

**GUÍA DE LABORATORIO 04**

**“Contenedores Microservicios”**

## **LABORATORIO**

### **Objetivos:**

- Crear contenedores
- Ejecutar contenedores

### **Equipos, Materiales, Programas y Recursos:**

- Computadora personal.

### **Introducción:**

En el presente laboratorio se propondrán maneras de implementar contenedores Docker.

### **Seguridad:**

- Ubicar maletines y/o mochilas en el gabinete al final de aula de laboratorio.
- No ingresar con líquidos ni comida al aula de laboratorio.
- Al culminar la sesión de laboratorio, apagar correctamente la computadora.

### **Procedimiento y Resultados:**

***El laboratorio será desarrollado en forma individual.***

#### **Tarea**

1. Crear un aplicación en express que muestre las siguientes rutas,  
(su app debe ejecutarse en el contenedor en el puerto 5000)  
    /clientes (mostrar 3 clientes)  
    /productos (mostrar 3 productos)
2. Crear su repositorio en github
3. Subir su app a su repositorio
4. Crear su archivo Dockerfile (exponer el puerto 9000) y crear su imagen
5. Mostrar el URL

ip:9000  
ip:9000/clientes  
ip:9000/productos

#### **CODIGO:**

##### **App.js:**

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

app.use(bodyParser.json())
app.use(express.json());
app.set('view engine', 'ejs')
```

```

app.get('/', (req, res) => {
  const profesor = {
    nombre: "Profesor bobadilla"
  }
  res.render('index', profesor)
});

app.post('/clientes', (req, res) => {
  const clientes = req.body;
  console.log(clientes);
  res.render('clientes', clientes);
})

app.post('/productos', (req, res) => {
  const productos = req.body;
  console.log(productos);
  res.render('productos', productos);
})

const PORT = 5000;
app.listen(PORT, () => {
  console.log(`Servidor corriendo en http://localhost:${PORT}`);
});

```

**Index.ejs:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body style="display: flex; justify-content: center; align-items: center; height: 100%; background-color: black; ">
  <div >
    <p style="text-align: center; color: white;">Hola profesor <strong><%= nombre %></strong></p>
  </div>
</body>
</html>

```

**clientes.ejs:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CLIENTES</title>
</head>
<body style="background-color: black;">
  <ol style="color: white">
    <% clientes.forEach(cliente => { %>
      <li><%= cliente.nombre %></li>
    <% }); %>
  </ol>
</body>
</html>

```

**productos.ejs:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Productos</title>
</head>
<body style="background-color: black;">
  <ol style="color:white">
    <% productos.forEach(producto => { %>
      <li><%= producto.nombre %> </li>
    <% }> %>
  </ol>
</body>
</html>
```

**Dockerfile:**

```
FROM node
WORKDIR /app
COPY . .
COPY package.json package-lock.json .
RUN npm install
EXPOSE 5000
CMD ["npm", "start"]
```

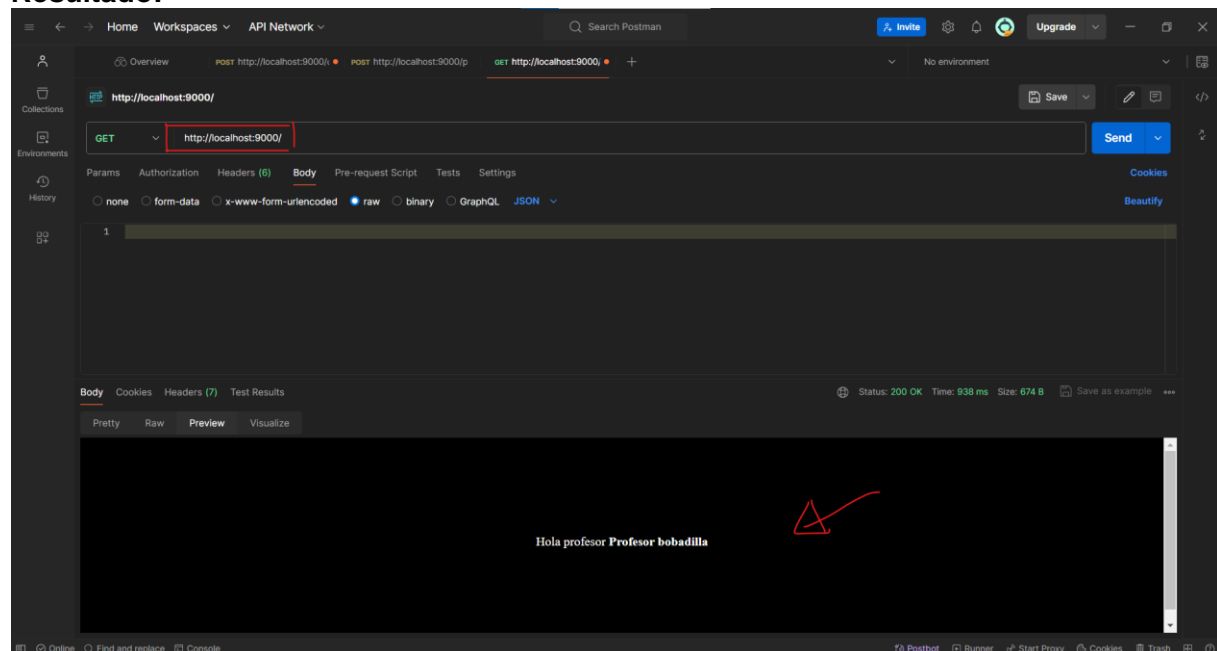
**Comandos utilizados:**

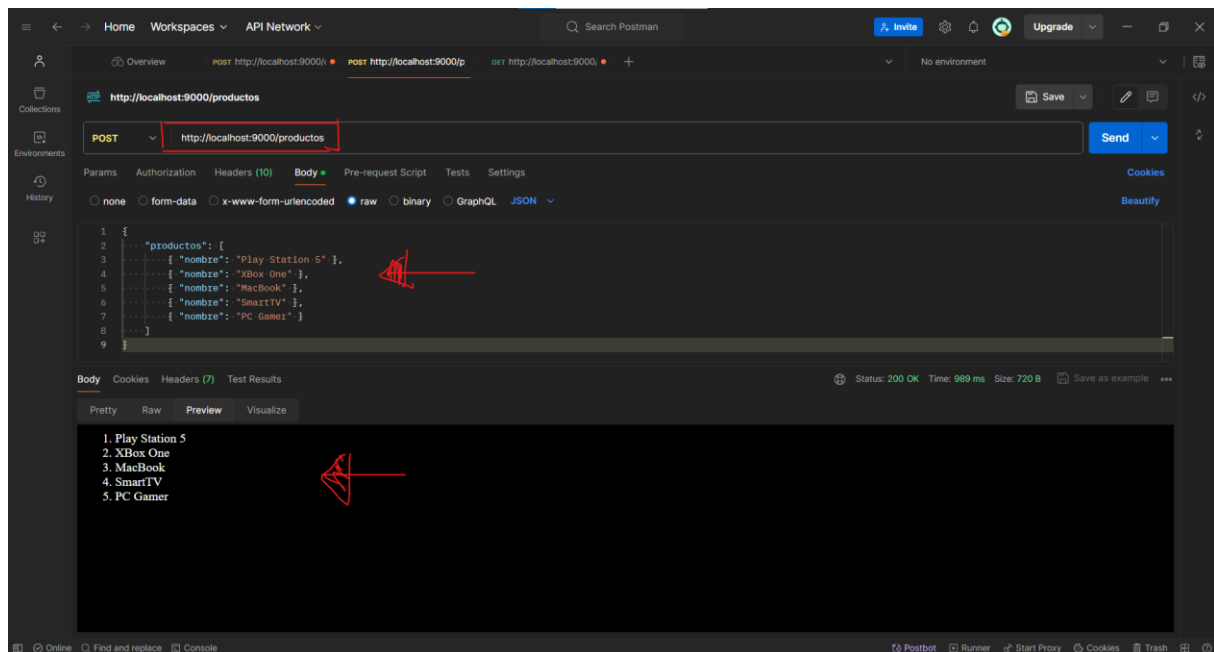
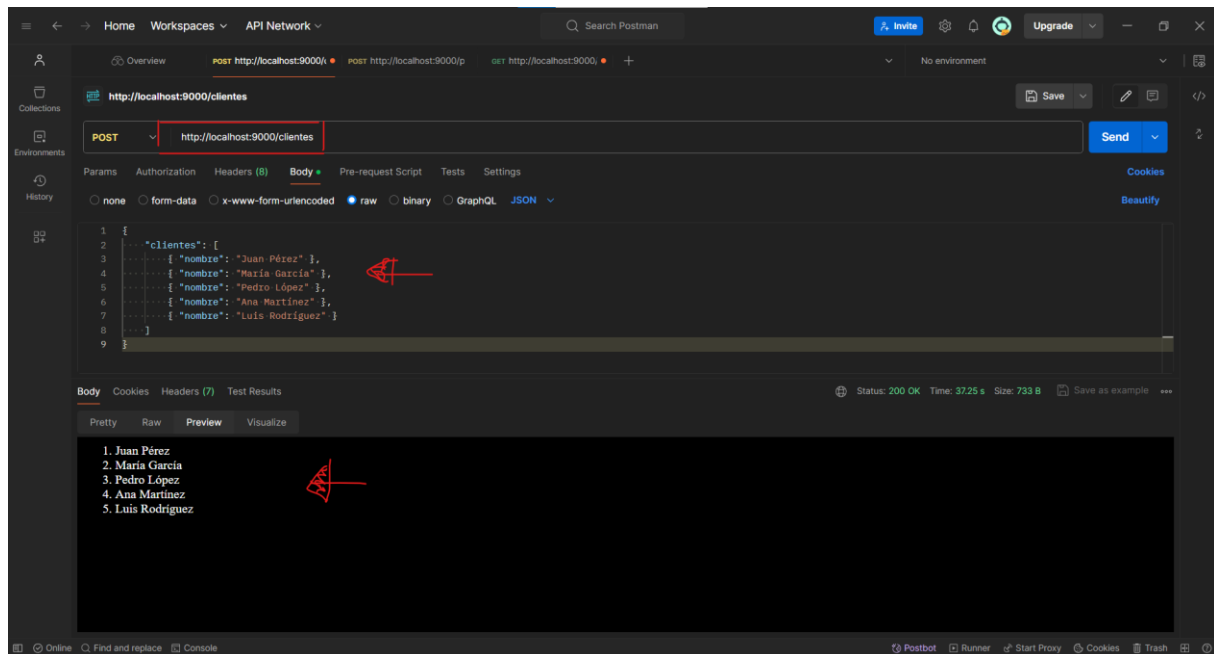
Construir la imagen llamada "laboratorio":

```
docker build -t laboratorio04 .
```

Se ejecuta el contenedor con la imagen "laboratorio04":

```
docker run -dp 9000:5000 laboratorio04
```

**Resultado:**



Hechandle un vistazo al ultimo contenedor creado:

```

C:\Users\User\Downloads\DSeN-Lab04>docker ps -l
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
9cf5d3b7b4eb   laboratorio04   "docker-entrypoint.s..." 33 minutes ago Up 32 minutes   0.0.0.0:9000->5000/tcp   exciting_chatelet

```

### Conclusiones:

1. Se logro realizar el uso de librerías de nodejs para la creación de aplicaciones web. Entre las librerías usadas se encuentran body-parser, express y ejs
2. Se logro realizar la configuración de un archivo Dockerfile con el fin de facilitar la creación, despliegue y la gestión de aplicaciones mediante el uso de imágenes y contenedores.
3. Se logro realizar la gestión de contenedores e imágenes mediante línea de comandos con el propósito de crear, detener, mostrar, ejecutar y actualizar imágenes y contenedores.

