



# **Presentación**

**Nombre:** Andy Manuel

**Apellidos:** Reyes Del Rosario

**Matricula:** 100488534

**Carrera:** Licenciatura en Informática

**Asignatura:** INF5170

**Sección:** 01

**Maestro:** Radhames Silverio González

# En PHP

El servidor MySQL soporta el uso de diferentes capas de transporte para conexiones. Las conexiones usan TCP/IP, sockets de dominio Unix o tuberías con nombre de Windows.

El nombre del host localhost tiene un significado especial. Está vinculado al uso de sockets de dominio Unix. No es posible abrir una conexión TCP/IP usando como nombre de host localhost, se debe usar 127.0.0.1 en su lugar.

## Ejemplo #1 Significado especial de localhost

```
<?php
$mysqli = new mysqli("localhost", "usuario", "contraseña", "basedatos");
if ($mysqli->connect_errno) {
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
}
echo $mysqli->host_info . "\n";

$mysqli = new mysqli("127.0.0.1", "usuario", "contraseña", "basedatos", 3306);
if ($mysqli->connect_errno) {
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
}

echo $mysqli->host_info . "\n";
?>
```

El resultado del ejemplo sería:

localhost via UNIX socket

127.0.0.1 via TCP/IP

*Valores predeterminados de los parámetros de conexión*

Dependiendo de la función de conexión usada se pueden omitir varios parámetros. Si no se proporciona un parámetro, la extensión intentará usar los valores predeterminados que están establecidos en el fichero de configuración de PHP.

## Ejemplo #2 Configuración predeterminada

mysqli.default\_host=192.168.2.27

mysqli.default\_user=root

mysqli.default\_pw=""

mysqli.default\_port=3306

mysqli.default\_socket=/tmp/mysql.sock

Los valores resultantes de los parámetros son pasados a la biblioteca cliente que esté usando esta extensión. Si la biblioteca cliente detecta parámetros vacíos o no establecidos, puede usar los valores internos predeterminados de la biblioteca.

#### *Valores predeterminados internos de conexión de la biblioteca*

Si el valor del host no está establecido o está vacío, la biblioteca cliente usará una conexión de socket Unix sobre localhost. Si el socket no está establecido o está vacío, y es solicitada una conexión de socket Unix, se intentará una conexión al socket predeterminado de /tmp/mysql.sock.

En sistemas Windows, el nombre de host . es interpretado por la biblioteca cliente como un intento de abrir una conexión basada en una tubería con nombre de Windows. En este caso el parámetro del socket se interpreta como el nombre de la tubería. Si no se proporciona o está vacío, se usará como socket (nombre de la tubería) \\.\pipe\MySQL.

Si no está establecida una conexión basada en un socket de dominio Unix ni en una tubería con nombre de Windows y el valor del parámetro del puerto no está establecido, la biblioteca usará como puerto predeterminado el 3306.

La biblioteca mysql nd y la Biblioteca Cliente de MySQL (libmysqlclient) implementan la misma lógica para determinados valores predeterminados.

#### **Opciones de conexión**

Las opciones de conexión están disponibles para, por ejemplo, establecer comandos iniciales que son ejecutados sobre la conexión, o para solicitar el uso de ciertos conjuntos de caracteres. Las opciones de conexión deben ser establecidas antes de que se establezca una conexión de red.

Para configurar una opción de conexión, la operación de conexión ha de ser realizada en tres pasos: crear un gestor de conexión con mysql\_init(), establecer las opciones solicitadas usando mysql\_options(), y establecer la conexión de red con mysql\_real\_connect().

## **Caché de conexiones**

La extensión mysqli soporta conexiones persistentes a bases de datos, las cuales son un tipo especial de conexiones almacenadas en caché. Por defecto, cada conexión a una base de datos abierta por un script es cerrada explícitamente por el usuario durante el tiempo de ejecución o liberada automáticamente al finalizar el script. Una conexión persistente no. En su lugar, se coloca en una caché para su reutilización posterior, si una conexión es abierta al mismo servidor usando el mismo nombre de usuario, contraseña, socket, puerto y base de datos predeterminada. La reutilización ahorra gastos de conexión.

Cada procesos de PHP utiliza su propia caché de conexiones mysqli. Dependiendo de modelo de distribución del servidor web, un proceso PHP puede servir una o múltiples peticiones. Por lo tanto, una conexión almacenada en caché puede ser utilizada posteriormente por uno o más scripts.

# Conexiones persistentes

Si una conexión persistente no usada con una combinación dada de host, nombre de usuario, contraseña, socket, puerto y base de datos predeterminada no se puede encontrar en la caché de conexiones, mysqli abrirá una nueva conexión. El uso de conexiones persistentes se puede habilitar y deshabilitar usando la directiva de PHP `mysqli.allow_persistent`. El número total de conexiones abiertas por un script puede ser limitado con `mysqli.max_links`. El número máximo de conexiones persistentes por proceso de PHP puede restringirse con `mysqli.max_persistent`. Observe que el servidor web puede engendrar muchos procesos de PHP.

Una queja común sobre las conexiones persistentes es que su estado no es reiniciado antes de su uso. Por ejemplo, las transacciones abiertas y no finalizadas no son automáticamente reanudadas. También, los cambios de autorización que ocurran durante la colocación de la conexión en la caché y su reutilización no están reflejados. Esto puede verse como un efecto secundario no deseado. Al contrario, el nombre persistente puede entenderse como una promesa de que el estado persiste.

La extensión `mysqli` soporta ambas interpretaciones de una conexión persistente: el estado persiste, y el estado se reinicia antes de la reutilización. Lo predeterminado es la reiniciación. Antes de que una conexión sea reutilizada, la extensión llama implícitamente a `mysqli_change_user()` para reiniciar el estado. La conexión persistente aparece al usuario como si estuviera recién abierta. No son visibles ningún artefacto de los usos previos.

La función `mysqli_change_user()` es una operación cara. Para un mejor rendimiento, los usuarios pueden recompilar la extensión con la bandera de compilación `MYSQLI_NO_CHANGE_USER_ON_PCONNECT` establecida.

Corresponde al usuario elegir entre comportamiento seguro o mejor rendimiento. Ambas son metas de optimización válidas. Para facilitar el uso, el comportamiento seguro es el predeterminado a expensas de un rendimiento máximo.