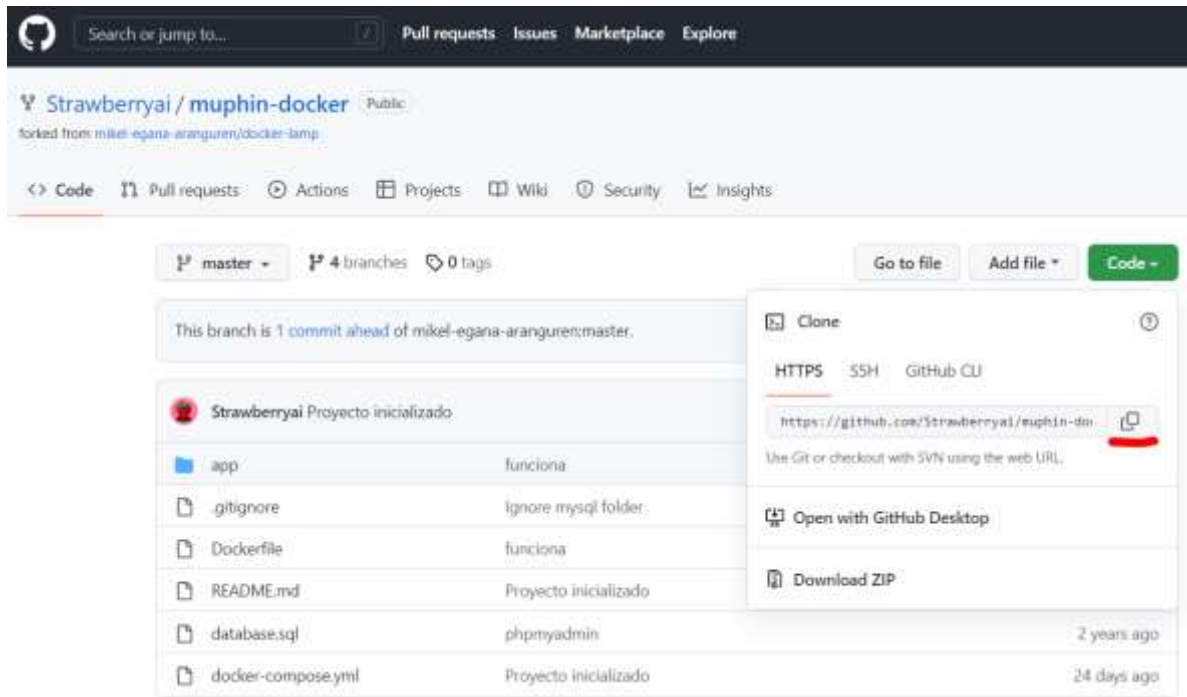


## REQUISITOS

Para empezar a usar el sistema, primero debemos clonar el repositorio desde GitHub.

Link para acceder a nuestro repositorio: <https://github.com/Strawberryai/muphin-docker>

Esto lo hacemos con la opción de clonar mediante HTTPS:



Después de copiar el link, abrimos la terminal y ejecutamos el comando en el directorio que queremos:

```
$ git clone *link*
```

siendo el link: <https://github.com/Strawberryai/muphin-docker.git>

```
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo$ git clone https://github.com/Strawberryai/muphin-docker.git
Clonando en 'muphin-docker'...
remote: Enumerating objects: 555, done.
remote: Total 555 (delta 0), reused 0 (delta 0), pack-reused 555
Recibiendo objetos: 100% (555/555), 1.69 MiB | 159.00 KiB/s, listo.
Resolviendo deltas: 100% (322/322), listo.
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo$
```

Ya clonado el repositorio, cambiamos de rama a “Entrega1”, con el comando:

```
$ git checkout Entrega1      o      $sudo git checkout Entrega1      (en caso de error)
```

```
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo/muphin-doc...
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$ git checkout Entrega1
Ya en 'Entrega1'
Tu rama está actualizada con 'origin/Entrega1'.
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$
```

Ahora lo

que debemos hacer son las instrucciones adjuntadas en el [READ.ME](#)

```
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo/muphin-doc...
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$ docker build -t="web" .
Sending build context to Docker daemon 1.851MB
Step 1/2 : FROM php:7.2.2-apache
--> 08685253e459
Step 2/2 : RUN docker-php-ext-install mysqli
--> Using cache
--> 59a186e99d2f
Successfully built 59a186e99d2f
Successfully tagged web:latest
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$ docker-compose up -d
Creating network "muphin-docker_default" with the default driver
Creating muphin-docker_db_1 ... done
Creating muphin-docker_phpmyadmin_1 ... done
Creating muphin-docker_web_1 ... done
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$
```

## ERRORES

1) Puede ser que nos de un error al hacer el “docker build”

Habría que añadir “sudo al comando”:

```
$sudo docker build -t="web" .
```

2) En caso de error al hacer el “docker-compose up”, paramos los servicios con:

```
$docker-compose down
```

Puede pasar que se esté ejecutando en el mismo puerto otro proceso, así que miramos con el comando:

```
$ docker ps
```

```
adrian@adrian-VirtualBox: ~/Escritorio/ejemplo/muphin-doc...
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
adrian@adrian-VirtualBox:~/Escritorio/ejemplo/muphin-docker$
```

En el caso de que haya un proceso en ejecución en uno de los puertos que vamos a utilizar (81 y 8889), eliminamos el proceso:

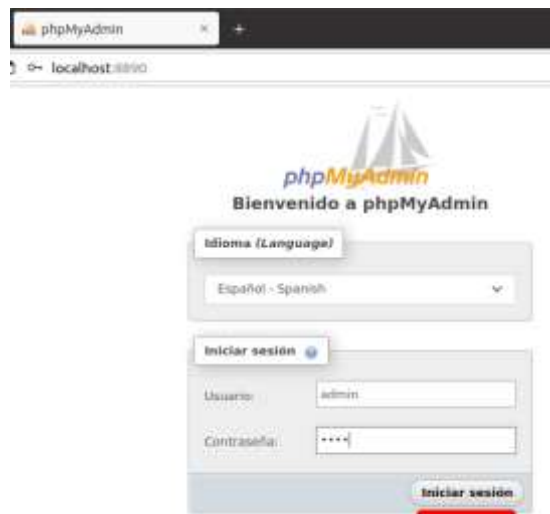
```
$ docker kill <Container id>
```

Volvemos a poner en marcha los servicios con:  
\$ docker-compose up -d

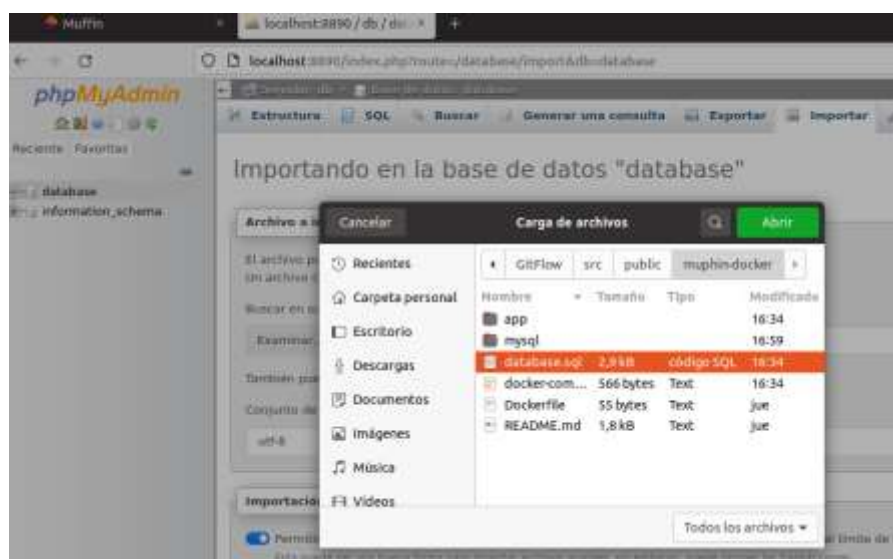
## BASE DE DATOS

Para empezar a usar la web, debemos abrir PHPMyAdmin para importar la base de datos. Vamos a nuestro navegador y usamos la dirección: <http://localhost:8890>  
Iniciamos la sesión, con:

Usuario: admin  
Password: test



Ya loggeados, clickamos en database y seleccionamos la opción “Importar”. Ahí hacemos click sobre “Examinar” y seleccionamos database.sql, que se encontrará en el directorio donde hayamos clonado el repositorio.



Importamos la base de datos con el botón “Importar”.

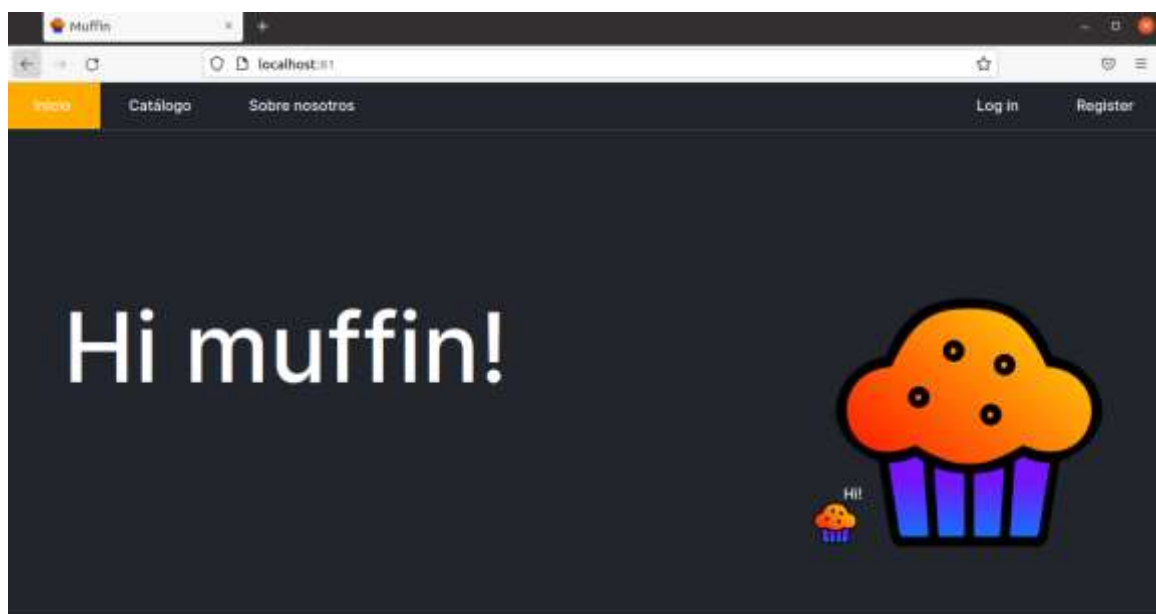


## WEB

Importada ya la base de datos, nos situamos con el navegador en la dirección del localhost:

<http://localhost:81>

¡Ya tenemos la página web!



En ella tenemos cinco pestañas:

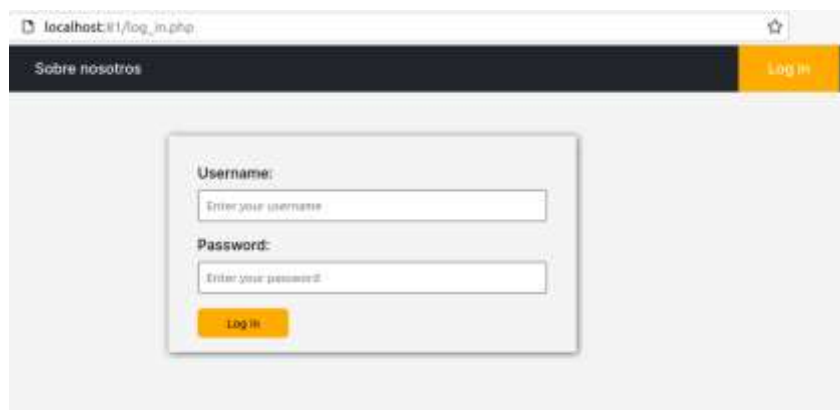
Inicio → página inicial dónde podremos ver una pequeña introducción a la temática de la web.

Sobre nosotros → descripción de los miembros del grupo que han realizado la web.

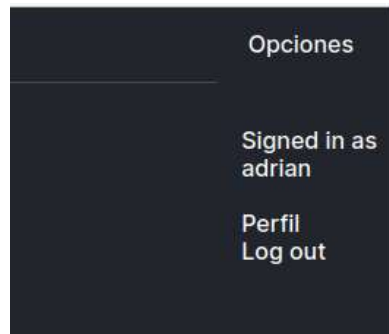


Log in → podemos identificarnos en cualquier momento si ya nos hemos registrado previamente o con los siguientes usuario de prueba:

Usuario	Password
Mikel001	test
Mikel002	test
Mikel003	test
Aitor001	test



Una vez identificado, tendremos una pestaña llamada “Opciones”, donde se nos indica con qué usuario estamos loggeados y la opción de poder cerrar la sesión mediante “Perfil Log Out”.



Register → también podemos registrarnos en el caso de que no queramos usar estos usuarios.  
Para ello tendremos un formulario que deberemos rellenar con nuestros datos.  
Guardaremos el usuario y contraseña porque nos servirá para logearnos posteriormente.

A registration form with the following fields: Username (with a hint 'Enter username'), Nombre y apellidos (with a hint 'Enter your name and surname'), DNI (with a hint 'Enter your DNI'), Teléfono (with a hint 'Enter your telephone number'), Email (with a hint 'Enter your email'), Fecha nacimiento (with a hint 'dd / mm / aaaa'), New password (with a hint 'Enter a new password'), and Repeat password (with a hint 'Repeat your password'). There is a 'Register' button at the bottom.

El

formato de algunos campos debe ser el siguiente:

Nombre y apellidos: sólo texto.

DNI: número de 8 cifras + Letra en mayúscula (sin guión en medio).  
Ejemplo: 00523821F

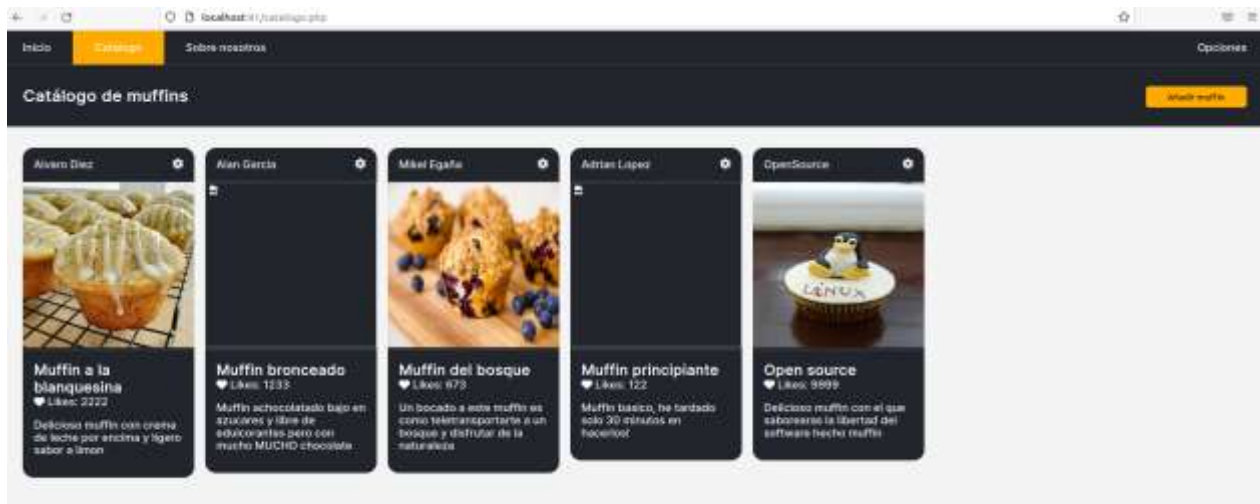
Teléfono: número de 9 dígitos.  
Ejemplo: 600000000

Fecha de nacimiento: aaaa/mm/dd , es decir, año, mes y día.  
Ejemplo: 2022/10/16

Email: ejemplo@servidor.extensión  
Ejemplo: muffin@gmail.com

Catálogo → tenemos una colección de muffins, con sus respectivos atributos.



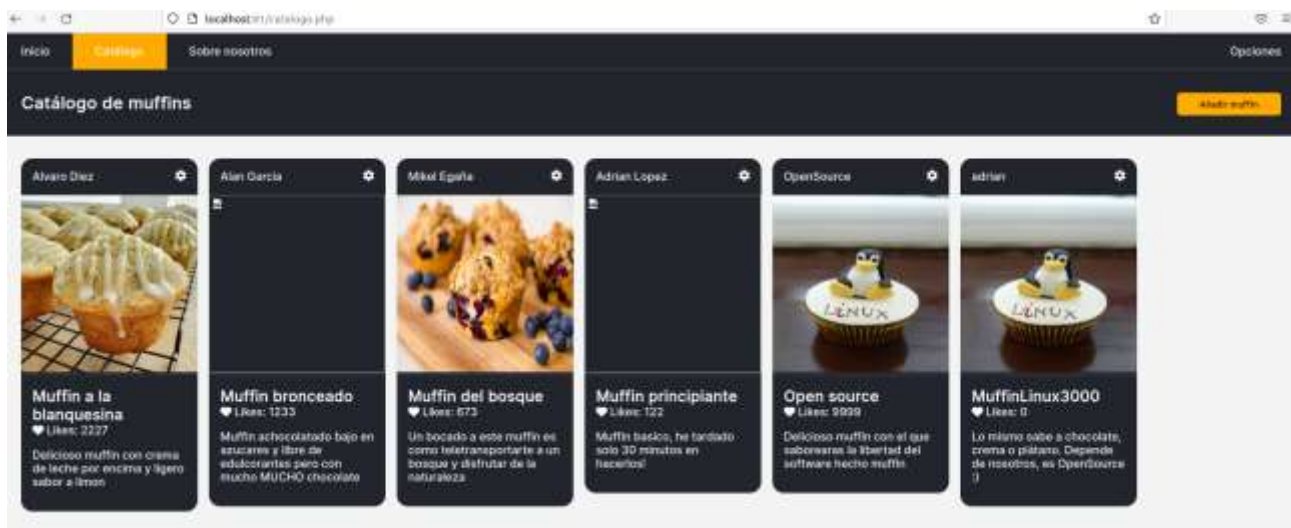


Cada muffin tiene cuatro atributos visibles:

- Dueño: será del usuario loggeado que añade el muffin o “Anónimo” si no estamos loggeados.
- Imagen
- Título: Muffin a la blanquesina, etc.
- Likes
- Descripción

En este catálogo, tenemos varias acciones para poder hacer:

- “Dar like” a un muffin:  
Para ello clickamos en el emoticono del corazón ♥  
Estos likes son acumulables, es decir, podemos dar los que queramos :)
- Añadir muffin:  
Usamos el botón “Añadir Muffin”  
Nos aparece un formulario en el que elegimos el título, la imagen (tenemos una lista de cinco imágenes distintas) y la descripción.



- Modificar o eliminar muffin:

Clickamos en el “engranaje” que se sitúa a la derecha del dueño del muffin  
 Nos aparece un formulario a rellenar en caso de  
 querer modificar los datos del muffin  
 Pulsamos el botón “Modificar datos”

Catálogo de muffins

Descripción:

Muffin de Mickey Mouse. ¡Ajá!

Título:

MuffinMouse

Usuario:

adrian

Likes:

69

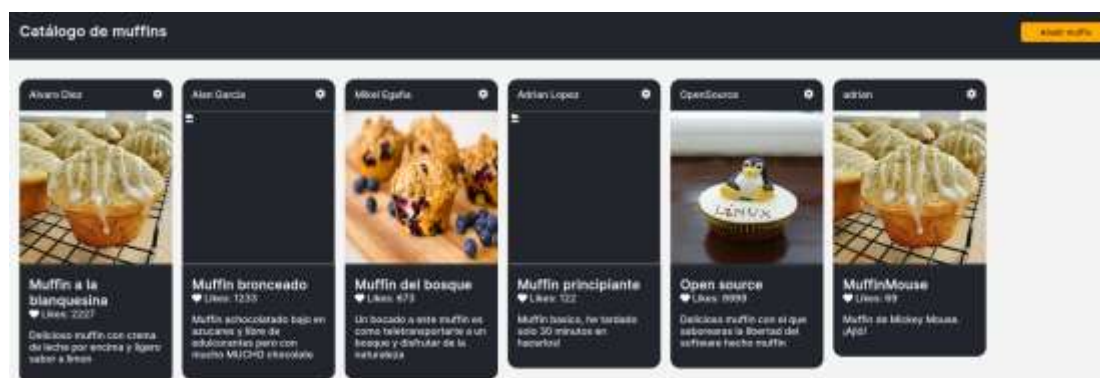
Tipo: ( Muffin con crema ▼ )

Id de tu muffin:

8

Modificar datos

Eliminar muffin





Si elegimos la opción de eliminar el muffin nos aparecerá un mensaje de confirmación



Catálogo de muffins

Nombre del muffin a eliminar:

MuffinMouse

Confirmar eliminación

Además, debajo del todo, tenemos una sección donde podrás ver nuestro repositorio de iconos y el enlace al repositorio de GitHub donde se encuentra nuestra proyecto.



**Comparte o modifica muffins a tu gusto**

Puedes tanto crear una nueva tarjeta como modificar las magdalenas ya existentes a tu gusto. ¡Este es un espacio para todos!

**Go Make An Awesome Muffin**  
Get yourself in and show off your best and latest pastry creations.  
mu.

**Links**  
[GitHub](#)  
[Repositorio de iconos](#)