

1.- Captura con las entradas en la aplicación web vistas en dentro del contenedor mysql al hacer un:

```
select * from todo_items;
```

```
mysql> select * from todo_items;
+-----+-----+-----+
| id                | name                | completed |
+-----+-----+-----+
| f439ab5f-88fd-4f64-88f0-4c45f9a763a9 | Miguel Soler Base de Datos | 0        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

2.- captura donde se vea la salida de dos comandos:

2.1.- `docker ps` (para que se vean los dos contenedores)

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
fd3fac7d35a2	node:18-alpine	"docker-entrypoint.s..."	7 minutes ago	Up 7 minutes	127.0.0.1:3000->3000/tcp	miguelsolerpw
62eb3ae12c6a	mysql:8.0	"docker-entrypoint.s..."	16 minutes ago	Up 16 minutes	3306/tcp, 33060/tcp	miguelsolerdb

2.2.- `docker logs -f <id_contenedor_app>` para que se vea que usa la base de datos mysql

```
root@miguel:/getting-started-app# docker logs -f fd3fac7d35a2
yarn install v1.22.19
[1/4] Resolving packages...
success Already up-to-date.
Done in 0.83s.
yarn run v1.22.19
$ nodemon -L src/index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Waiting for mysql:3306.
Connected!
Connected to mysql db at host mysql
Listening on port 3000
```