



# Certified Tech Developer

The Ultimate Degree

## Infraestructura I

# Objetivos

Repasar los términos vistos de módulos de PowerShell y ponerlos en práctica, descargando un módulo y corriendo comandos.

## Requisitos previos

MySQL es una base de datos común entre muchas organizaciones. Como tal, al crear un script de automatización que necesita consultar datos desde algún lugar, es posible que tenga la necesidad de consultar una base de datos MySQL.

Para consultar una base de datos MySQL, suponemos que ya tiene un servidor Windows MySQL y una base de datos con al menos una tabla que les gustaría recuperar. También vamos a suponer que no administra este servidor desde el servidor en sí, sino desde un software cliente, como MySQL Workbench, que le permite conectarse a su instancia de MySQL de forma remota.

Desafortunadamente, PowerShell no viene equipado con soporte para MySQL. Tenemos que descargar e instalar un software que se llama Connector/NET. Este es un controlador ADO al que PowerShell puede conectarse para administrar bases de datos MySQL. Si están trabajando en una máquina que ya tiene el software MySQL Workbench con todas las funciones, no tendrán que instalarlo.

Una vez que se haya instalado el controlador ADO, deberás descargar algún código de PowerShell que ya se haya creado para hacer esto. Utilizarán un módulo de PowerShell llamado MySQL. Este módulo consta de varios comandos que se pueden utilizar para administrar bases de datos MySQL.

## Construyendo el ambiente

### Opción 1:

Podés realizar el ejercicio en tu computadora, como requisito necesario para realizarlo es tener MySQL instalado.

### Opción 2:

Si no tenés instalado MySQL o MariaDB, podés ejecutar el motor de base de datos dentro de un contenedor, para esto deberás tener instalado Docker.

## Ejercicio

En el siguiente ejercicio trabajaremos con módulos de PowerShell.

Si no tenés instalado MySQL o MariaDB, podés ejecutar el motor de base de datos dentro de un contenedor:

```
docker run -d --rm -e MARIADB_ROOT_PASSWORD=Password123 --name testdb -p 3306:3306 mariadb:latest
```

Y luego ejecutar el siguiente comando para ingresar en el contexto del contenedor:

```
docker exec -it testdb bash
```

Una vez dentro del contenedor, ejecutar el siguiente comando para ingresar en MySQL:

```
mysql -p
```

El comando anterior nos va a solicitar ingresar con la contraseña del usuario root. En caso de que hayas decidido ejecutar la base de datos con un container de Docker, la

contraseña es la que colocamos en la variable de entorno MARIADB\_ROOT\_PASSWORD. En otras palabras "Password123" sin comillas. Luego deberás crear una base de datos en MySQL (o MariaDB) ejecutando estos comandos:

```
CREATE DATABASE test;
```

```
Use test;
```

```
CREATE TABLE test (field1 INT);
```

```
ALTER TABLE
```

```
test
```

```
CONVERT TO CHARACTER SET utf8mb4
```

```
COLLATE utf8mb4_unicode_ci;
```

```
INSERT INTO test (field1) VALUES (1);
```

En PowerShell vamos a buscar un módulo que se llama "invokequery":

```
find-module -name invokequery
```

Si lo encontramos, lo instalamos:

```
install-module -name invokequery
```

Puede que PowerShell te pregunte si querés confiar en el repositorio remoto, contestá que sí ingresando una "Y". A continuación vamos a utilizar el módulo recién instalado para ejecutar una query contra la base de datos que creamos en una instancia anterior. Para eso primero vamos a guardar las credenciales de MySQL en nuestro objeto PSCredential:

```
$creds = Get-Credential
```

En esta instancia, PowerShell te va a solicitar que ingreses las credenciales en un prompt. Si usaste Docker para iniciar la DB, el usuario es "root" (sin comillas) y la contraseña es la que colocamos en la variable de entorno MARIADB\_ROOT\_PASSWORD ("Password123" sin comillas). A continuación vamos a escribir la query SQL que vamos a ejecutar contra la DB y guardarla en una variable:

```
$query = New-SqlQuery -sql "select * from test"
```

Finalmente, vamos a ejecutar el comando contra la DB:

```
Invoke-MySqlQuery -SqlQuery $query -Credential $creds -Server localhost -Database  
test
```