

Tema IX Ejercicio IV: Mantis Bug Tracker

Nicolás A. Ortega Froysa

15 de marzo de 2022

Índice

1. Introducción	3
2. Instalación	3
2.1. Dependencias	3
2.2. Instalación de MantisBT	5
3. Configuración	7
4. Creación y Manejo de Tareas	8
5. Conclusión	10
6. Derechos de Autor y Licencia	12

1. Introducción

Muchas veces cuando gestionamos cualquier tipo de proyecto, es necesario tener alguna herramienta que nos ayude a manejar las tareas (o incidencias) que vayan surgiendo al respecto. Esto facilita mucho tanto mantener de forma organizada nuestras incidencias, como también poderlas asignar y actualizar conforme a sus necesidades.

Una de estas herramientas es *Mantis Bug Tracker* (o *MantisBT*). Es una herramienta que se puede usar desde la *web*. Es *software libre*, y nos permite montar nuestra propia instancia de la misma en nuestro propio servidor.

En este documento revisaremos cómo instalar y configurar *MantisBT* y sus dependencias en un sistema GNU/Linux. Además revisaremos un uso básico, siendo cómo crear una tarea nueva y modificar su estado.

2. Instalación

2.1. Dependencias

Antes de empezar será necesario instalar y configurar primero las dependencias. *MantisBT* precisa de tres:

- PHP (versión 7 o mayor)
- MariaDB o MySQL (en nuestro caso usaremos MariaDB)
- Apache 2 (o cualquier servidor HTTP, como NGinx; mas en nuestro caso nos centraremos en Apache)

En nuestro caso, estamos instalando desde un sistema de Linux Mint, que usa el sistema de gestión de paquetes `apt` (figura 1a), como en Debian y Ubuntu. Por lo tanto, para instalar nuestras dependencias usaríamos el siguiente comando:

```
sudo apt install php php-cli php-fpm php-mysql \  
  php-zip php-gd php-mbstring php-curl php-xml \  
  php-pear php-bcmath apache2 mariadb-server \  
  mariadb-client
```

Una vez instaladas estas cosas, los servicios se activarán de forma automática (generalmente) en sistemas basadas en Debian, pero si este no es tu caso, puedes activarlas con los siguientes comandos:

```

sudo apt install php php-cli php-fpm php-mysql php-zip php-gd php-mbstring
File Edit View Search Terminal Help
nicolas ~/ $ sudo apt install php php-cli php-fpm php-mysql php-zip php-gd php-mbstring
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.9).
The following additional packages will be installed:
  galera-3 libapache2-mod-php7.4 libconfig-inifiles-perl libdbd-mysql-perl
  libdbi-perl libhtml-template-perl libionio5 libterm-readkey-perl
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common
  mariadb-server-10.3 mariadb-server-core-10.3 php-common php7.4 php7.4-bcmath
  php7.4-cli php7.4-common php7.4-curl php7.4-fpm php7.4-gd php7.4-json
  php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-readline php7.4-xml
  php7.4-zip
Suggested packages:
  libldb-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
  galera-3 libapache2-mod-php7.4 libconfig-inifiles-perl libdbd-mysql-perl
  libdbi-perl libhtml-template-perl libionio5 libterm-readkey-perl
  mariadb-client mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common
  mariadb-server mariadb-server-10.3 mariadb-server-core-10.3 php php-bcmath
  php-cli php-common php-curl php-fpm php-gd php-mbstring php-mysql php-pear

```

(a) Instalación de dependencias.

```

File Edit View Search Terminal Help
nicolas ~/ $ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.3.34-MariaDB-Debian0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> UPDATE mysql.user SET plugin = 'mysql_native_password' WHERE U
ser = 'root';
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
MariaDB [(none)]> QUIT;
nicolas ~/ $

```

(b) Configuración de usuario en base de datos.

```

File Edit View Search Terminal Help
nicolas ~/ $ sudo mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

```

(c) Instalación segura de MariaDB.

Figura 1: Instalación y configuración de dependencias.

```

sudo systemctl start apache2
sudo systemctl start mariadb

```

Una vez que se haya completado la instalación inicial, hay que configurar la base de datos (figura 1b). Para esto entramos en la base de datos y configuramos el usuario *root* de la forma siguiente para actualizar el *plugin* de autenticación:

```

mysql -u root
UPDATE mysql.user SET plugin = 'mysql_native_password'
  WHERE User = 'root';
FLUSH PRIVILEGES;
QUIT;

```

Una vez que hayamos acabado con esto, podemos finalizar la configuración de nuestra base de datos con el siguiente comando (figura 1c):

```
sudo mysql_secure_installation
```

Esto es un pequeño programa de configuración e instalación que lo usaremos para configurar la nueva contraseña de *root*. Cuando pregunta por la contraseña actual, que no tenemos, simplemente introduces una línea en blanco. Luego preguntará por la nueva contraseña. En todos los demás simplemente introducimos una línea en blanco para tomar la decisión por defecto.

Después de todo esto, tan sólo hemos de configurar el usuario que usará MantisBT, y crear la base de datos en sí. Esto se hace con los siguientes comandos:

```
CREATE USER 'mantisbt'@'localhost' IDENTIFIED BY
    'CatholicMantis';
CREATE DATABASE mantisbt;
GRANT ALL PRIVILEGES ON mantisbt.* TO 'mantisbt'@'localhost';
FLUSH PRIVILEGES;
QUIT;
```

Aquí se puede cambiar *CatholicMantis* por otra contraseña. El valor *mantisbt* es el nombre de nuestra base de datos. Nuestro usuario también se nombra *mantisbt*. Para comprobar si se ha configurado correctamente este usuario, se puede hacer entrando en el *shell* de MariaDB usando el comando siguiente:

```
mysql -u mantisbt -p
```

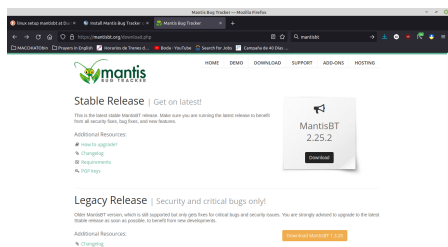
2.2. Instalación de MantisBT

Una vez configuradas nuestras dependencias, podemos instalar MantisBT en sí. Esto se hace primero descargando el archivo comprimido que contiene el código de la herramienta (figura 2a). Esto se puede hacer en la siguiente URL:

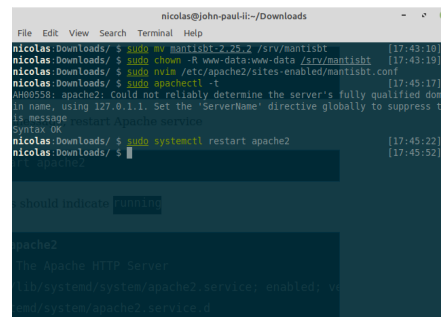
```
https://mantisbt.org/download.php
```

Una vez que hayamos descargado nuestro código hemos de descomprimirlo, moverlo a una ruta más accesible para el servicio de HTTP (i.e. Apache), configurar nuestro servidor para apuntar a esa ruta, y finalmente reiniciar el servicio HTTP (figura 2b). Esto se hace corriendo los siguientes comandos:

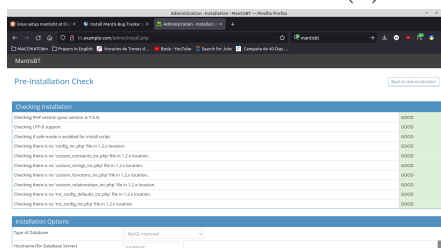
```
unzip mantisbt-{version}.zip
sudo mkdir -p /srv/
```



(a) Descarga.



(b) Instalación de MantisBT



(c) Primera vista de MantisBT.

Figura 2: Instalación de MantisBT

```

sudo mv mantisbt-{version}/ /srv/
sudo chmod -R www-data:www-data /srv/mantisbt

```

Con esto ya tendremos los archivos instalados, pero aún tenemos que configurar Apache para que apunte a la ruta correcta para leer nuestros archivos. Para esto, creamos un nuevo archivo:

```
/etc/apache2/sites-enabled/mantisbt.conf
```

E introducimos dentro de él lo siguiente:

```

<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot "/srv/mantisbt"
    ServerName bt.example.com
    ServerAlias www.bt.example.com
    ErrorLog "/var/log/apache2/mantisbt-error_log"
    TransferLog "/var/log/apache2/mantisbt-access_log"
    <Directory "/srv/mantisbt/">
        DirectoryIndex index.php index.html
        Options FollowSymLinks
    </Directory>
</VirtualHost>

```

```
        AllowOverride None
        Require all granted
        Options MultiViews FollowSymlinks
    </Directory>
</VirtualHost>
```

En este código, los variables importantes son los siguientes:

- `admin@example.com` es el correo electrónico del administrador de nuestro servicio.
- `/srv/mantisbt` es la ruta donde se encuentra MantisBT.
- `bt.example.com` es el nombre de dominio que tendrá (inclusive su *alias* en la línea siguiente).

Para verificar que el código lo hemos escrito (o copiado) correctamente, podemos probarlo corriendo el comando:

```
sudo apachectl -t
```

Si éste responde con la línea `Syntax OK`, entonces es que se ha escrito correctamente y Apache podrá leer sin errores lo que hemos escrito.

En nuestro caso, como hemos puesto un nombre de dominio que no es nuestro (i.e. `bt.example.com`), y que no enruta a nuestro servidor, podemos simularlo modificando el archivo `/etc/hosts` y añadiendo la línea siguiente:

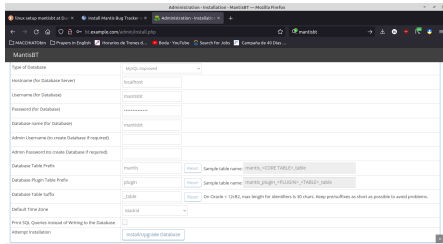
```
127.0.0.1    bt.example.com
```

Al hacer todo esto, podemos ir a nuestro navegador e introducir en la entrada de URL `http://bt.example.com` y nos debería salir ya MantisBT (figura 2c).

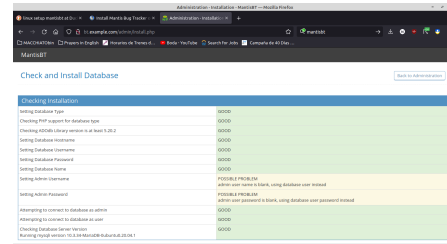
3. Configuración

Para configurar nuestro servicio de MantisBT, hemos de configurarlo de acuerdo a las opciones que habíamos configurado antes para nuestro base de datos con MariaDB (figura 3a). Principalmente nos interesa modificar los siguientes apartados:

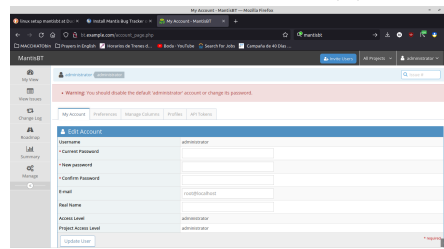
- **Username (for Database):** mantisbt
- **Password (for Database):** CatholicMantis



(a) Opciones de la base de datos.



(b) Resultado de configuración.



(c) Panel administrador.

Figura 3: Configuración de MantisBT.

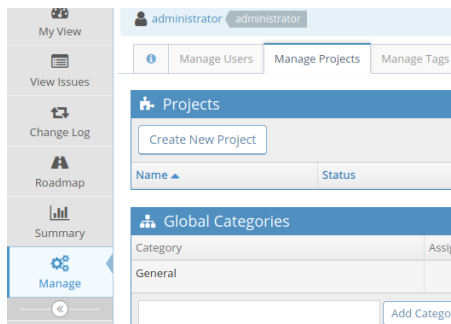
- **Database name (for Database):** mantisbt
- **Default Time Zone:** Madrid

Con todo esto configurado, le podemos dar ya al botón que dice *Install/Upgrade Database*. Esto empezará por poblar nuestra base de datos con tablas y entradas necesarias para MantisBT, y por lo tanto es algo que puede tardar unos minutos, aunque parezca que la conexión esté colgada.

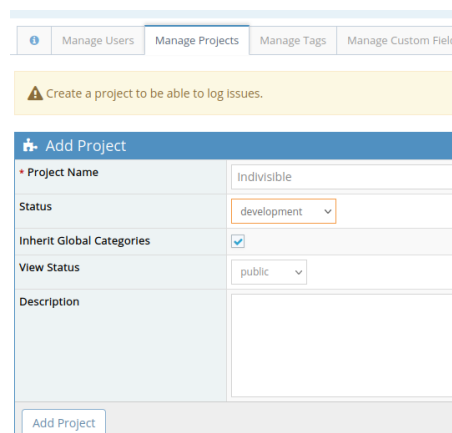
Cuando por fin termina, devolverá una nueva página con los resultados (figura 3b), donde no debería de haber ningún error. Si hubiese algún error es preciso buscar cómo resolverlo. Si vamos al cabo de la página encontraremos un botón para ir al panel del administrador (figura 3c). Aquí se nos pedirá entrar en sesión del administrador, que por defecto tiene usuario `administrator` y de contraseña `root`. Cuando entremos en sesión por primera vez, se nos pedirá cambiar esta contraseña.

4. Creación y Manejo de Tareas

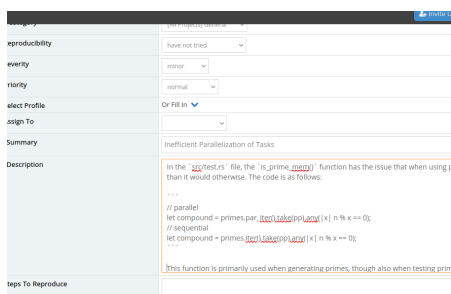
Para empezar a usar nuestra herramienta, lo más útil es empezar creando un proyecto. Esto lo podemos hacer entrando en *Manage* → *Manage Projects* → *Create New Project* (figura 4a). Esto nos llevará a una nueva pantalla donde podremos editar las propiedades del proyecto, aunque el único campo preciso es el de *Project Name* (figura 4b).



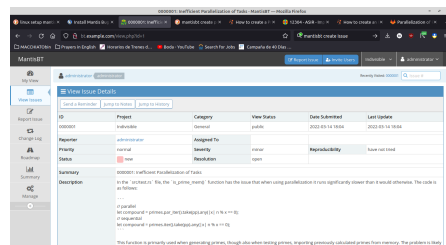
(a) Crear proyecto.



(b) Propiedades del proyecto.



(c) Crear una tarea nueva.



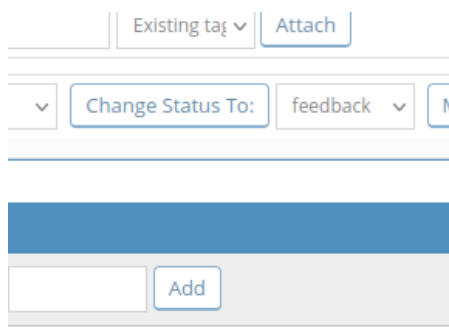
(d) Vista de tarea.

Figura 4: Creación de una tarea.

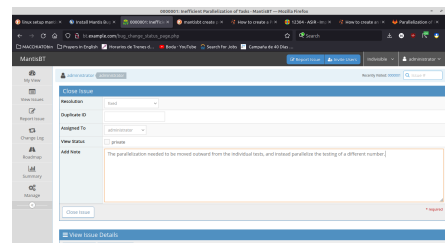
Para crear una tarea nueva, hemos de entrar en donde dice *Report Issue*. Esto nos llevará a una página que se parece a aquella donde configuramos las propiedades del proyecto, pero con más opciones (figura 4c). Introducimos la información pertinente que nos hace falta para nuestra tarea y la mandamos. Al finalizarla veremos un resumen de nuestra tarea con todos los estados actuales que tiene (figura 4d).

Los casos en sí tienen muchas opciones, pero la que más nos importa es su estado (i.e. la propiedad *Status*). Ésta la podemos modificar en lo bajo del reportaje donde pone *Change Status to:* (figura 5a). Nos provee una variedad de opciones:

- **New:** una tarea nueva que se acaba de crear. Se mantiene en este estado hasta que sea asignada o reconocida (i.e. *Acknowledged*).
- **Acknowledged:** se reconoce que la tarea es algo en que quieren trabajar los desarrolladores.



(a) Asignar estado.



(b) Cerrar caso.

ID	Project	Category	View Status	Date Submitted	Last Update
0000001	Indivisible	General	public	2022-03-14 18:04	2022-03-14 18:07
Reporter	administrator	Assigned To	administrator		
Priority	normal	Severity	minor	Reproducibility	have not tried
Status	closed	Resolution	fixed		
Summary	0000001: Inefficient Parallelization of Tasks				

(c) Caso cerrado.

Figura 5: Estados de un caso.

- **Confirmed:** se ha confirmado y reproducido la incidencia relacionada con la tarea en cuestión.
- **Assigned:** se le ha asignado la tarea a algún miembro para que trabaje sobre el caso como miembro dedicado.
- **Resolved:** cuando una tarea se resuelve, aunque no necesariamente que se haya arreglado la incidencia. Puede tomar varias formas (e.g. *fixed*, *won't fix*, *duplicate*, etc.).
- **Closed:** cuando se cierra definitivamente una tarea de tal modo que ya no se aceptan más aportaciones sobre el asunto.

Si intentamos cerrar el caso cambiando el estado a *Closed* nos traerá a una nueva página donde describimos por qué (figura 5b). Cuando finalicemos esto pulsando *Close Issue*, cerramos el caso y nos devolverá a la página de propiedades del caso con el caso ya cerrado (figura 5c).

5. Conclusión

La interfaz de MantisBT no es demasiado intuitiva, y se puede perderse uno fácilmente. Dicho lo cual, sí que parece una herramienta muy especializada para su tarea, comparada al menos con otras que vienen integradas dentro de repositorios públicos, como serían aquellas de GitHub o GitLab. A

la misma vez, tener tantas opciones también puede conducir a una sobrecarga de información, causando confusión en vez de ayudar a resolver problemas.

6. Derechos de Autor y Licencia

Copyright © 2022 Nicolás A. Ortega Froya <nicolas@ortegas.org>

Este documento se distribuye bajo los términos y condiciones de la licencia Creative Commons Attribution No Derivatives 4.0 International.