Práctica 2 HTTPS

Vamos a crear un **despliegue completo** en Docker:

- Frontend React/Vite servido con **Nginx(servidor inverso) + HTTPS**
- **MySQL + phpMyAdmin**
- Backend Express con **CORS** y **variables de entorno**

1. Crea la estructura de archivos con los que se proporcionan y levanta el despliegue, te vas a encontrar un fallo que debes de solucionar para que funcione plenamente el servidor ngninx como proxy inverso y que puedas ver la api desde https://localhost/api/saludo. Debes de indicar el fallo cuál es y mostrar pantallazos de https://localhost, https://localhost/api/saludo, https://localhost/api/despedida, http:localhost:5000/api/saludo y http://localhost:5000/api/despedida

proyecto/
│
├── backend/
│ ├── index.js
│ ├── package.json
│ ├── .env.development
│ ├── .env.production
│
├── frontend/
│ ├── index.html
│ ├── main.jsx
│ ├── style.css
│ ├── package.json
│ ├── vite.config.js
│ ├── nginx.conf
│ └── certs/
│ ├── selfsigned.crt
│ └── selfsigned.key
│
└── docker-compose.yml

2. Análisis del backend/package.json

```
{
 "name": "backend",
 "version": "1.0.0",
 "main": "index.js",
 "scripts": {
```

```
    "start": "node index.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1"
  }
}
```

- https://www.npmjs.com/package/dotenv visita esta web y explica porque instalamos dotenv como dependencia lo más extenso posible.

3. Uso básico de CORS en Express

Si llamas a `https://localhost/api/saludo` desde tu frontend, el navegador **puede bloquear la petición** por CORS (Cross-Origin Resource Sharing), porque el frontend y backend tienen **orígenes distintos** (`https://localhost` vs `http://backend:5000` internamente en Docker).

- `app.use(cors())` → permite **todas las peticiones de cualquier origen**.
- Para producción, lo recomendable es **restringir el origen** a tu frontend:

```
app.use(cors({
  origin: 'https://localhost' // o tu dominio real
}));
```

```
// backend/index.js
const express = require('express');
const cors = require('cors');   // Importamos CORS
const app = express();
const port = 5000;

// Permitir cualquier origen (solo para desarrollo)
app.use(cors());

// Ruta de ejemplo
app.get('/api/saludo', (req, res) => {
   res.json({ mensaje: '¡Hola desde el backend!' });
});

app.listen(port, () => console.log(`Backend escuchando en ${port}`));
```

Explicación

4. Añade una api llamada /api/servidor que muestre el mensaje “servidor inverso”

```
const express = require('express');
const cors = require('cors');
require('dotenv').config(); // carga variables de entorno

const app = express();
const port = process.env.PORT || 5000;
```

```
// Configuración CORS: permite solo el frontend servido por Nginx
app.use(cors({
  origin: 'https://localhost'
}));

// api
app.get('/api/saludo', (req, res) => {
    res.json({ mensaje: '¡Hola desde el backend!' });
});

app.listen(port, () => console.log(`Backend escuchando en ${port}`));
```

5. **Añade un tercer botón nuevo que muestre la api servidor**
6. **Usa tu cuenta de github para subir este despliegue escribe la dirección del mismo.**