**Trabajo Final**

Vamos a utilizar UBUNTU 24.04 para el proyecto. Es una distribución de GNU/LINUX basada en Debian muy utilizada. Utiliza el sistema de paquetes apt y los mismos son administrados por Canonical, quienes luego de un proceso de pruebas extensivo y de pasar por los canales de Unstable, Testing Beta pasan a ser parte del estable que permiten que nuestro sistema tome del repositorio. Además, es una versión que tendrá un kernel LTS (Long Term Support) y evitará inconvenientes con actualizaciones de drivers de dispositivos y funciones nuevas en el sistema de archivos.

En el sistema de archivos utilizaremos BTRFS ya que nos permitirá utilizar snapshots y compresión persistente en todo el sistema de forma automática. Esto será perfecto, no solo para mantener el tamaño de las bases de datos comprimidas todo el tiempo, sino también para que en caso de algún inconveniente con quien administre el sistema, pueda volver rápidamente a una versión anterior del S.O.

El backup local, será simplemente un crontab que estará cargando la base de datos en un archivo externo. Como medio de respaldo remoto, utilizaremos RSYNC de ese mismo archivo en una red, conectada como NAS y en caso de que la institución lo considere necesario, podrá ser simplemente un USB remoto que servirá como espejo en caso de falla de algún disco mecánico o sólido.

**Instalación y puesta a punto**

Si bien, podría utilizarse una pequeña computadora de bajo consumo dedicada a este tipo de tareas como un Raspberry Pi, en este caso, utilizaremos una computadora X86/AMD-64 que se encuentra disponible por la institución que nos lo solicitaria.

Descargaremos una ISO de Ubuntu 24.04 de 64bits con el entorno de escritorio predeterminado GNOME y también la herramienta de creación de discos Ventoy.

El programa Ventoy simplemente permite seleccionar el pendrive a formatear para poder ejecutar las iso que uno necesite para hacer de BOOT. Una vez Ventoy formatee nuestro pendrive, colocaremos la ISO dentro de la partición exFAT creada.

Insertaremos el pendrive y reiniciaremos la PC a instalar el sistema operativo objetivo. Apretaremos las teclas F8 u F12 según corresponda y seleccionaremos nuestro pendrive dentro del selector de la BIOS/UEFI (puede que sea necesario desactivar SECURE BOOT cuando convive en sistemas antiguos).

Ahora comenzará la instalación de UBUNTU. La cual, dejaremos todas las opciones en predeterminado a excepción de cuando pida nuestra región que aclararemos Montevideo/URUGUAY (GMT-3) y la distribución de teclado en Español Latinoamericano.

Tras completar el proceso, el sistema habrá creado las particiones necesarias e instalará los paquete predeterminados de la distribución (Únicamente se hará algo distinto si es necesario que conviva con otro sistema operativo. Adicionalmente, el sistema de archivos predeterminado para Root y Home es EXT4 y es el que utilizaremos en el manual a diferencia del BTRFS sugerido).

Una vez iniciado el sistema, simplemente instalaremos por terminal los paquetes de GIT y MariaDB (la SQL) que nos permitirán descargar la aplicación y utilizarla

*sudo apt install git*

*sudo apt install mariadb-server*

Configuramos el SQL con su usuario y para que sea persistente en sistema con systemd

*sudo systemctl enable mariadb*

*sudo systemctl start mariadb*

*sudo mysql -u prestausuario -p prestacontrasena*

Tras instalarse, descargaremos el repositorio donde se encuentra nuestro programa. Lo haremos en la carpeta Documentos de la carpeta personal

*cd $HOME/Documentos*

*git clone https://github.com/rayonalamarante1/PrestamosSoft*

*cd PrestamosSoft*

Finalmente, daremos permisos de ejecución a todos los scripts ejecutables que utilizaremos.

*chmod +x abmlgestor.sh backup\_prestamos.sh operador\_prestamos.sh inicializar\_bd.sh*

**Operación de PrestamosSoft**

### ***Paso 1: Inicializar BD***

*Ejecutamos el script inicializar\_bd.sh que creará la base de datos del programa.*

### ***Paso 2: Script para respaldo manual***

### *Para realizar un backup, simplemente ejecutaremos el script backup\_prestamos.sh*

### ***Paso 2b Crear crontab para respaldos automáticos***

*Este cron, va a crear los respaldos cada medianoche ejecutando el script anterior.*

*crontab -e*

*# Añadir la siguiente línea:*

*0 0 \* \* \* /$HOME/Documentos/PrestamosSoft/backup\_prestamos.sh*

### ***Paso 2c Crear copia de seguridad en soporte remoto***

Insertaremos un pendrive y lo sincronizaremos con rsync.

Utilizaremos en la terminal el siguiente comando para ver los dispositivos montados

*df -h*

Y con la ruta conseguida, ejecutaremos la sincronización hasta que muestre el progreso completado.

*rsync -av --progress ~/Documentos/PrestamosSoft/ /media/$user/soporte\_remoto/PrestamosSoft*

### ***Paso 3: Operación principal de usuarios y grupos***

*Ejecuta el script principal:*

*./abml\_gestor.sh*

Este script es un script Bash que interactúa con una base de datos MySQL para gestionar usuarios y equipos de un sistema llamado "PrestamosElPinar". su funcionamiento por partes:

### 1. Variables de configuración

Configuramos las variables necesarias para conectarse a la base de datos MySQL: el nombre de la base de datos, el usuario y la contraseña.

### 2. Funciones para gestionar usuarios

-crear\_usuario: Esta función solicita información sobre un nuevo usuario (nombre, apellido, email, teléfono) e inserta estos datos en la tabla Usuarios de la base de datos.

-modificar\_usuario: Permite modificar los detalles de un usuario existente. Solicita el ID del usuario y el campo a modificar (nombre, apellido, email, teléfono). Si el usuario existe, se actualiza la base de datos.

-eliminar\_usuario: Permite eliminar un usuario. Primero verifica si el usuario existe. Luego pide confirmación para proceder con la eliminación del usuario.

-listar\_usuarios: Muestra todos los usuarios almacenados en la base de datos.

### 3. Funciones para gestionar equipos

-crear\_equipo: Solicita el nombre y la descripción de un equipo, y luego inserta estos datos en la tabla Equipos.

-modificar\_equipo: Permite modificar los detalles de un equipo. Solicita el ID del equipo y el campo a modificar (nombre, descripción). Si el equipo existe, se actualiza la base de datos.

-eliminar\_equipo: Permite eliminar un equipo, verificando si el equipo existe y pidiendo confirmación antes de proceder con la eliminación.

-listar\_equipos: Muestra todos los equipos almacenados en la base de datos.

### 4. Menú principal y ciclo de ejecución

El script tiene un menú interactivo donde el operador puede elegir qué acción ejecutar. El ciclo while true; do ... done mantiene el script en ejecución hasta que el usuario seleccione la opción de salir (opción 9).

-Mostrar menú: En cada iteración del ciclo, se presenta un menú con las opciones disponibles, como crear, modificar, eliminar usuarios o equipos, listar usuarios o equipos, y salir.

-Recepción de opción: Dependiendo de la opción que el usuario ingrese, se ejecuta la función correspondiente (crear, modificar, eliminar, listar).

### 5. Lógica de las consultas MySQL

En todas las funciones que interactúan con la base de datos, las consultas SQL se ejecutan a través del comando mysql -u $DB\_USER -p$DB\_PASS -e "...".

-Las consultas SQL se realizan dentro de bloques <<EOF que permiten ejecutar múltiples líneas de código SQL dentro de un solo comando.

-SELECT se utiliza para verificar la existencia de un usuario o equipo antes de realizar acciones como actualización o eliminación.

-INSERT INTO se usa para agregar nuevos usuarios o equipos a las tablas.

-UPDATE y DELETE se usan para modificar o eliminar registros existentes, respectivamente.

### Resumen

Este script facilita la gestión de usuarios y equipos en la base de datos "PrestamosElPinar". El operador interactúa con el sistema a través de un menú de opciones, y el script realiza las operaciones correspondientes sobre la base de datos MySQL. Es una herramienta útil para administradores que necesiten gestionar estos datos de forma rápida y sencilla desde la línea de comandos.

### ***Paso 4 : Operación principal de prestamos***

Ejecutamos ./operador\_prestamos.sh

Este script es una herramienta para gestionar préstamos dentro del sistema "PrestamosElPinar" que interactúa con una base de datos MySQL. A continuación, te explico cada parte del script para que un operador pueda entender cómo usarlo y qué hace.

### **2. Funciones principales**

#### **crear\_prestamo:**

Esta función permite crear un nuevo préstamo en el sistema. A continuación se detalla cómo funciona:

1. Solicitar información: El script solicita al operador los siguientes datos:
   * id\_usuario: El identificador del usuario que solicita el préstamo.
   * id\_equipo: El identificador del equipo que se va a prestar.
   * fecha\_prestamo: La fecha en la que se realiza el préstamo.
   * fecha\_devolucion: La fecha en la que el préstamo debe ser devuelto.
2. Insertar el préstamo en la base de datos: Después de obtener los datos, el script ejecuta una consulta SQL que inserta un nuevo préstamo en la tabla Prestamos de la base de datos. Los valores insertados son:
   * id\_usuario, id\_equipo, fecha\_prestamo, fecha\_devolucion: Los valores introducidos por el operador.
   * caducado: Inicialmente se establece en 0, lo que indica que el préstamo no ha caducado.
   * borrable: También se establece en 0, lo que indica que el préstamo no se puede eliminar automáticamente.
3. El comando SQL se ejecuta con mysql -u $DB\_USER -p$DB\_PASS, usando la base de datos configurada previamente.
4. Confirmación: Una vez ejecutada la consulta, el script imprime un mensaje indicando que el préstamo se ha creado exitosamente.

#### **listar\_prestamos:**

Esta función permite listar todos los préstamos registrados en la base de datos.

1. Ejecutar una consulta SQL: El script consulta la base de datos para obtener todos los préstamos, mostrando los campos:
   * id\_prestamo, id\_usuario, id\_equipo, fecha\_prestamo, fecha\_devolucion, caducado, borrable.
2. Esta consulta se ejecuta de nuevo usando el comando mysql con la configuración de conexión que se ha especificado al principio.
3. Mostrar resultados: Los resultados de la consulta se imprimen en la terminal, lo que permite al operador ver todos los préstamos registrados en el sistema.

### **3.** Menú principal y ciclo de ejecución

El script presenta un menú interactivo en la terminal donde el operador puede elegir entre diferentes opciones:

#### **mostrar\_menu:**

Este método muestra las opciones disponibles en el script:

**-** Crear un nuevo préstamo: Llama a la función crear\_prestamo para crear un nuevo préstamo.

**-** Listar préstamos: Llama a la función listar\_prestamos para listar todos los préstamos.

**-** Salir: Termina la ejecución del script.

#### Ciclo de ejecución:

El ciclo while true; do ... done mantiene el script ejecutándose de manera continua hasta que el operador elige la opción de salir (opción 3). Cada vez que el operador selecciona una opción, el script ejecuta la función correspondiente. Si el operador elige una opción no válida, el script imprimirá un mensaje de error y volverá a mostrar el menú.

### 4. Interacción con la base de datos

En las funciones que interactúan con la base de datos (crear\_prestamo y listar\_prestamos), se utilizan consultas SQL para:

-Insertar nuevos registros (INSERT INTO) para los préstamos.

-Consultar los registros existentes (SELECT) para listar los préstamos.

### Resumen

Este

script proporciona una interfaz simple para gestionar préstamos en el sistema. Los operadores pueden:

Crear nuevos préstamos proporcionando detalles sobre el usuario, el equipo y las fechas. Listar todos los préstamos existentes en el sistema.

El script permite interactuar fácilmente con la base de datos, asegurando que los préstamos sean gestionados adecuadamente.

Historial de Terminal

1 sudo apt install git

2 sudo apt install mariadb-server

3 sudo systemctl enable mariadb SYSTEMD

4 sudo systemctl start mariadb

5 sudo mysql -u prestausuario -pprestacontrasena

sudo mysql -u root

SELECT User, Host FROM mysql.user;

CREATE USER 'prestausuario'@'localhost' IDENTIFIED BY 'prestacontrasena';

GRANT ALL PRIVILEGES ON \*.\* TO 'prestausuario'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

**6. Verificamos que los usuarios estén correctos dentro de la SQL:**

CREATE USER 'prestausuario'@'localhost' IDENTIFIED BY 'prestacontrasena';

GRANT ALL PRIVILEGES ON \*.\* TO 'prestausuario'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

**7. Verificamos que la contraseña esté correcta dentro de la SQL:**

ALTER USER 'prestausuario'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'prestacontrasena';

FLUSH PRIVILEGES;

**8. Inicializamos el programa:**

cd Documentos

git clone <https://github.com/gabrielamarante1/PrestamosSoft>

cd PrestamosSoft

Damos Permisos de ejecución:

chmod +x abml\_gestor.sh backup\_prestamos.sh operador\_prestamos.sh inicializar\_bd.sh

Inicializamos la base de datos: ./inicializar\_bd.sh

Corremos cada script ./abml\_gestor.sh