**Informe de Implementación UARGFlow**



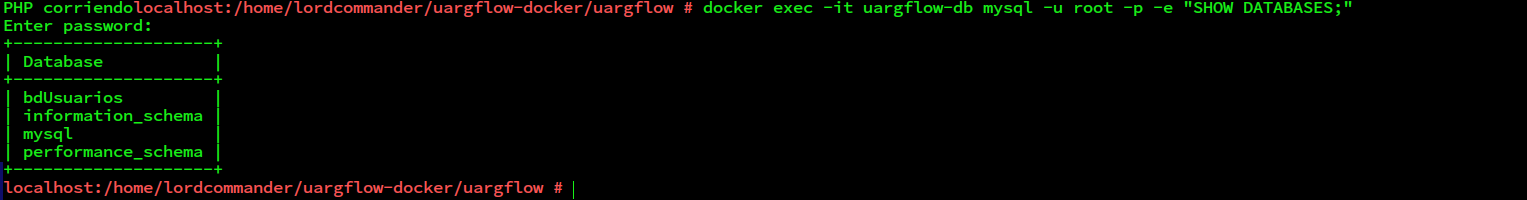
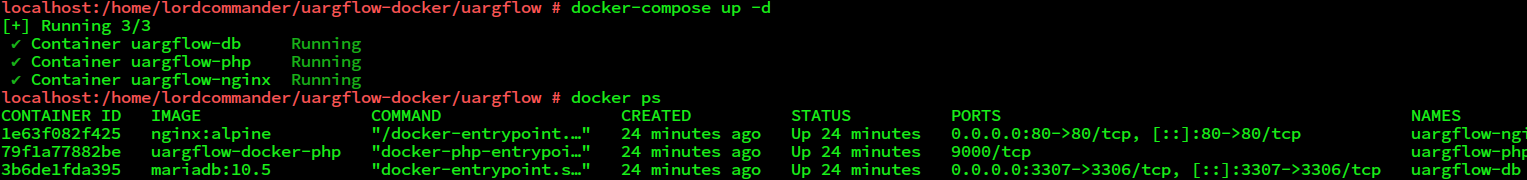
**Integrantes:** Nicolás Butterfield, Cristian Carranza, Martina Gagna, Abril Álvarez

### Objetivo

Explicar los pasos necesarios para lograr que el sistema UARGFlow funcione correctamente y luego aplicar un entorno local mediante Docker, replicando el comportamiento que se obtendría con XAMPP.

### Configuración de la Base de Datos

* Se levantaron los contenedores para PHP, Nginx y MariaDB mediante Docker Compose.
* Se configuró MariaDB en el contenedor con usuario root y contraseña, importando el script SQL que contiene las tablas y datos iniciales.
* La configuración de red y puertos permitió que PHP y Nginx accedan a la base de datos desde sus respectivos contenedores.



### Ajustes menores en el código

1. **Duplicación de JQuery en app/index.php:**

<!-- Esta importación de JQuery estaba repetida -->

<script type="text/javascript" src="../lib/JQuery/jquery-3.3.1.js"></script>

* Se identificó que la duplicación podía causar conflictos en la carga de scripts y comportamiento de la interfaz.

1. **Referencia en ControlAcceso.php:**

if (!$this->datos) {

// Se modifica la referencia a ObjetoDatos por BDConexion

throw new Exception(BDConexion::getInstancia()->errno, BDConexion::getInstancia()->error);

}

* Esto permitió que la función setPermisos funcione correctamente sin romper la lógica de permisos.

1. **Evitar redirecciones infinitas en el constructor de ControlAcceso:**

if ($this->ubicacion != Constantes::HOMEURL &&

$this->ubicacion != Constantes::SERVER . '/app/index.php') {

unset($\_SESSION["HTTP\_REFERER"]);

self::verificaLogin();

} else {

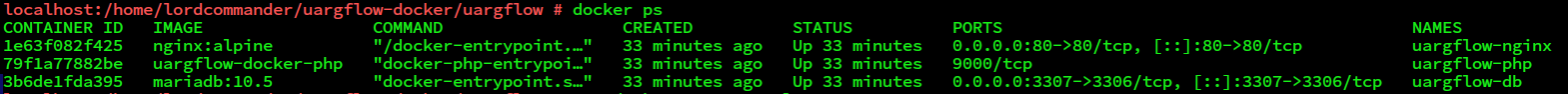
$\_SESSION["HTTP\_REFERER"] = Constantes::HOMEURL;

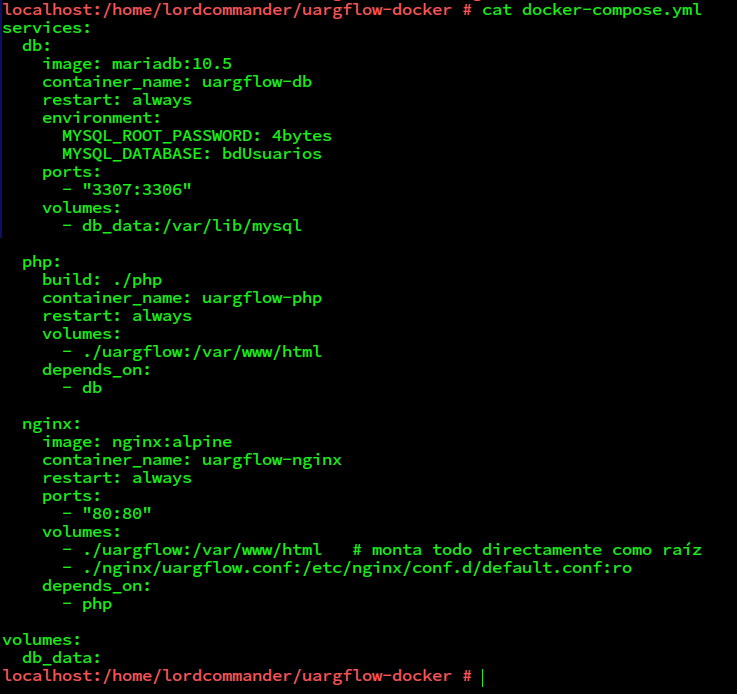
}

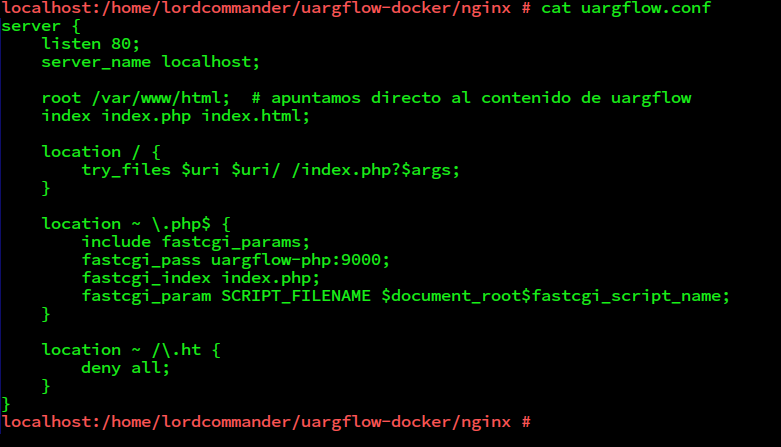
* Esto evitó que el login se redirija a sí mismo de forma indefinida.

### Montaje del sistema en Docker

* Se creó un Docker Compose con tres servicios: db (MariaDB), php (PHP5 + FPM) y nginx.
* Se configuró Nginx para que sirva el proyecto directamente desde la raíz /var/www/html del contenedor PHP.
* Se resolvieron los problemas de rutas relativas, eliminando la necesidad de incluir /uargflow en la URL.

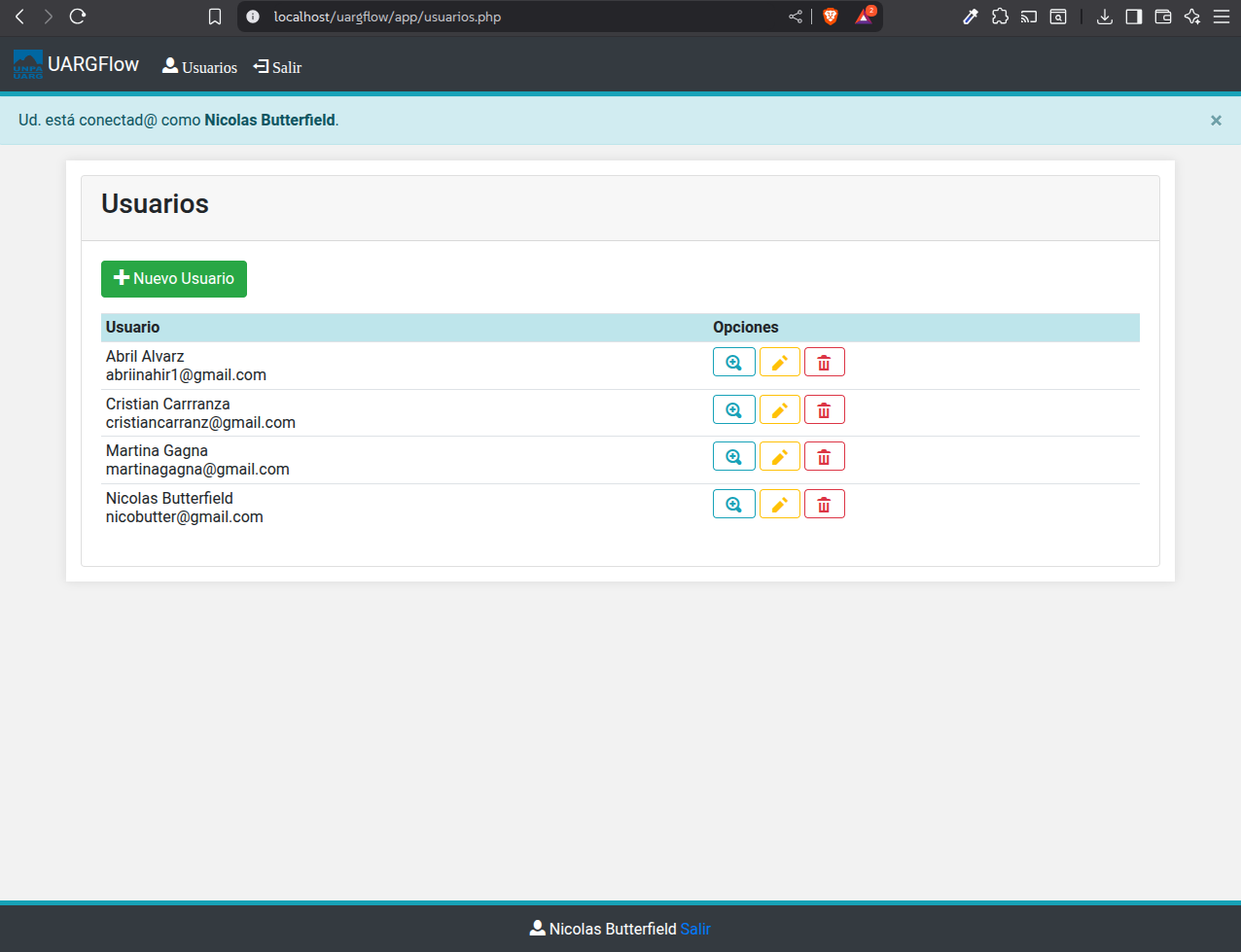






### Resultados

* El sistema arranca correctamente desde http://localhost/ sin necesidad de modificar rutas internas.
* La autenticación funciona como si se ejecutara en XAMPP.
* Se pueden acceder a todas las secciones del sistema, incluyendo Usuarios y otras funcionalidades.



### Conclusiones

* Se logró que el sistema UARGFlow funcione correctamente en un entorno Docker, replicando el comportamiento de XAMPP.
* Se realizaron ajustes mínimos en el código para garantizar compatibilidad con PHP5 y evitar problemas de redirecciones o duplicación de scripts.
* La experiencia demuestra que la configuración del entorno de ejecución es crucial para que sistemas antiguos funcionen correctamente en entornos modernos.