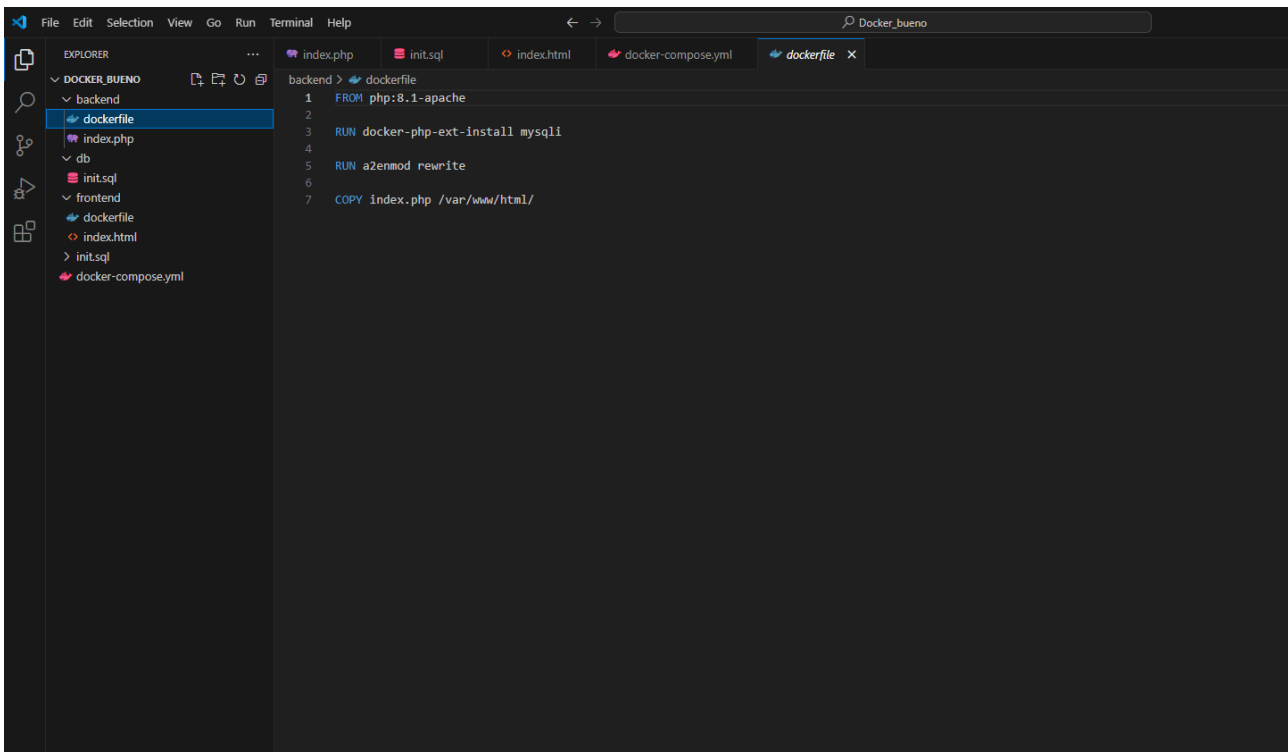


## PRÁCTICA DOCKER(JOSÉ MARÍA ANGULO RIVAS)

### Creación de los contenedores:

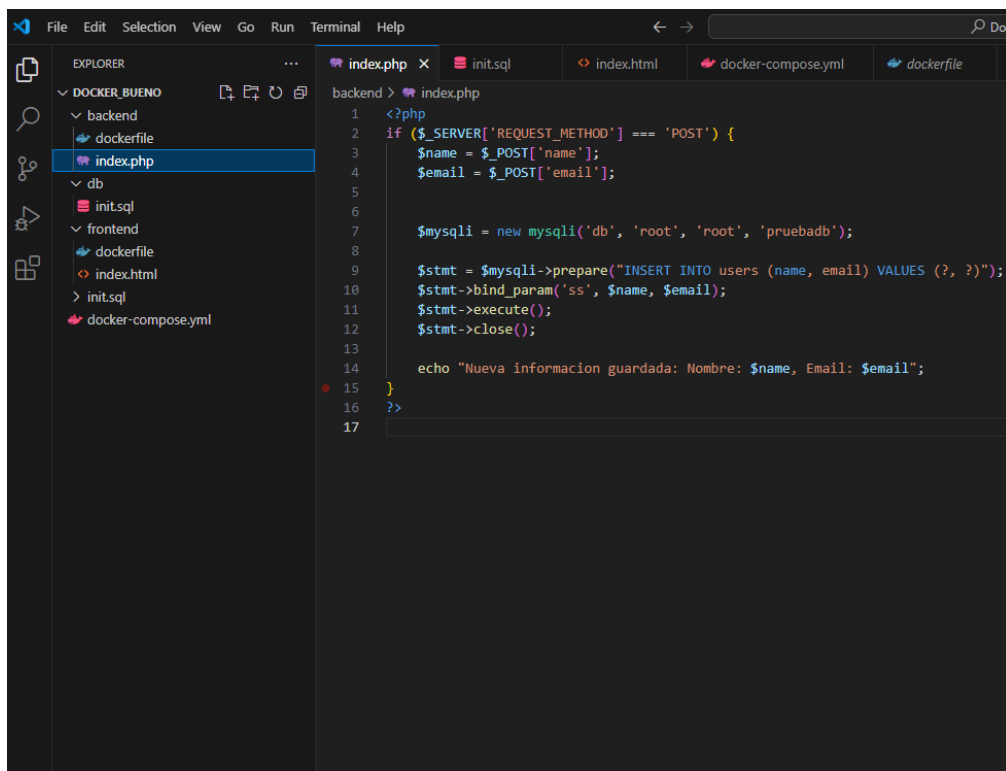
Para comenzar vamos a crear los archivos en los cuales se van a iniciar en los contenedores, primero el dockerfile del backend:



The screenshot shows the Visual Studio Code editor with the 'dockerfile' file open in the 'backend' directory. The file contains the following Dockerfile instructions:

```
1 FROM php:8.1-apache
2
3 RUN docker-php-ext-install mysqli
4
5 RUN a2enmod rewrite
6
7 COPY index.php /var/www/html/
```

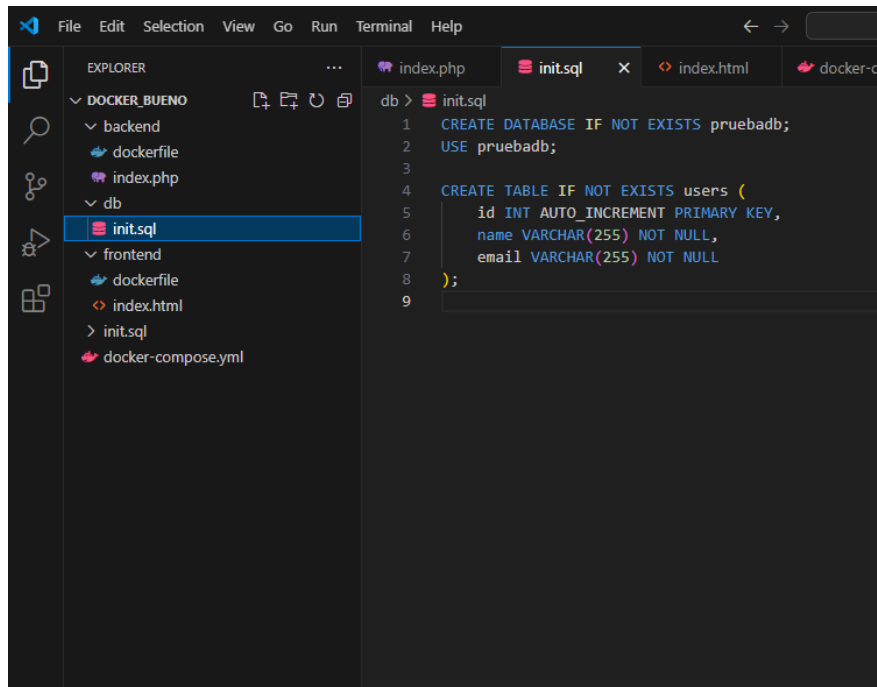
Ahora el index.php:



The screenshot shows the Visual Studio Code editor with the 'index.php' file open in the 'backend' directory. The file contains the following PHP code:

```
1 <?php
2 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3     $name = $_POST['name'];
4     $email = $_POST['email'];
5
6
7     $mysqli = new mysqli('db', 'root', 'root', 'pruebasdb');
8
9     $stmt = $mysqli->prepare("INSERT INTO users (name, email) VALUES (?, ?)");
10    $stmt->bind_param('ss', $name, $email);
11    $stmt->execute();
12    $stmt->close();
13
14    echo "Nueva informacion guardada: Nombre: $name, Email: $email";
15 }
16 ?>
17
```

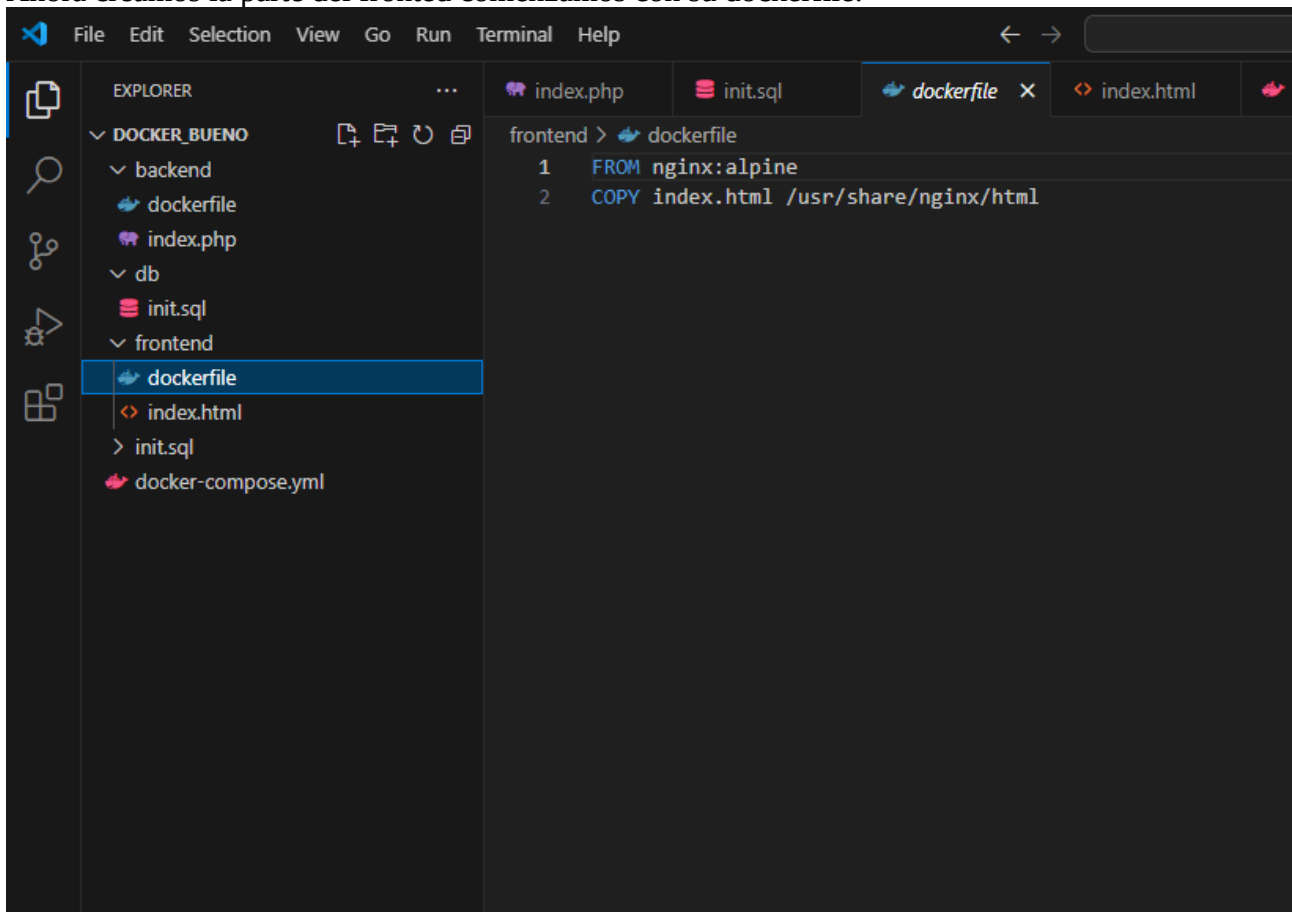
Despues creamos el la tabla de la base de datos:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The project is named 'DOCKER\_BUENO'. Under the 'db' folder, the 'init.sql' file is selected. The main editor displays the SQL code for this file.

```
db > init.sql
1 CREATE DATABASE IF NOT EXISTS pruebadb;
2 USE pruebadb;
3
4 CREATE TABLE IF NOT EXISTS users (
5     id INT AUTO_INCREMENT PRIMARY KEY,
6     name VARCHAR(255) NOT NULL,
7     email VARCHAR(255) NOT NULL
8 );
9
```

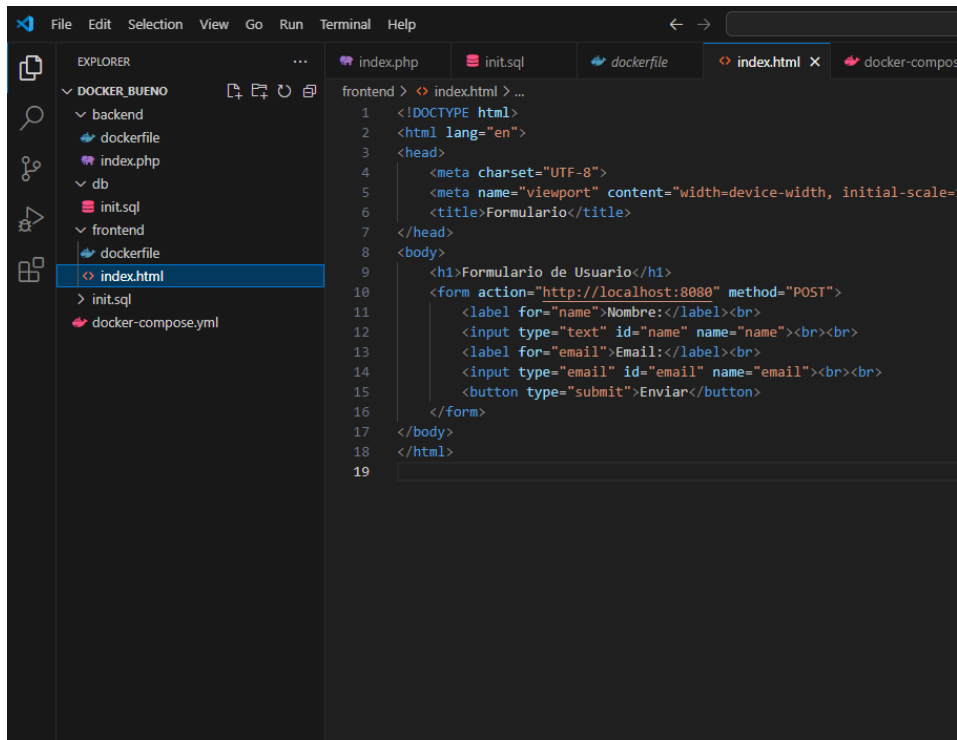
Ahora creamos la parte del fronted comenzamos con su dockerfile:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The project is named 'DOCKER\_BUENO'. Under the 'frontend' folder, the 'dockerfile' file is selected. The main editor displays the Dockerfile code for this file.

```
frontend > dockerfile
1 FROM nginx:alpine
2 COPY index.html /usr/share/nginx/html
```

y por ultimo el index.html que contendrá el formulario del usuario:



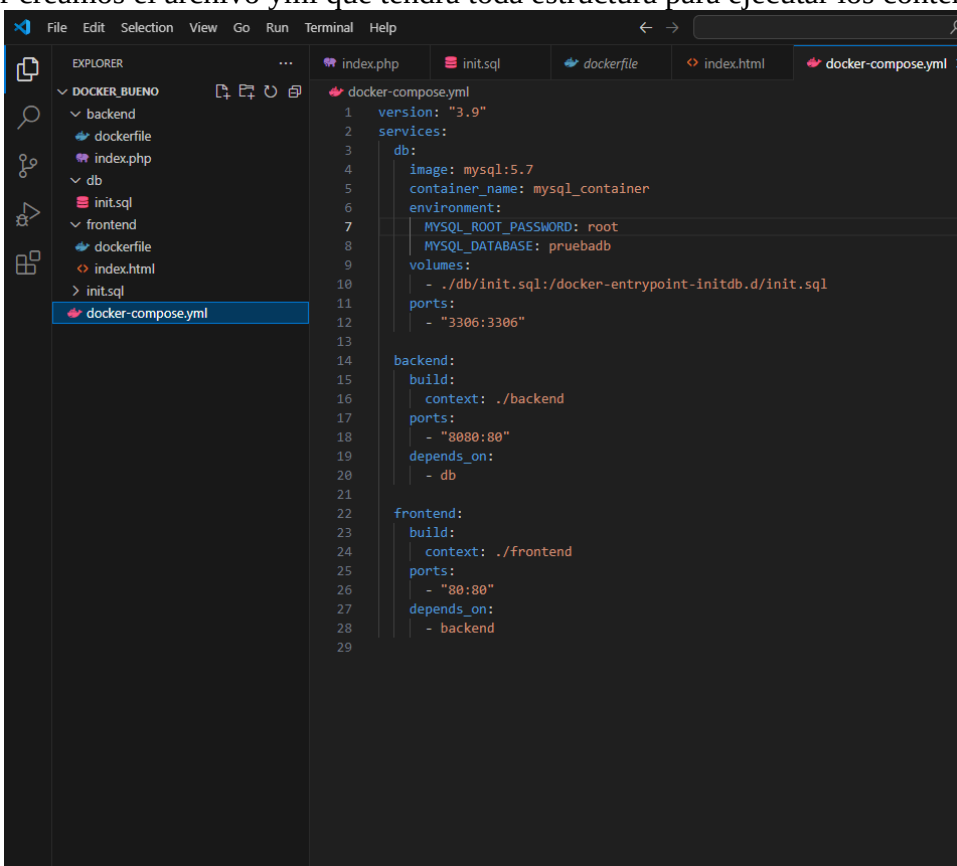
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The project structure is as follows:

- DOCKER\_BUENO
  - backend
    - dockerfile
  - db
    - index.php
    - init.sql
  - frontend
    - dockerfile
    - index.html (selected)
- docker-compose.yml

The main editor displays the content of `index.html`:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Formulario</title>
7 </head>
8 <body>
9   <h1>Formulario de Usuario</h1>
10  <form action="http://localhost:8080" method="POST">
11    <label for="name">Nombre:</label><br>
12    <input type="text" id="name" name="name"><br><br>
13    <label for="email">Email:</label><br>
14    <input type="email" id="email" name="email"><br><br>
15    <button type="submit">Enviar</button>
16  </form>
17 </body>
18 </html>
19
```

Para finalizar creamos el archivo `yml` que tendrá toda estructura para ejecutar los contenedores



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The project structure is the same as the previous screenshot, but now `docker-compose.yml` is selected in the Explorer and its content is visible in the main editor:

```
1 version: "3.9"
2 services:
3   db:
4     image: mysql:5.7
5     container_name: mysql_container
6     environment:
7       MYSQL_ROOT_PASSWORD: root
8       MYSQL_DATABASE: pruebadb
9     volumes:
10       - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql
11     ports:
12       - "3306:3306"
13
14   backend:
15     build:
16       context: ./backend
17     ports:
18       - "8080:80"
19     depends_on:
20       - db
21
22   frontend:
23     build:
24       context: ./frontend
25     ports:
26       - "80:80"
27     depends_on:
28       - backend
29
```

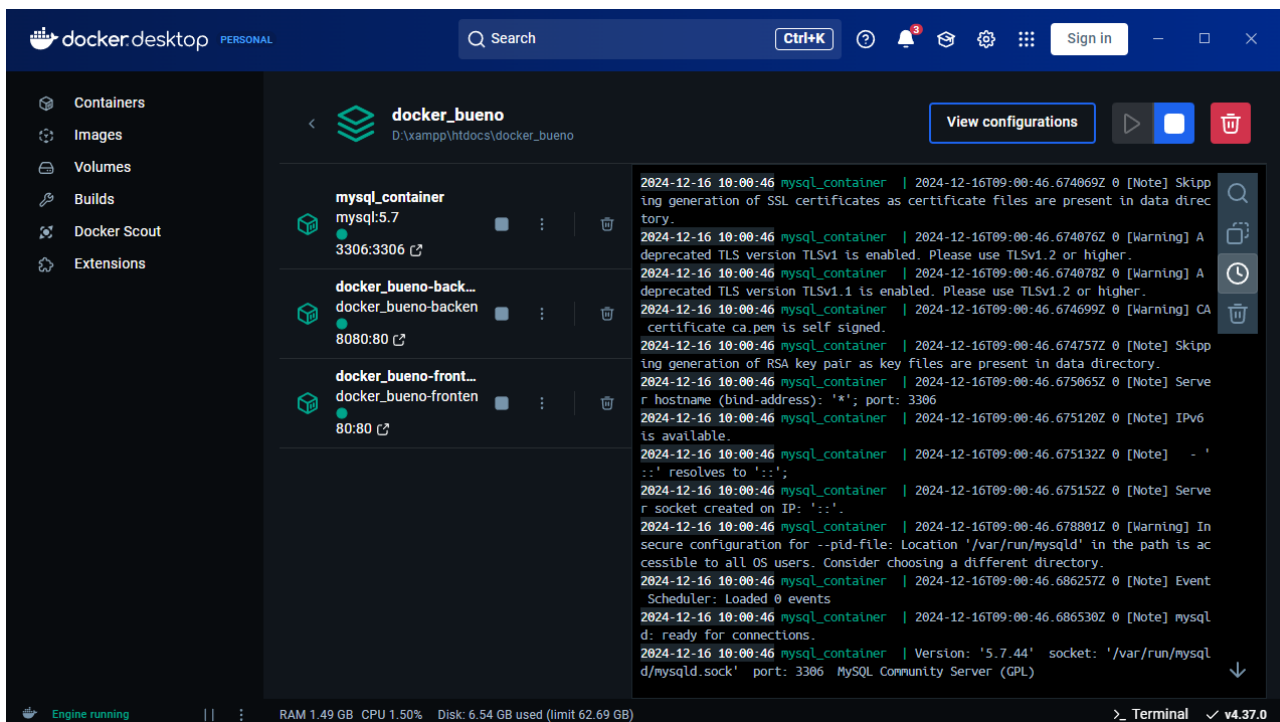
Ahora nos colocamos desde git en nuestro proyecto y ponemos `docker-compose up --build` para que nos cree los contenedores

```

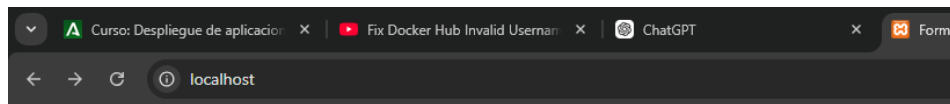
profesor@DESKTOP-3LHBG1B MINGW64 /d/xampp/htdocs/docker_bueno
$ docker-compose up --build
time="2024-12-16T10:00:32+01:00" level=warning msg="project has been loaded without an explicit name from a symlink. Using name \"docker_bueno\""
time="2024-12-16T10:00:32+01:00" level=warning msg="D:\\xampp\\htdocs\\docker_bueno\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 3.1s (18/18) FINISHED                                docker:desktop-linux
=> [backend internal] load build definition from dockerfile      0.0s
=> => transferring dockerfile: 149B                             0.0s
=> [backend internal] load metadata for docker.io/library/php:8.1-apache 0.0s
=> [backend internal] load .dockerignore                        0.0s
=> => transferring context: 2B                                    0.0s
=> [backend 1/4] FROM docker.io/library/php:8.1-apache@sha256:c8c0f3429a44974cdef762deee7b99a07a61ea04dcbf4814f0 0.0s
=> => resolve docker.io/library/php:8.1-apache@sha256:c8c0f3429a44974cdef762deee7b99a07a61ea04dcbf4814f04a8b161c 0.0s
=> [backend internal] load build context                        0.0s
=> => transferring context: 31B                                    0.0s
=> CACHED [backend 2/4] RUN docker-php-ext-install mysqli      0.0s
=> CACHED [backend 3/4] RUN a2enmod rewrite                    0.0s
=> CACHED [backend 4/4] COPY index.php /var/www/html/          0.0s
=> [backend] exporting to image                                0.2s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:b1ae093f7967bebd7d664b365bcfddcc963accff7af10909dfbffd99ba1610c 0.0s
=> => exporting config sha256:87fe2d6afd756195f7a0652c6af4c8a3900efc260b0a28c8b810ce42725050b0 0.0s
=> => exporting attestation manifest sha256:fb13331abc115cb02b57af9be904d7030a970c42ca162937d138aee5f03fd112 0.0s
=> => exporting manifest list sha256:18797464ace1e67c73defea1801daba39eee503846a1585c48fabc28f7271c0b 0.0s
=> => naming to docker.io/library/docker_bueno-backend:latest 0.0s
=> => unpacking to docker.io/library/docker_bueno-backend:latest 0.0s
=> [backend] resolving provenance for metadata file            0.0s

```

Una vez que se ha iniciado y se ha creado podemos ver que se han creado e iniciado en el docker desktop



Ahora si nos vamos a nuestro localhost se nos tendrá que abrir nuestro formulario

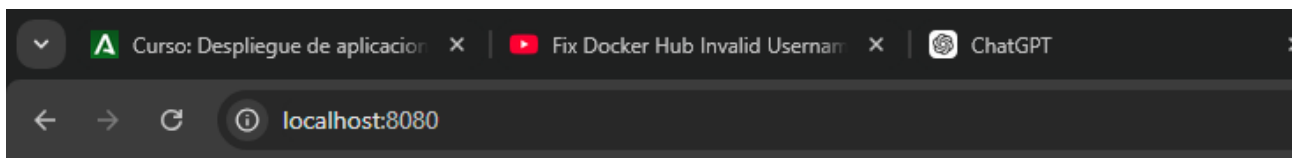


## Formulario de Usuario

Nombre:

Email:

Si todo a funcionado al darle a enviar se nos meterá un nuevo usuario



Nueva informacion guardada: Nombre: Jose Maria, Email: josemariaangulorivas1023@gmail.com

Ahora si queremos ver los usuarios que hemos metido desde la terminal ponemos el comando docker ps y nos mostrará nuestros contenedores activos

```
profesor@DESKTOP-3LHBG1B MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
e3dfb78ee234   docker_bueno-frontend              "/docker-entrypoint..." 3 minutes ago  Up 3 minutes  0.0.0.0:80->80/tcp
f9359da47838   docker_bueno-frontend-1            "/docker-entrypoint..." 3 minutes ago  Up 3 minutes  0.0.0.0:8080->80/tcp
2af036600ae0   docker_bueno-backend-1             "docker-php-entrypoi..." 3 minutes ago  Up 3 minutes  0.0.0.0:3306->3306/tcp,
mysql_container mysql:5.7
```

Copiamos el código de nuestro contenedor y ponemos `docker exec -i "nuestro código" bash` después ponemos `mysql -u root -p` y nuestra contraseña. Una vez dentro ponemos `use "el nombre de nuestra database"` y ponemos `select * from "nuestra tabla"` y se nos mostrarán los datos

```
MINGW64/c/Users/profesor
Learn more at https://docs.docker.com/go/debug-cli/

profesor@DESKTOP-3LHBG1B MINGW64 ~
$ docker exec -it 2af036600ae0 bash
bash-4.2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use pruebaadb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from users;
+----+-----+-----+
| id | name  | email                                     |
+----+-----+-----+
| 1  | Jose Maria | josemariaangulorivas1023@gmail.com |
| 2  | Jose Maria | josemariaangulorivas1023@gmail.com |
+----+-----+-----+
```