

## Instrucciones para la ejecución del laboratorio

Primero investigamos que usar si una imagen de bitnami para contenedor en Docker u .ovs para Virtualbox

Por lo cual optamos por Docker

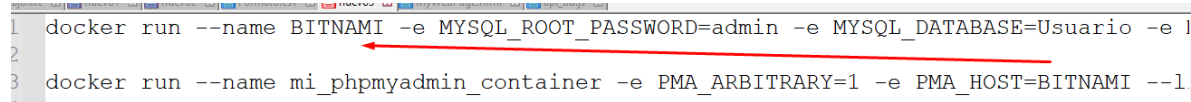
Según la investigación crearemos un contenedor previamente configurada con este comando

```
“docker run --name BITNAMI -e MYSQL_ROOT_PASSWORD=admin -e  
MYSQL_DATABASE=Usuario -e MYSQL_USER=admin1 -e MYSQL_PASSWORD=1234 -  
p 3306:3306 -d bitnami/mysql:latest”
```

Para poder acceder a estos servicios hicimos uso de un contenedor de phpmyadmin para ingresar a la base de datos


```
“docker run --name mi_phpmyadmin_container -e PMA_ARBITRARY=1 -e  
PMA_HOST=BITNAMI --link BITNAMI:db -p 8080:80 -d phpmyadmin/phpmyadmin”
```

Como podemos ver hace referencia al contenedor de BITNAMI.



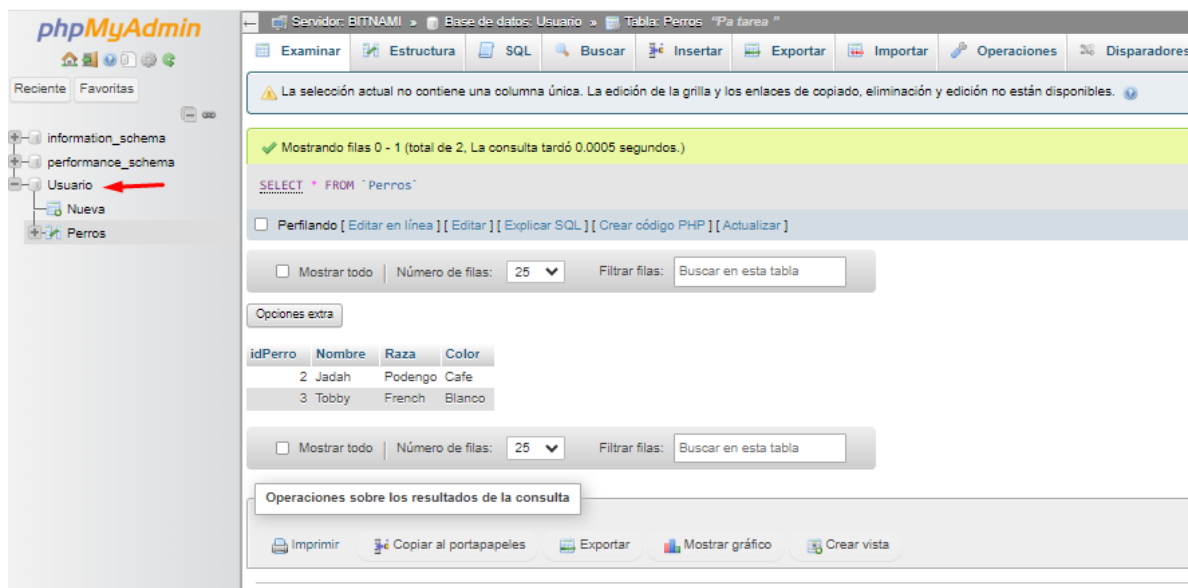
```
1 docker run --name BITNAMI -e MYSQL_ROOT_PASSWORD=admin -e MYSQL_DATABASE=Usuario -e l  
2  
3 docker run --name mi_phpmyadmin_container -e PMA_ARBITRARY=1 -e PMA_HOST=BITNAMI --l
```

Imágenes de acceso a la base



The image shows the phpMyAdmin login interface. At the top, there is a logo with a sailboat and the text "phpMyAdmin". Below it, the text "Bienvenido a phpMyAdmin" is displayed. The interface includes a language selection dropdown menu set to "Español - Spanish". Below this is a login section with the title "Iniciar sesión". It contains three input fields: "Servidor:" with the value "BITNAMI", "Usuario:" with the value "admin1", and "Contraseña:" with masked characters. A "Iniciar sesión" button is located at the bottom right of the login section.

Imagen de DB llamada Usuario con el script para Docker.



The image shows the phpMyAdmin interface displaying a database named "Usuario". The left sidebar shows the database structure with "Usuario" selected. The main area shows the "Perros" table with the following data:

idPerro	Nombre	Raza	Color
2	Jadah	Podengo	Cafe
3	Tobby	French	Blanco

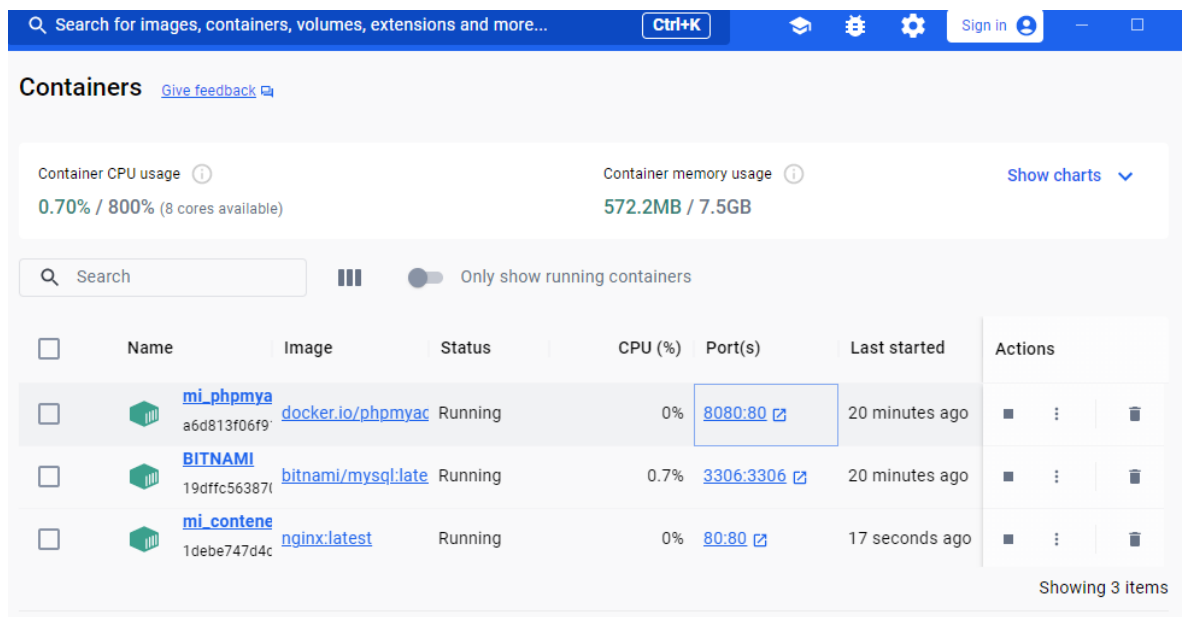
The interface also includes a SQL query editor with the query "SELECT \* FROM 'Perros'" and various operation buttons like "Examinar", "Estructura", "SQL", "Buscar", "Insertar", "Exportar", "Importar", "Operaciones", and "Disparadores".

Para el servicio web

Se creo contenedor con nginx

```
docker run --name mi_contenedor_nginx -p 80:80 -d nginx:latest
```

Imagen para vizualizar los 3 contenedores



Se realizó una pagina web con bootstrap y para consumir el servicio se realizo un .js con node.

Acá los códigos para levantar el servicio de node.js

```
npm init -y
```





```
npm install express
```

```
npm install mysql
```

```
node .
```

Imagen con archivos creados

▼ ayer

 package.json	1/25/2024 21:03	Archivo JSON	1 KB
 package-lock.json	1/25/2024 21:03	Archivo JSON	23 KB
 apibd.js	1/25/2024 20:40	JSFile	3 KB
 node_modules	1/25/2024 21:03	Carpeta de archivos	

## Imagen del front

XIMENA ELIZARDI - 1101720

JOSE DE LEON - 1170419

### Formulario de Perros

ID Perro:

Nombre:

Raza:

Color:

[Guardar](#) [Ver](#) [Eliminar](#) [Update](#)

## Imagen de datos en la base

Usuario

Nueva

Perros

SELECT \* FROM `Perros`

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [Actualizar](#) ]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:

Opciones extra

idPerro	Nombre	Raza	Color
2	Jadah	Podengo	Cafe
3	Tobby	French	Blanco

(el registro con id 1 fue borrado con las pruebas).