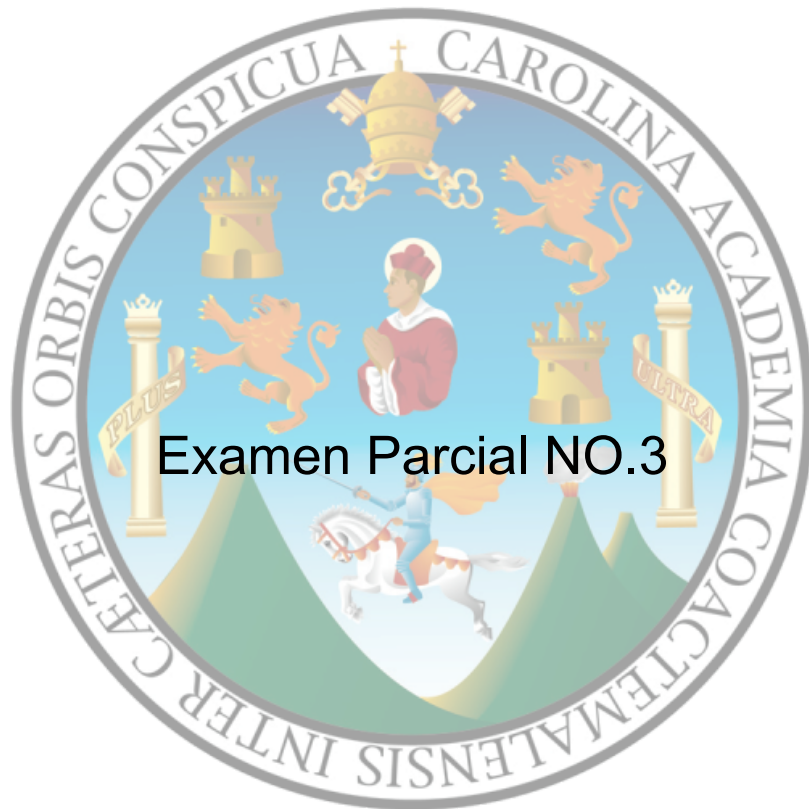


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SISTEMAS OPERATIVOS 1
SECCIÓN "N"
ING. JOAQUIN ADOLFO GUERRERO
AUX. RANDY FERNANDO JUÁREZ
SEGUNDO SEMESTRE 2023



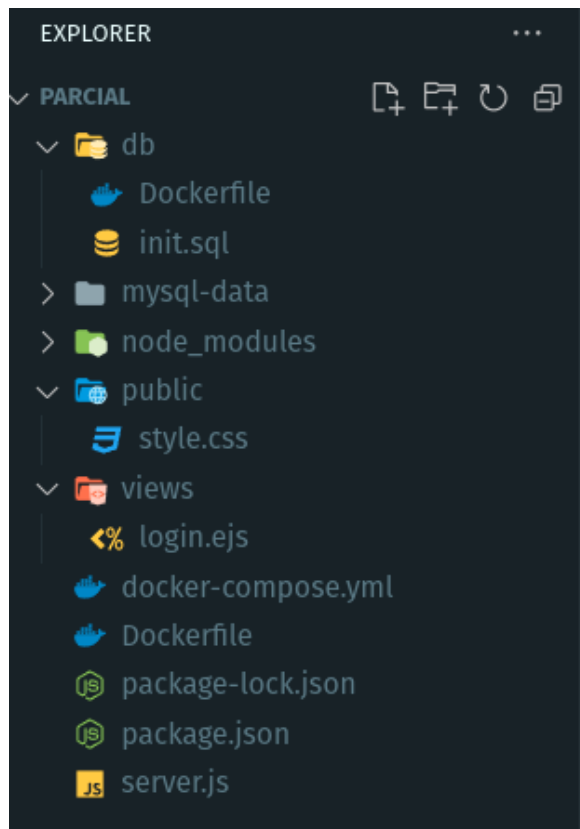
Josue Rolando Gramajo Roldan
202000895
29/12/2023

Serie I (5 Pts. / 1.45 horas). Crear una aplicación básica de login de usuario contraseña con una base de datos dentro de un contenedor docker.

Personalizar todas las configuraciones en donde aplique (contenedor, base de datos, etc.) con la letra de su primer nombre y apellido seguido de su carné, ejemplo: jguerrero201923148.

Realizar un informe y video (3 minutos máximo) de configuraciones.

Estructura del proyecto:



Configuración docker-compose:

```
docker-compose.yml
1  version: '3'
2  services:
3    mysql:
4      build:
5        ./db/
6      ports:
7        - "3310:3306"
8      environment:
9        MYSQL_ROOT_PASSWORD: jgramajo202000895
10       MYSQL_DATABASE: node_app_db
11    frontend:
12      build:
13        ./
14      ports:
15        - "5000:3000"
```

Configuración **dockerfile**:

```
Dockerfile
1  # Usa la imagen oficial de Node.js
2  FROM node:14
3
4  # Establece el directorio de trabajo en /usr/src/app
5  WORKDIR /usr/src/app
6
7  # Copia el package.json y package-lock.json para instalar dependencias
8  COPY package*.json ./
9
10 # Instala las dependencias
11 RUN npm install
12
13 # Copia los archivos de la aplicación
14 COPY . .
15
16 # Comando para ejecutar la aplicación
17 CMD ["npm", "start"]
```

Configuración **server.js** aplicación de node:

```
server.js > ...
1  const express = require('express');
2  const mysql = require('mysql2');
3  const bodyParser = require('body-parser');
4
5  const app = express();
6  const cors = require('cors');
7  const port = 3000;
8  app.use(bodyParser.urlencoded({ extended: true }));
9  app.use(cors());
10 app.use(express.static('public'));
11 app.set('view engine', 'ejs');
12
13 > const db = mysql.createConnection({
14   host: 'localhost',
15   user: 'root',
16   password: 'root',
17   database: 'users'
18 });
19
20 > db.connect(err => {
21   if (err) throw err;
22   console.log('Conectado a la base de datos');
23 });
24
25 > app.get('/', (req, res) => {
26   res.render('index');
27 });
28
29 > app.post('/login', (req, res) => {
30   const { username, password } = req.body;
31
32   const query = 'SELECT * FROM users WHERE username = ? AND password = ?';
33   console.log(req.body);
34   // console.log(password);
35   db.query(query, [username, password], (err, results) => {
36     if (err) throw err;
37     if (results.length > 0) {
38       res.render('login', { message: 'Bienvenido' });
39     } else {
40       res.render('login', { message: 'Credenciales incorrectas' });
41     }
42   });
43 });
44
45 > app.listen(port, () => {
46   console.log(`Escuchando en el puerto ${port}`);
47 });
48
49 >
```

Configuración login.js aplicación de node:

```
views > login.ejs > html > body > div.container > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>...
8  </head>
9  <body>
10   <div class="container">
11     <h2>Login</h2>
12     <form method="post" action="/login" onsubmit="handleLogin()">
13       <input type="text" name="username" placeholder="Username" required>
14       <input type="password" name="password" placeholder="Password" required>
15       <button type="submit">Login</button>
16     </form>
17
18   <script>
19     function handleLogin() {
20       const credentials = {
21         username: document.getElementsByName('username')[0].value,
22         password: document.getElementsByName('password')[0].value
23       };
24       fetch('/login', {
25         method: 'POST',
26         body: JSON.stringify(credentials),
27         headers: {
28           'Content-Type': 'application/json'
29         }
30       })
31       .then(response => response.json())
32       .then(data => {
33         if (data.success) {
34           alert(data.message);
35         } else {
36           alert(data.message);
37         }
38       })
39       .catch(error => {
40         console.error('Error:', error);
41         alert('An error occurred during login.');
```

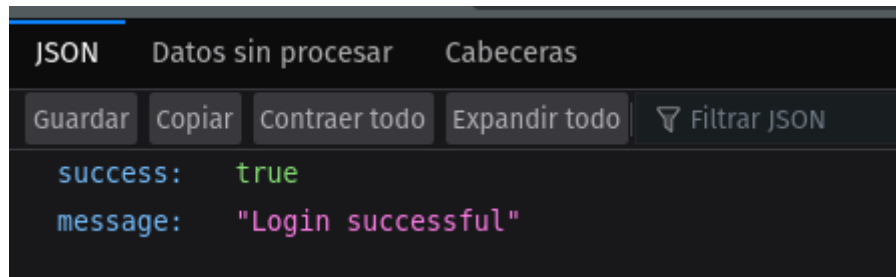
Construcción del contenedor

```
removing network partial_default...
iJosuer on ~/Escritorio/Parcial
> sudo docker-compose up --build
Creating network "partial_default" with the default driver
Building mysql
[+] Building 1.4s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 240B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/mysql:8.0
=> [internal] load build context
=> => transferring context: 30B
=> [1/2] FROM docker.io/library/mysql:8.0
=> CACHED [2/2] COPY ./init.sql /docker-entrypoint-initdb.d/
=> exporting to image
=> => exporting layers
=> => writing image sha256:774679d872862a89390ff1c5d540631cc9636a581e3c1
=> => naming to docker.io/library/partial_mysql
Building frontend
[+] Building 12.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 449B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:14
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c59
=> [internal] load build context
=> => transferring context: 199.09kB
=> CACHED [2/5] WORKDIR /usr/src/app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install
=> [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:bd945786a08148e56b5eace228e9c10afd1d4661b5038
=> => naming to docker.io/library/partial_frontend
```

EJECUCIÓN

Login

Login



LINK VIDEO

<https://drive.google.com/drive/u/1/folders/1guy-MDPRUuoJL5mf5IAZbkOj5IObA1KM>