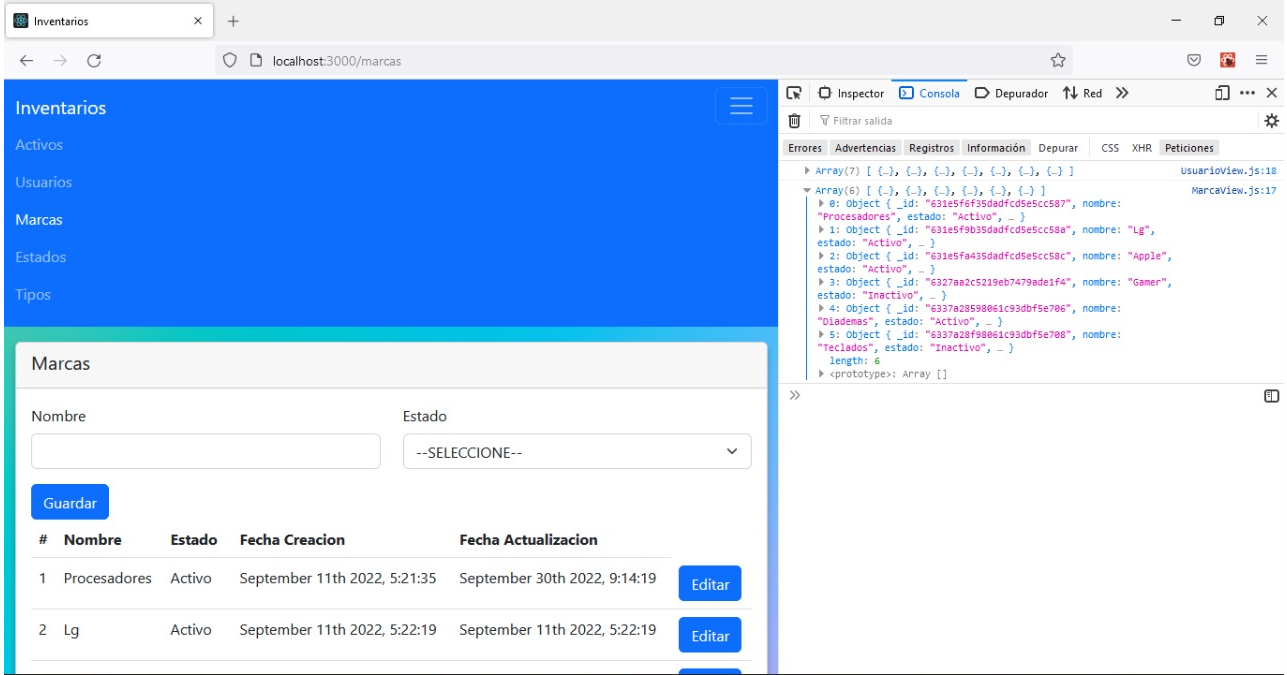


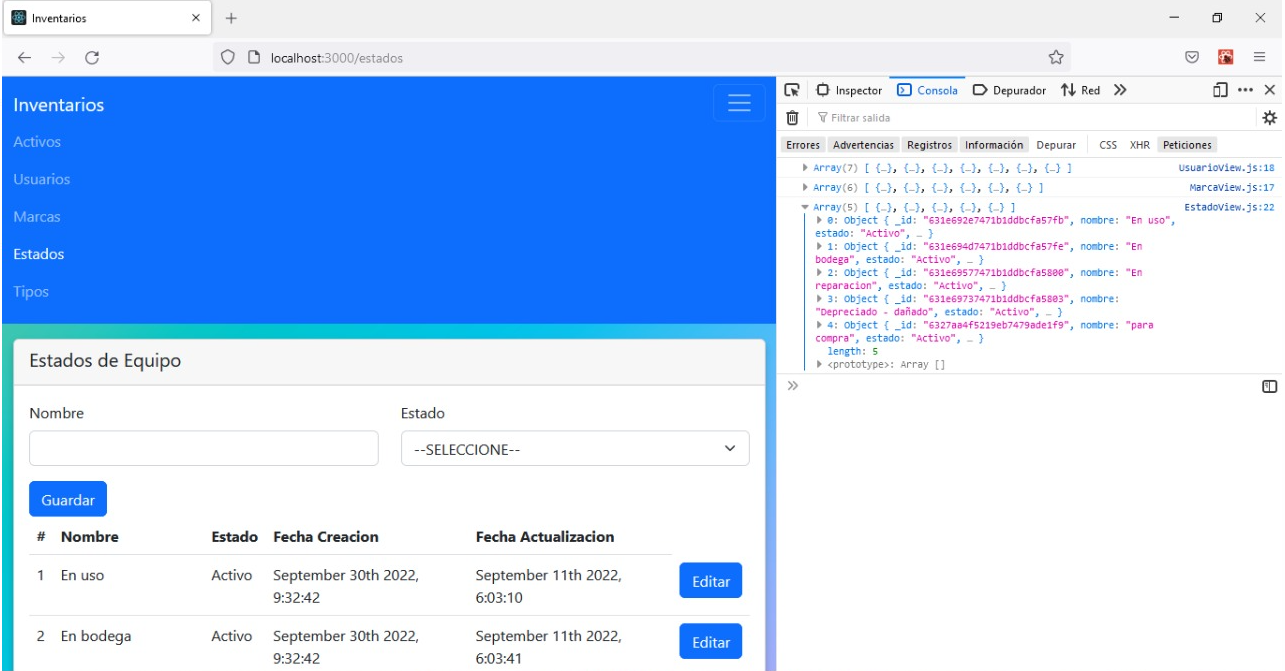




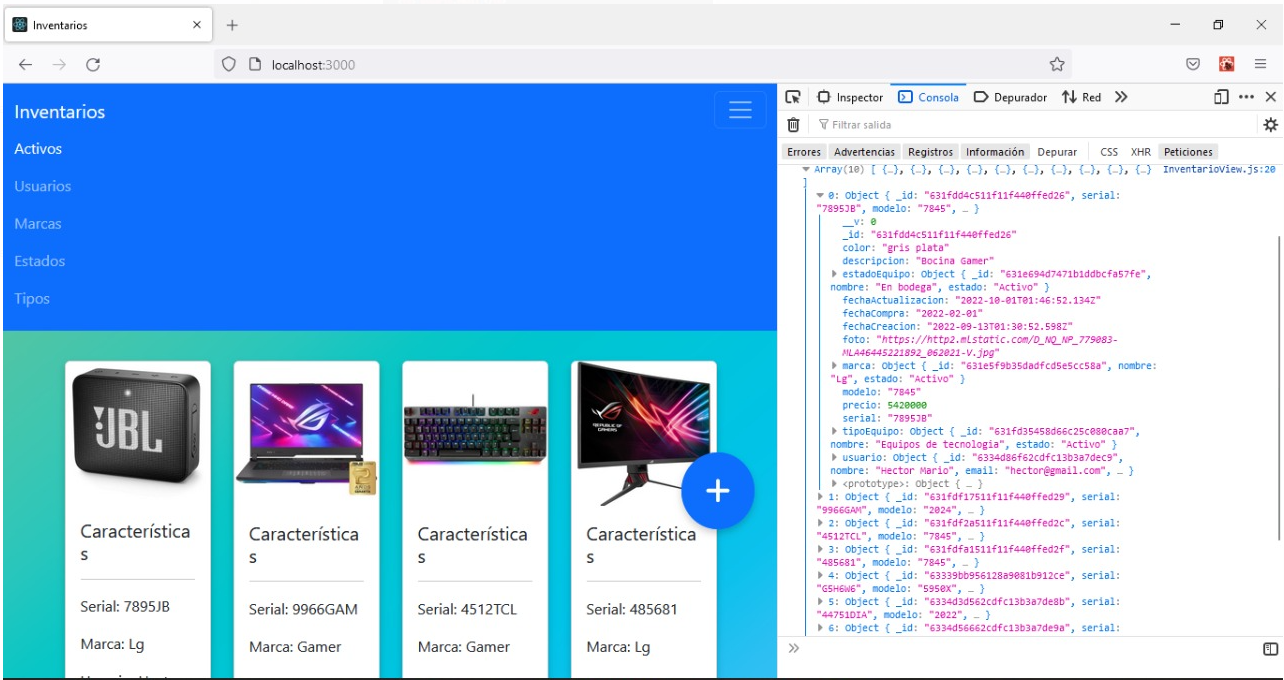












**CONCLUSIONES**

* Docker es un software que ayuda a simplificar y realizar de una forma más rápida el trabajo, además que brinda muchas herramientas y utilidades para los desarrolladores para realizar las pruebas de los proyectos a realizar.
* Implementar en proyectos tipos de arquitectura como es la de despliegue ayudan a visualizar tanto el hardware como el software de la aplicación, con esto se puede entender de una forma más clara y sencilla cómo se desplegará.