

Universidad Peruana de Ciencias Aplicadas



(1ASI0726)

SISTEMAS OPERATIVOS

Práctica Calificada 1 (PC1)

Informe

Carrera de Ingeniería de Software

NRC: 7216

Alumnos:	
Código	Nombres y apellidos
U20241C134	Tumi Oliden, Manuel Ignacio
U20221C726	Hanco Poma, Keyner Ivan
U202022211	Javier Murillo, Mathias

2025-2

1. Introducción

El mercado de electrodomésticos en el Perú ha experimentado un crecimiento impulsado por la digitalización de los canales de venta y el cambio en los hábitos de consumo. Empresas como Oechsle han expandido sus operaciones hacia plataformas de comercio electrónico, lo que les permite atender una base de clientes más amplia y mejorar la eficiencia de sus procesos logísticos y comerciales.

Sin embargo, muchas organizaciones del sector aún enfrentan limitaciones relacionadas con la gestión de inventarios, el manejo de grandes volúmenes de transacciones en línea y la necesidad de garantizar la seguridad en el procesamiento de pagos.

Este proyecto busca plantear y elaborar una arquitectura de E-Commerce en la nube, considerando que la marca “ElectroHogar” es un negocio con años de experiencia y con un amplio inventario de electrodomésticos que busca digitalizar su negocio para poder superar un estancamiento de ventas y crecer el valor y exposición de su marca según la demanda del mercado actual. La solución plantea el uso de tecnologías modernas y buenas prácticas para el desarrollo de Software, por ende, el sistema será desarrollado en una plataforma de nube pública (Microsoft Azure) con una base de datos relacional (MySQL) en una máquina virtual con sistema operativo Linux Ubuntu.

2. Descripción del Problema: Caso del negocio

La empresa minorista enfrenta los siguientes desafíos:

- Dependencia de tiendas físicas: aunque existe presencia digital, la mayor parte de las ventas aún ocurre de manera presencial.
- Limitaciones en la disponibilidad del servicio: no se cuenta con un canal de atención disponible 24/7.
- Gestión manual y descentralizada de inventarios: ocasiona errores en stock y retrasos en la entrega de productos.
- Altos costos de operación por falta de integración: los sistemas actuales no están unificados, lo que provoca duplicidad de esfuerzos y pérdidas de eficiencia.
- Seguridad insuficiente en transacciones en línea: lo que genera desconfianza en los clientes y limita el crecimiento del canal digital.

Frente a estas problemáticas, la empresa busca implementar ElectroHogar, una plataforma en la nube que optimice la gestión de productos electrodomésticos y brinde una experiencia confiable y moderna a los clientes.

3. Objetivo del trabajo

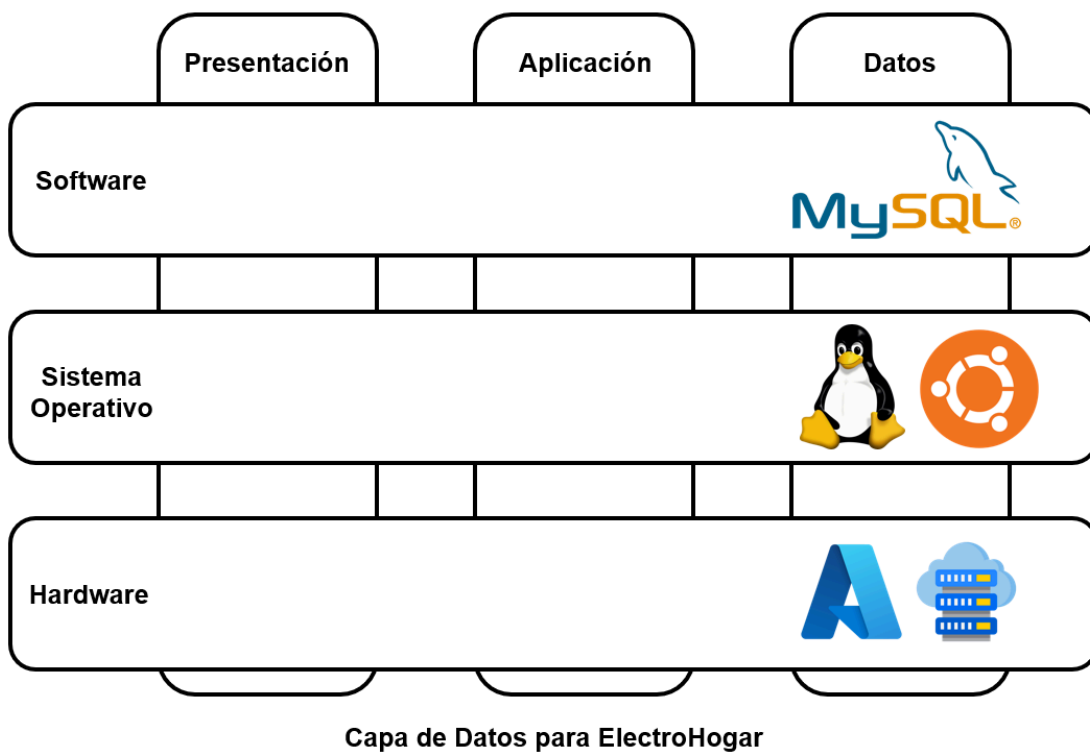
Objetivo General:

Diseñar e implementar la arquitectura de ElectroHogar, un sistema E-Commerce en la nube que permita a la empresa gestionar de manera eficiente el catálogo de electrodomésticos, optimizar la atención al cliente y garantizar la seguridad de las transacciones en línea.

Objetivos Específicos:

- Crear una arquitectura escalable de tres capas (presentación, aplicación y datos).
- Definir los requisitos técnicos de hardware, software y comunicaciones necesarios para soportar picos de demanda.
- Incorporar medidas de seguridad para proteger la información sensible de clientes y transacciones.
- Establecer mecanismos de integración con servicios de pago, logística y notificaciones.
- Presentar un diagrama de la arquitectura propuesta.

4. Análisis técnico de los S.O. a usar (por capas)



4.1 Requisitos Técnicos

Usuarios

- Concurrentes: 400 usuarios.
- Potenciales: 5000 usuarios.

Tamaño de transacción

- Catálogo de productos e imágenes: 250 KB por transacción.
- Datos de carrito y cliente: 50 KB.
- Información de pago y logística: 30 KB.
- **Total promedio:** 330 KB por transacción.

Requerimientos de comunicación (ancho de banda)

- Ancho de banda mínimo: 80 Mbps dedicado.
- Considera 300 usuarios concurrentes con un factor de simultaneidad del 25%, y overhead por cifrado y protocolos de seguridad.

Tabla 1: Comparativa de motores de base de datos

Características	MySQL	SQL Server 2022	Firebird SQL	Oracle	PostgreSQL
					

Tipo de licencia	Código abierto (GPL) y versiones comerciales disponibles (Oracle)	Comercial (con versión gratuita)	Código abierto	Comercial (propietaria)	Código abierto
Lenguaje de consulta	SQL estándar	T-SQL	SQL estándar	PL/SQL	SQL estándar + (PL/pgSQL)
Soporte de transacciones	Sí (ACID, con motores como InnoDB)	Sí (ACID)	Sí (ACID)	Sí (ACID)	Sí (ACID)
Escalabilidad	Alta	Muy alta	Media	Muy alta	Alta
Flexibilidad	Alta	Alta	Media	Alta	Alta
Herramienta de administración	MySQL Workbench / phpMyAdmin / DBeaver	SQL Server Management Studio (SSMS)	FlameRobin	Oracle SQL Developer	pgAdmin, DBeaver, psql (terminal)
Costos	Gratuito	Gratuito/Pago	Gratuito	Costoso	Gratuito




4.2 Comparativa de Sistemas Operativos

Características	Linux (Debian 12)	Linux (Ubuntu 24.04)	Fedora 40	Windows Server (v.2019) Standard
				
Rendimiento	Muy alto, ideal para producción crítica	Alto, optimizado para hardware moderno	Alto, optimizado para hardware moderno	Alto, especialmente con soluciones Microsoft
Compatibilidad con aplicaciones	Amplia compatibilidad con software de código abierto y servidores.	Buena compatibilidad, tiene soporte para contenedores y servicios en la nube.	Buena compatibilidad, pero especialmente para software moderno.	Alta compatibilidad con software de Microsoft.
Facilidad de administración	Requiere conocimientos técnicos.	Interfaz para cualquier persona y de fácil administración.	Requiere de familiarización con nuevas tecnologías, pero tiene herramientas modernas.	Interfaz gráfica muy intuitiva y múltiples herramientas.

Compatibilidad con hardware	Muy bueno, especialmente con hardware antiguo.	Amplia compatibilidad con hardware moderno.	Soporte para hardware nuevo, pero no mucho con el antiguo.	Compatibilidad sólida con hardware certificado.
Actualizaciones y parches	Ciclo de actualizaciones moderados; prioriza la estabilidad.	Actualizaciones regulares, pero con soporte a largo periodo.	Actualizaciones frecuentes, pero pueden introducir inestabilidades.	Actualizaciones periódicas, requiere licencias para recibir parches.
Costo y lineamientos	Gratuito y de código abierto, sin costos de licencia.	Gratuito para uso general.	Gratuito y de código abierto.	Requiere licencias pagadas.
Comunidad y soporte	Comunidad activa y mucha documentación.	Comunidad activa y abundante documentación.	Comunidad activa enfocada en desarrolladores.	Soporte profesional de Microsoft, comunidad activa en foros.
Escalabilidad	Muy escalable, adecuado para servidores de cualquier tamaño.	Buena escalabilidad, dirigido a entornos de nube y servidores empresariales.	Escalable, con enfoque en tecnologías nuevas.	Escalabilidad sólida en entornos empresariales.

Seguridad	Alta seguridad enfocado en estabilidad y control de paquetes.	Buena seguridad con actualizaciones regulares y soporte.	Buena seguridad, pero algunas actualizaciones requieren atención.	Seguridad empresarial con herramientas integradas, pero requiere configuración adicional.
------------------	--	---	--	--

Tabla 6. Comparativa de plataformas en la nube

Características	Microsoft Azure	Google Cloud Platform	Amazon web (AWS)
			
Interfaz de gestión	Intuitiva y profesional	Moderna, centrada en Big Data	Completa pero más compleja
Despliegue de máquinas virtuales	Sencillo y con plantillas	Rápido, con opciones preconfiguradas	Versátil, muchas AMIs disponibles
Compatibilidad	Alta con tecnologías Microsoft y Linux	Alta con contenedores y Kubernetes	Amplia, incluso con arquitecturas híbridas
Costo inicial	Gratuito por 12 meses con servicios limitados; luego pago por uso	Créditos iniciales, pero precios elevados a largo plazo	Gratuito por 1 año (uso limitado), luego alto

Seguridad y cumplimiento	Estándares empresariales, integración con Active Directory	Alta, certificaciones múltiples	Alta, con configuraciones avanzadas
Escalabilidad	Muy buena, servicios integrados (App Services, DBaaS)	Excelente, autoscaling automático	Muy buena, con soporte a microservicios
Comunidad y soporte	Soporte oficial, comunidad técnica	Soporte oficial, comunidad activa	Comunidad masiva, soporte empresarial
Integración CI/CD	Muy buena con GitHub Actions y DevOps	Muy buena con Cloud Build	Alta con CodePipeline y terceros

4.4 Configuración Propuesta

Sistema Operativo: Ubuntu 22.04 LTS (Linux).

Motor de Base de Datos: MySQL 8 (con soporte ACID e integridad transaccional).

Cliente de Conexión: MySQL Workbench / DBeaver / phpMyAdmin.

Software de Aplicación: Apache 2.4 + PHP 8.3 (con opción de usar Laravel como framework MVC).

Software de Presentación: ReactJS, HTML5, CSS3, JavaScript.

Hardware (recomendado para entorno académico):

- **Datos (Base de Datos):** 2 vCPU, 4 GB RAM, 50 GB SSD.
- **Aplicación (Backend):** 2 vCPU, 4 GB RAM, 50 GB SSD.

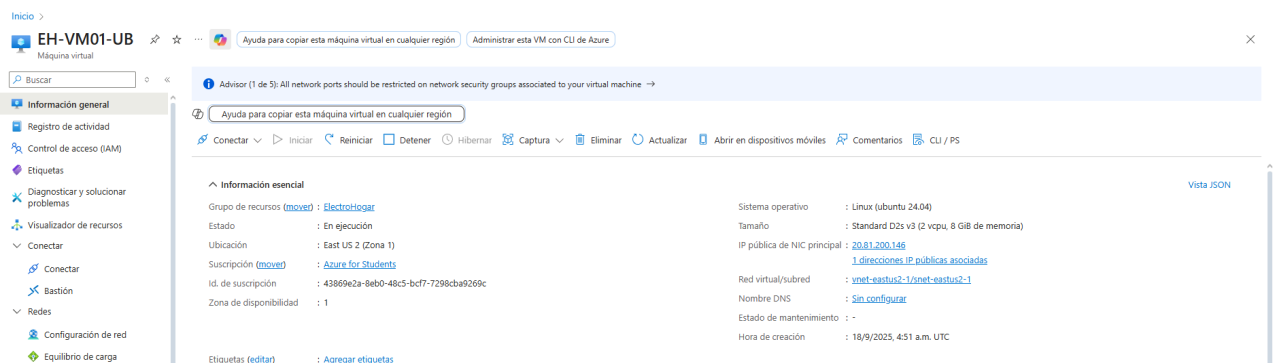
- **Presentación (Frontend):** 1 vCPU, 2 GB RAM, 20 GB SSD.

Infraestructura: Implementación en **Microsoft Azure** mediante máquina virtual con Ubuntu Server, con posibilidad de integrar servicios básicos de escalabilidad y balanceo de carga según requerimientos.

5. Evidencias de Servicios configurados

Creación de la máquina virtual de Linux Ubuntu en Azure:

En la imagen se observa cómo el apartado “Información General” nos muestra las especificaciones de la máquina virtual, podemos apreciar el sistema operativo, estado, tamaño y dirección IP, también desde este apartado se puede configurar la VM según los requerimientos.



Para acceder a la VM a través del CLI o GitBash:

Asignación de permisos de lectura a la llave: `chmod 400 <nombre_de_la_llave.pem>`

Conexión vía SSH: `ssh -i <nombre_de_la_llave.pem>
<username>@<IP_publica>`

Se emplea la llave .pem como método de acceso, debido a la capa de seguridad adicional que ofrece frente a ataques cibernéticos, y validamos el acceso con las credenciales de la VM (username e IP pública)

MINGW64:/c/Users/User/Documents/ElectroHogar

```
User@DESKTOP-KTA4U00 MINGW64 ~  
$ cd Documents/ElectroHogar  
  
User@DESKTOP-KTA4U00 MINGW64 ~/Documents/ElectroHogar  
$ chmod 400 ElectroHogarKey.pem  
  
User@DESKTOP-KTA4U00 MINGW64 ~/Documents/ElectroHogar  
$ ssh -i ElectroHogarKey.pem ElectroHogarAdmin@20.81.200.146
```

Sistema operativo
Tamaño
IP pública de NIC p
ual/subred
Nombre DNS
Estado de mantenir

ElectroHogarAdmin@EH-VM01-UB: ~

```
$ chmod 400 ElectroHogarKey.pem  
  
User@DESKTOP-KTA4U00 MINGW64 ~/Documents/ElectroHogar  
$ ssh -i ElectroHogarKey.pem ElectroHogarAdmin@20.81.200.146  
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.11.0-1018-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of Thu Sep 18 17:03:19 UTC 2025  
  
System load:  0.02          Processes:            137  
Usage of /:   3.8% of 60.95GB Users logged in:      0  
Memory usage: 8%           IPv4 address for eth0: 172.17.0.4  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status
```

Instalación del DBMS (MySQL):

Para instalar MySQL dentro de la VM debemos ejecutar el siguiente comando: `sudo apt install mysql-server`

Para verificar el estado de MySQL, utilizamos el comando: `sudo service mysql status`

```
ElectroHogarAdmin@EH-VM01-UB: ~
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-09-18 16:58:33 UTC; 6min ago
     Process: 1944 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0>
   Main PID: 1953 (mysqld)
    Status: "Server is operational"
     Tasks: 37 (limit: 9449)
    Memory: 382.6M (peak: 384.6M)
       CPU: 2.655s
    CGroup: /system.slice/mysql.service
           └─1953 /usr/sbin/mysqld

Sep 18 16:58:32 EH-VM01-UB systemd[1]: Starting mysql.service - MySQL Community Server...
Sep 18 16:58:33 EH-VM01-UB systemd[1]: Started mysql.service - MySQL Community Server.
~
~
~
~
```

Para verificar los puertos en uso:

A través del comando `netstat -tulpn` mostramos los puertos que se encuentran siendo usados por los servicios del sistema.

Localizamos el servicio asociado al puerto 3306 (que corresponde a MySQL), esto significa que MySQL está preparado para aceptar conexiones a través de este puerto.

```
ElectroHogarAdmin@EH-VM01-UB: ~  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ElectroHogarAdmin@EH-VM01-UB:~$ netstat -tulpn  
(No info could be read for "-p": geteuid()=1000 but you should be root.)  
Active Internet connections (only servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
PID/Program name  
tcp        0      0 127.0.0.53:53          0.0.0.0:*                LISTEN  
tcp        0      0 127.0.0.1:3306         0.0.0.0:*                LISTEN  
tcp        0      0 127.0.0.1:33060        0.0.0.0:*                LISTEN  
tcp        0      0 127.0.0.54:53          0.0.0.0:*                LISTEN  
tcp        0      0 0.0.0.0:22             0.0.0.0:*                LISTEN  
tcp6       0      0 :::22                  :::*                    LISTEN  
udp        0      0 127.0.0.54:53          0.0.0.0:*                  
udp        0      0 127.0.0.53:53          0.0.0.0:*                  
udp        0      0 172.17.0.4:68          0.0.0.0:*                  
udp        0      0 127.0.0.1:323          0.0.0.0:*                  
udp6       0      0 :::1:323               :::*                      
ElectroHogarAdmin@EH-VM01-UB:~$ |
```

Configuración de puertos de entrada en Azure:

Antes de establecer conexión con MySQL, debemos comprobar que el puerto 3306 está activo. Para lograr esto, debemos acceder a la sección de “Configuración de Red” en Azure y realizamos la creación del puerto “Permitir-MySQL”



Permitir-MySQL

EHVM01UBnsg138



Origen ⓘ

Any



Intervalos de puertos de origen * ⓘ

*

Destino ⓘ

Any



Servicio ⓘ

MySQL



Intervalos de puertos de destino ⓘ

3306

Protocolo



Any



TCP



UDP



ICMPv4



ICMPv6

Acción



Permitir



Denegar

Prioridad * ⓘ

310



Nombre

Permitir-MySQL

Descripción



Guardar

Cancelar

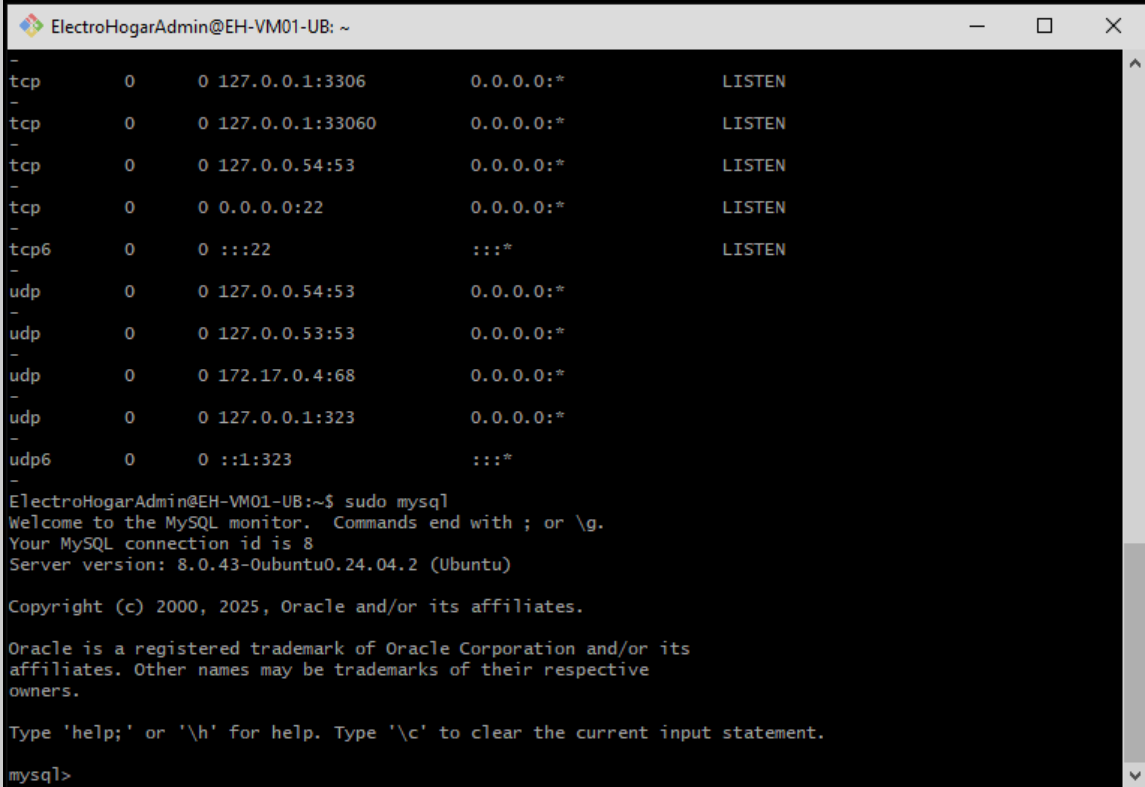


Enviar comentarios

Creación de usuario remoto en MySQL:

Accedemos a MySQL con usuario administrador (root) usando el comando: `sudo mysql`

Una vez dentro, mostramos los usuarios de MySQL con: `SELECT User, Host FROM mysql.user;`



The screenshot shows a terminal window titled 'ElectroHogarAdmin@EH-VM01-UB: ~'. It displays the status of the MySQL service, which is running and listening on several ports (3306, 33060, 5453, 22, 22, 5453, 5353, 468, 323, 323). Below the status, the user runs 'sudo mysql', which opens the MySQL command-line interface. The prompt changes from 'ElectroHogarAdmin@EH-VM01-UB:~\$' to 'mysql>'. The MySQL welcome message is displayed, including the connection ID (8) and server version (8.0.43-0ubuntu0.24.04.2).

```
ElectroHogarAdmin@EH-VM01-UB: ~  
-  
tcp        0      0 127.0.0.1:3306          0.0.0.0:*        LISTEN  
-  
tcp        0      0 127.0.0.1:33060         0.0.0.0:*        LISTEN  
-  
tcp        0      0 127.0.0.54:53          0.0.0.0:*        LISTEN  
-  
tcp        0      0 0.0.0.0:22              0.0.0.0:*        LISTEN  
-  
tcp6       0      0 :::22                   :::*              LISTEN  
-  
udp        0      0 127.0.0.54:53          0.0.0.0:*          
-  
udp        0      0 127.0.0.53:53          0.0.0.0:*          
-  
udp        0      0 172.17.0.4:68           0.0.0.0:*          
-  
udp        0      0 127.0.0.1:323          0.0.0.0:*          
-  
udp6       0      0 :::323                  :::*                
-  
ElectroHogarAdmin@EH-VM01-UB:~$ sudo mysql  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 8  
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)  
  
Copyright (c) 2000, 2025, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

Lo que queremos lograr es que el usuario creado pueda acceder desde cualquier host (diferentes IP's), para ello usamos el comando: `CREATE USER 'nombre_usuario'@'%' IDENTIFIED BY 'tu_contraseña';`

Después de esto el nuevo usuario se podrá visualizar en el listado de MySQL de manera similar al usuario “elcuality”.

Para otorgar permisos al usuario creado utilizamos: `GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'%;`

Aplicamos los cambios con: `FLUSH PRIVILEGES;`

```
ElectroHogarAdmin@EH-VM01-UB: ~
mysql> SELECT User, Host FROM mysql.user;
+-----+-----+
| User          | Host          |
+-----+-----+
| debian-sys-maint | localhost    |
| mysql.infoschema | localhost    |
| mysql.session   | localhost    |
| mysql.sys       | localhost    |
| root           | localhost    |
+-----+-----+
5 rows in set (0.00 sec)

mysql> CREATE USER 'ElectroHogarAdmin'@'%' IDENTIFIED BY 'Sistem@sOperativosGrupo1';
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT User, Host FROM mysql.user;
+-----+-----+
| User          | Host          |
+-----+-----+
| ElectroHogarAdmin | %            |
| debian-sys-maint | localhost    |
| mysql.infoschema | localhost    |
| mysql.session   | localhost    |
| mysql.sys       | localhost    |
| root           | localhost    |
+-----+-----+
6 rows in set (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'ElectroHogarAdmin'@'%' ;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

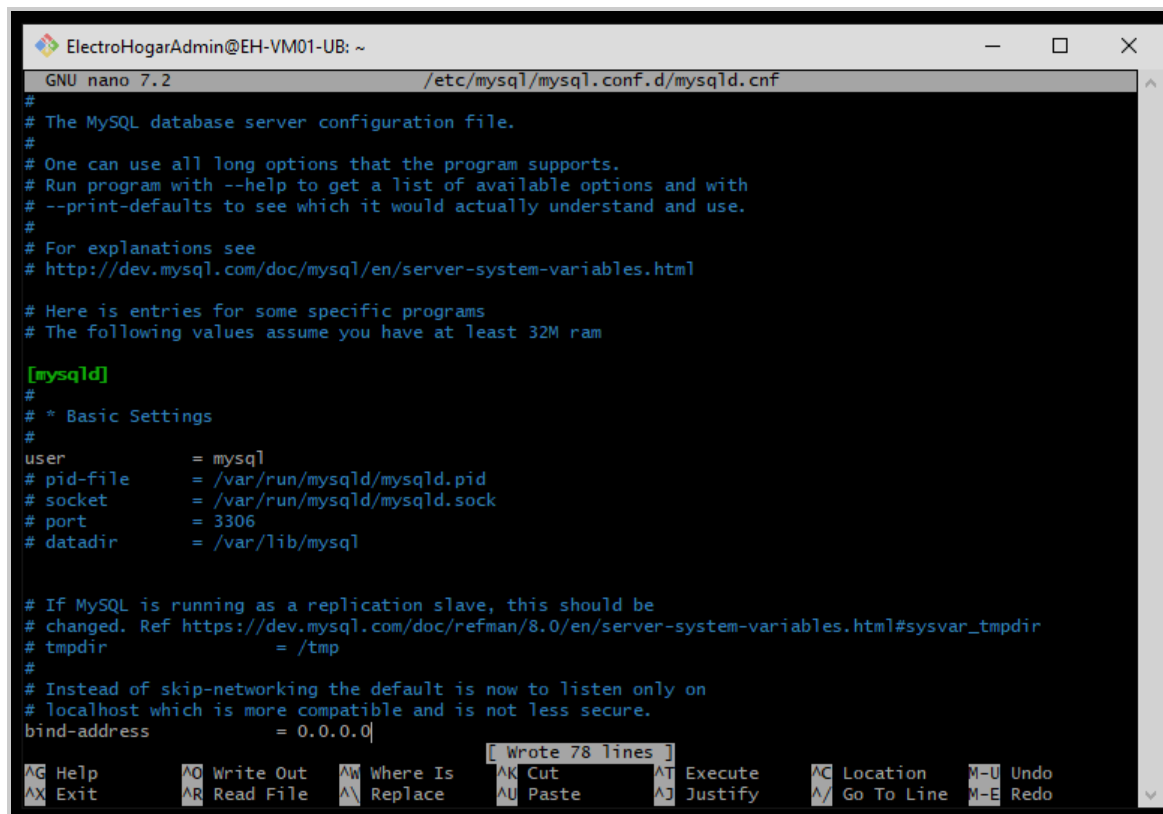
Salimos del entorno MySQL con: `EXIT;`

Edición de configuración de MySQL:

Un paso primordial para lograr establecer conexión remota es modificar el archivo de configuración de MySQL, podemos acceder a él con: `sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf`

Después de acceder, sobrescribimos la dirección del bind-address con “0.0.0.0”, para que MySQL pueda escuchar conexiones de entrada desde cualquier dirección IP.

Guardamos el archivo modificado y para aplicar los cambios reiniciamos el servicio MySQL con el comando: `sudo systemctl restart mysql`



The screenshot shows a terminal window titled "ElectroHogarAdmin@EH-VM01-UB: ~". The terminal is running the GNU nano 7.2 editor, editing the file `/etc/mysql/mysql.conf.d/mysqld.cnf`. The file content is as follows:

```
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
datadir             = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
tmpdir              = /tmp

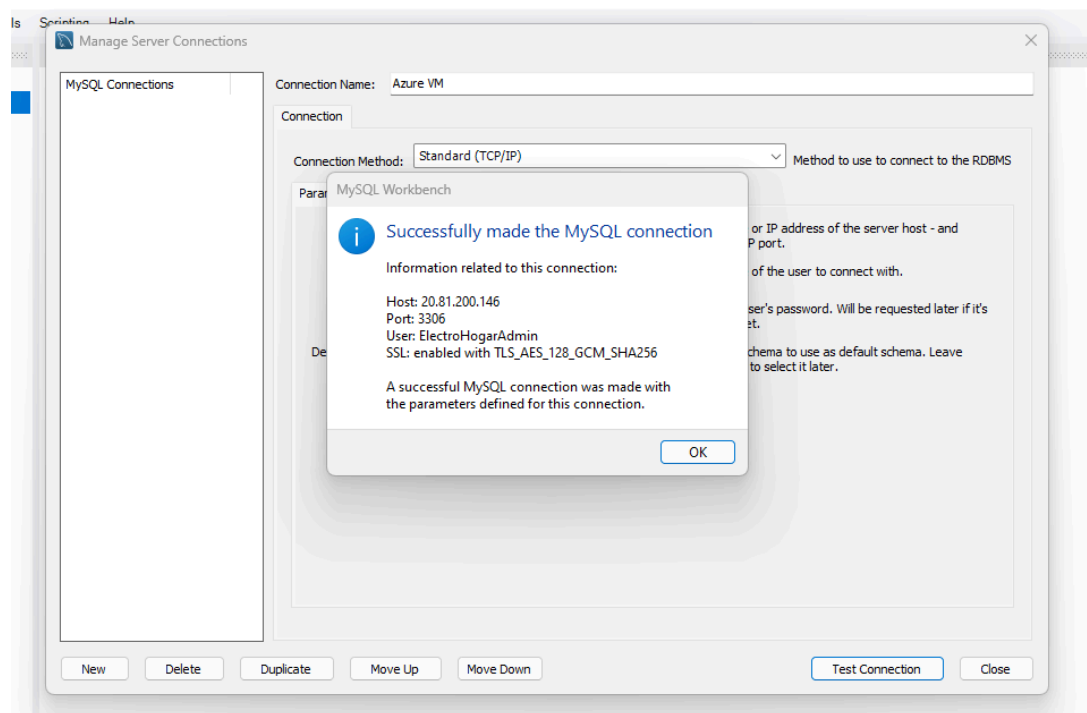
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0
```

The bottom of the terminal shows the nano editor's command shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo

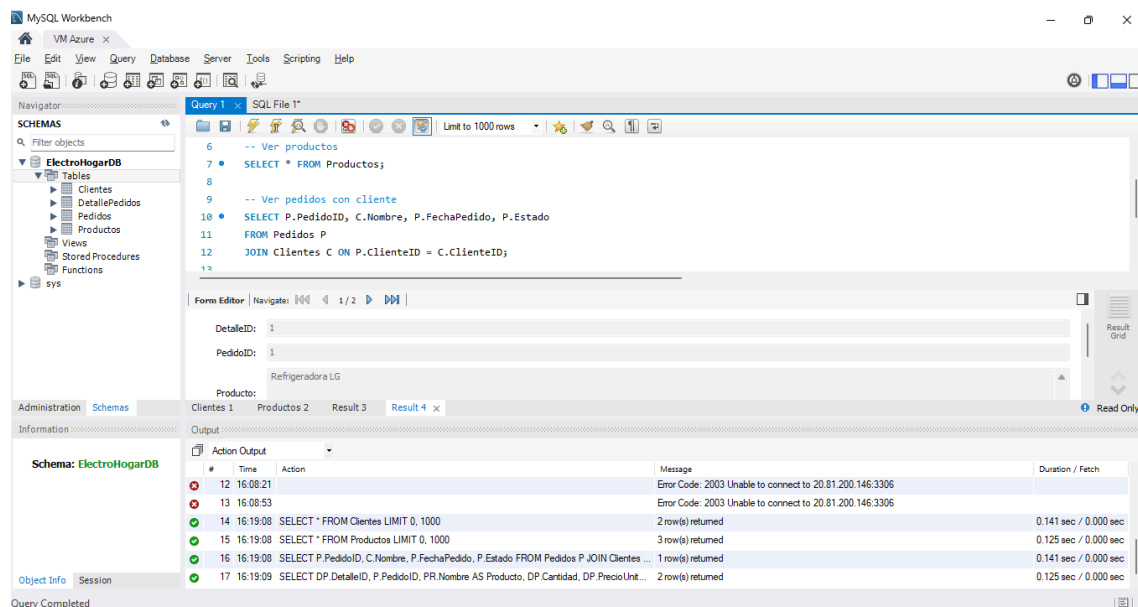
Conexión desde MySQL WorkBench:

Ahora solo tenemos que crear una conexión en MySQL Workbench y acceder con las credenciales que son necesarias como: IP de la VM, nombre de usuario y contraseña creado en el servicio de MySQL en la VM.



Creación de la base de datos y las tablas:

Una vez nos hayamos conectado de manera remota, podemos crear una base de datos y sus respectivas tablas desde MySQL Workbench.



También podemos verificar la creación de nuestra base de datos a través del comando:

SHOW DATABASES;

6. Arquitectura de la solución

La arquitectura propuesta de **ElectroHogar** es de **tres capas en la nube**:

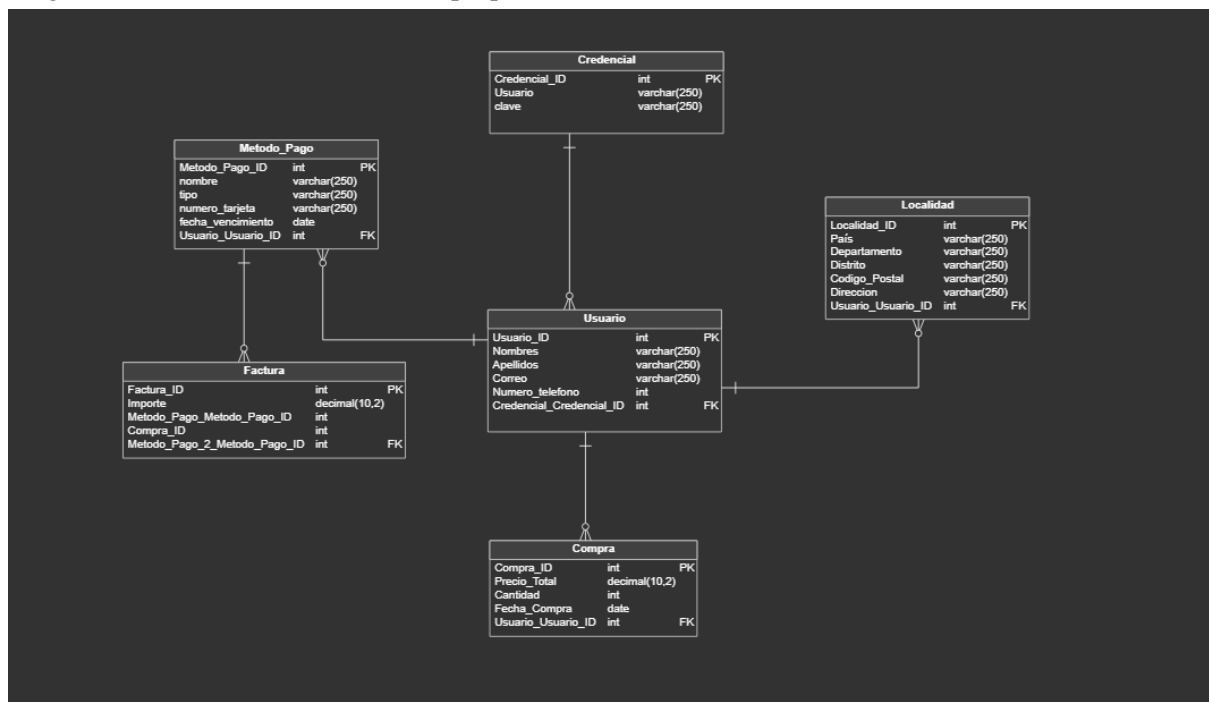
1. **Capa de Presentación:** Interfaz web y móvil que permite la interacción de los usuarios finales.
2. **Capa de Aplicación:** Lógica de negocio que gestiona pedidos, pagos, catálogos y usuarios.
3. **Capa de Datos:** Base de datos relacional con alta disponibilidad y respaldo.

7. Conclusiones

La implementación de **ElectroHogar** permitirá a la empresa de electrodomésticos mejorar su competitividad, optimizar procesos internos y ofrecer un servicio confiable y seguro a sus clientes. Asimismo, el diseño basado en tres capas garantiza escalabilidad y mantenimiento a largo plazo, alineándose con las mejores prácticas de la industria.

8. Anexos

Diagrama entidad-relación de nuestra propuesta:



MariaDB Foundation. (2025). *MariaDB Documentation*. <https://mariadb.org>

Debian Project. (2025). *Debian 12 Documentation*. <https://www.debian.org/doc/>

DBeaver. (2025). *DBeaver User Guide*. <https://dbeaver.io>

Microsoft Azure. (2025). *Azure Cloud Services*. <https://azure.microsoft.com>

Oracle. (2025). *MySQL 8.0 Reference Manual*. Oracle. Recuperado de <https://downloads.mysql.com/docs/refman-8.0-en.pdf> [MySQL Community Downloads](#)

Canonical Ltd. (2025). *Ubuntu Server documentation*. Ubuntu. Recuperado de <https://documentation.ubuntu.com/server/> [Documentación de Ubuntu](#)

Microsoft. (2025). *Virtual Machines in Azure*. Microsoft Learn. Recuperado de <https://learn.microsoft.com/en-us/azure/virtual-machines/> [Microsoft Learn](#)