

🌱 Entorno Desarrollo PHP + MySQL + Composer en WSL (Ubuntu/Debian)

🎯 Objetivo

Configurar un entorno **completo de desarrollo web** dentro de **Windows Subsystem for Linux (WSL)**, con soporte para:

- Servidor Apache con `userdir`
- PHP y Composer
- MySQL
- Ejecución de scripts y respaldos automáticos con `cron`

Ideal para prácticas del curso **Administración de Sistemas Operativos (Tramo 8 - UTU)**.

⚙️ Requisitos previos

1. Tener instalado **WSL 2** con Ubuntu o Debian:

```
``powershell
```

```
wsl --install -d Ubuntu
```

Actualizar paquetes dentro de WSL:

```
bash
```

Copy code

```
sudo apt update && sudo apt upgrade -y
```

Instalar editor (opcional):

```
bash
```

Copy code

```
sudo apt install nano vim -y
```

📁 Estructura del proyecto

```
bash
```

Copy code

📁 php-lab-wsl/

├─ 📄 install_php_mysql_wsl.sh ← Script principal

├─ 📄 .env ← Configuración de entorno

└─ 📁 backups/ ← Carpeta de respaldos

📄 Archivo .env

Crea este archivo antes de ejecutar el script:

```
env
```

Copy code

```
USER_NAME=wsluser
```

```
USER_PASS=User123
```

```
WEB_ROOT=public_html
```

```
DB_NAME=php_lab
```

```
DB_USER=php_user
```

DB_PASS=Secret123

⚠ En WSL, el usuario suele ser el mismo que aparece al iniciar la terminal.
Puedes dejar USER_NAME como tu usuario actual (whoami).

🚀 Script install_php_mysql_wsl.sh

bash

Copy code

#!/bin/bash

set -e

=== Cargar configuración ===

if [! -f .env]; then

echo "❌ Archivo .env no encontrado."

exit 1

fi

export \$(grep -v '^#' .env | xargs)

echo "🔧 Configurando entorno PHP + MySQL + Composer en WSL para \$USER_NAME..."

=== Instalar dependencias ===

sudo apt update -y

sudo apt install -y apache2 php php-cli php-mysql composer unzip mysql-server

=== Habilitar módulos y userdir ===

sudo a2enmod userdir

sudo systemctl enable apache2

sudo service apache2 start

=== Crear carpeta web ===

USER_HOME=\$(eval echo ~\$USER_NAME)

WEB_PATH="\$USER_HOME/\$WEB_ROOT"

BACKUP_PATH="\$USER_HOME/backups"

mkdir -p "\$WEB_PATH" "\$BACKUP_PATH"

chmod 755 "\$USER_HOME"

sudo chown -R \$USER_NAME:\$USER_NAME "\$WEB_PATH" "\$BACKUP_PATH"

=== Configurar base de datos MySQL ===

echo "🗄 Configurando base de datos..."

sudo service mysql start

sudo mysql -u root <<MYSQL_SCRIPT

CREATE DATABASE IF NOT EXISTS \$DB_NAME;

CREATE USER IF NOT EXISTS '\$DB_USER'@'localhost' IDENTIFIED BY '\$DB_PASS';

GRANT ALL PRIVILEGES ON \$DB_NAME.* TO '\$DB_USER'@'localhost';

FLUSH PRIVILEGES;

MYSQL_SCRIPT

=== Crear index.php de prueba ===

```

cat <<EOF > "$WEB_PATH/index.php"
<?php
\ $mysqli = new mysqli("localhost", "$DB_USER", "$DB_PASS", "$DB_NAME");
if (\ $mysqli->connect_error) {
    die("❌ Error de conexión: " . \ $mysqli->connect_error);
}
echo "<h2>✅ Conexión a la base de datos exitosa (WSL)</h2>";
\ $result = \ $mysqli->query("SHOW DATABASES;");
echo "<h3>Bases de datos disponibles:</h3><ul>";
while(\ $row = \ $result->fetch_assoc()) {
    echo "<li>" . \ $row["Database"] . "</li>";
}
echo "</ul>";
\ $mysqli->close();
?>
EOF

```

```

sudo chown $USER_NAME:$USER_NAME "$WEB_PATH/index.php"

```

```

# === Configurar respaldo automático ===
BACKUP_SCRIPT="$BACKUP_PATH/db_backup.sh"
cat <<'EOF' > "$BACKUP_SCRIPT"
#!/bin/bash
source $(dirname $0)/../.env
DATE=$(date +%Y-%m-%d_%H-%M)
mysqldump -u root $DB_NAME > "$BACKUP_PATH/${DB_NAME}_$DATE.sql"
find "$BACKUP_PATH" -type f -mtime +7 -delete
EOF
chmod +x "$BACKUP_SCRIPT"

```

```

# === Cron para respaldos semanales ===
(crontab -l 2>/dev/null; echo "0 2 * * 0 bash $BACKUP_SCRIPT") | crontab -

```

```

echo "🕒 Cron configurado para respaldar todos los domingos a las 02:00 AM."

```

```

# === Mensaje final ===
WSL_IP=$(hostname -I | awk '{print $1}')
echo "✅ Instalación completada."
echo "🌐 Acceso web: http://$WSL_IP/~$USER_NAME"
echo "📁 Archivos web en: $WEB_PATH"
🔧 Prueba del entorno
Iniciá los servicios dentro de WSL:

```

```

bash
Copy code
sudo service apache2 start
sudo service mysql start
Verificá el acceso desde Windows:

```

Abre tu navegador y escribí:

arduino

Copy code

`http://localhost/~wsluser`

Si todo está correcto, deberías ver:

csharp

Copy code



Conexión a la base de datos exitosa (WSL)



Respaldo y mantenimiento

El script crea un respaldo semanal automático (domingo 2:00 AM) que se guarda en:

bash

Copy code

`~/backups/php_lab_YYYY-MM-DD_HH-MM.sql`

y limpia automáticamente los archivos de más de 7 días.

Para ejecutar el respaldo manualmente:

bash

Copy code

`bash ~/backups/db_backup.sh`



Competencias desarrolladas

Instalación de servicios en entornos Linux simulados (WSL).

Configuración de Apache, PHP y MySQL en entorno local.

Automatización de tareas administrativas con cron.

Respaldo de bases de datos y mantenimiento básico del sistema.

Comprensión de los permisos de usuario y estructura del sistema Linux.



Recomendaciones

Verificar que los servicios se inicien en WSL con permisos sudo.

Usar `ip addr show eth0` si localhost no responde, para conocer la IP interna de WSL.

Integrar este entorno con GitHub para control de versiones del `public_html`.