



REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DEL PODER POPULAR PARA EDUCACIÓN UNIVERSITARIA
UNIVERSIDAD POLITÉCNICA TERRITORIAL DEL ESTADO ARAGUA
“FEDERICO BRITO FIGUEROA”
LA VICTORIA – ESTADO ARAGUA.

Administración de sistemas informáticos y redes.

Módulo: Administración de sistemas gestores de bases de datos.

Proyecto: Replicación de una base de datos.

Versión: 1.0.0

Nombre	Jose Gregorio
Apellidos	Moreno Marcano
Módulo/Crédito	
UF (solo ciclos LOE)	
Título de la actividad	Replicación de una base de datos
Profesor:	Ing. Jackson Pérez
Guardado en el repositorio Git	https://github.com/jmoreno182/AdmBD

HISTORIAL DE REVISIONES

Versión	Autor	Descripción	Fecha
1.0	José G. Moreno	Creación del documento	Jun 2023

Índice de contenido

Introducción	4
1. Análisis de diseño de la base de datos propuesta (diseño físico).....	5
1.1. Identificación de mejoras	5
2. Configuración de las Máquinas Virtuales.....	5
2.1. Selección de VirtualBox como plataforma de virtualización.....	5
2.2. Creación de las Máquinas Virtuales.....	6
2.3. Asignación de recursos a las Máquinas Virtuales	8
3. Instalación del Sistema Gestor de Base de Datos	9
3.1. Descarga e Instalación de XAMPP	9
3.2. Verificación de la Conectividad y Funcionamiento	10
4. Creación del Script SQL de la Base de Datos.....	12
4.1. Preparación del entorno.....	12
4.2. Creación del script SQL para la Base de datos	14
4.3. Creación del script SQL para las tablas y campos	15
4.4. Validación del script SQL	23
4.5. Definición de las relaciones entre las tablas.....	23
4.6. Desarrollo de las claves foráneas (Foreign Keys)	24
5. Configuración de la Replicación de la Base de Datos	32
5.1. Configuración de las Máquinas Master y Slave	32
5.2. Configuración del Sistema Gestor de Base de Datos	32
5.3. Inicio de la Replicación	37
6. Prueba y Validación del Sistema	44
6.1. Pruebas de Funcionalidad.....	44
6.2. Validación de Replicación	51
7. Conclusiones y Recomendaciones	54
7.1. Conclusiones	54
7.2. Recomendaciones	54

Introducción

El presente informe técnico tiene como objetivo proporcionar el paso a paso detallado para el despliegue de una base de datos y la configuración de la replicación utilizando máquinas virtuales, haciendo uso de Visual Studio Code como herramienta de desarrollo.

En este informe, se abordarán los aspectos clave relacionados con el despliegue de la base de datos, comenzando por el análisis del diseño propuestos, pasando por la configuración de las máquinas virtuales que albergarán el entorno de base de datos, la instalación del sistema gestor de bases de datos y la creación del script SQL necesario para la creación de la estructura y objetos de la base de datos.

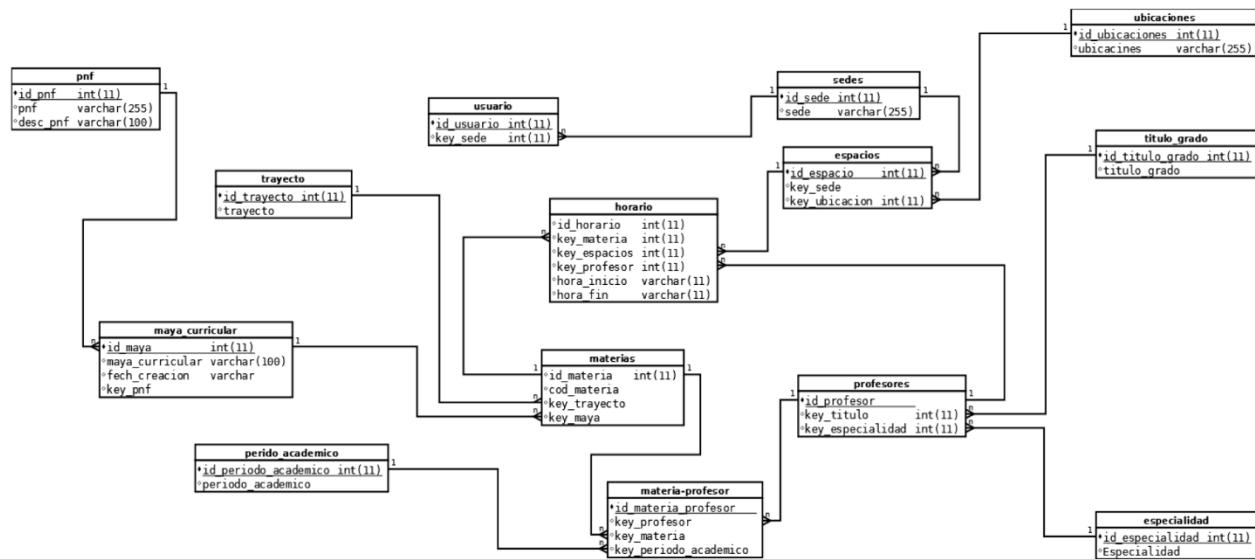
Además, se explorará la configuración de la replicación, un aspecto fundamental para garantizar la disponibilidad y la redundancia de la base de datos en entornos de alta demanda y críticos. Se presentarán distintas alternativas para la configuración de la replicación y se detallarán los pasos necesarios para su implementación y monitoreo.

El uso de máquinas virtuales ofrece flexibilidad y escalabilidad al entorno de base de datos, permitiendo la replicación y el despliegue en diferentes entornos sin la necesidad de infraestructuras físicas costosas. Asimismo, Visual Studio Code proporciona una plataforma de desarrollo versátil y amigable, facilitando la creación y modificación del script SQL de manera eficiente.

Es importante destacar que este informe se basa en un enfoque general y se adaptará según los requisitos y especificaciones propuestos para dicha actividad. A lo largo del documento se proporcionarán recomendaciones y mejores prácticas, permitiendo una implementación exitosa y la base para futuras mejoras del entorno.

A continuación, se procederá con el análisis del diseño físico propuesto de la base de datos, donde se evaluarán los requisitos, objetos, entidades, atributos, tipo, longitudes y valores para la posterior creación del script SQL.

1. Análisis de diseño de la base de datos propuesta (diseño físico)



1.1. Identificación de mejoras

- ✓ Para todas las llaves primarias y foráneas se utilizarán tipo de datos INT con longitud (11), auto incrementables.
- ✓ En la tabla “maya_curricular” se propone cambiar el tipo de dato VARCHAR del campo “fech_creación” por el tipo de dato datetime con DEFAULT current_timestamp().
- ✓ En la tabla “horario” se propone cambiar el tipo de dato VARCHAR de los campos “hora_inicio” y “hora_fin” por el tipo de dato time.

2. Configuración de las Máquinas Virtuales

En esta sección, se describirá el proceso de configuración de las máquinas virtuales utilizando VirtualBox como plataforma de virtualización. A continuación, se detallarán los pasos necesarios para crear y configurar la máquina virtual en la que se desplegará como esclavo el entorno de base de datos.

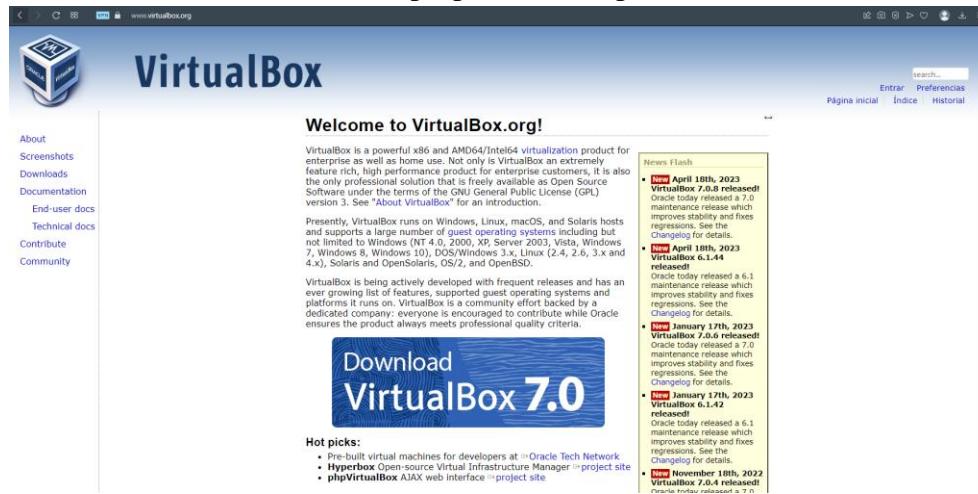
2.1. Selección de VirtualBox como plataforma de virtualización

VirtualBox es una plataforma de virtualización ampliamente utilizada y de código abierto que permite crear y ejecutar máquinas virtuales en diversos sistemas operativos. Proporciona una interfaz intuitiva y robusta para administrar y configurar entornos virtuales, lo cual resulta ideal para nuestro escenario de despliegue de base de datos.

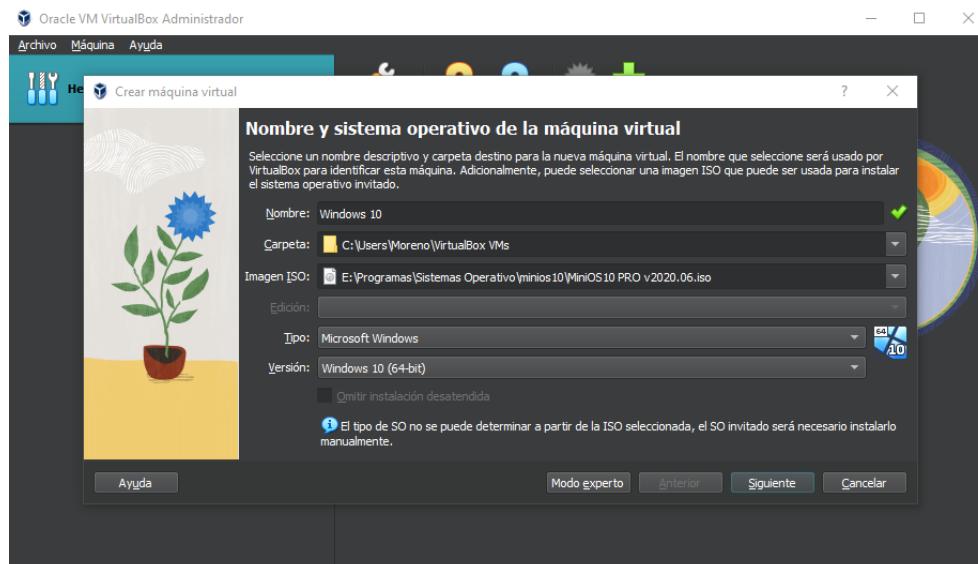
2.2. Creación de las Máquinas Virtuales

A continuación, se describirán los pasos para crear las máquinas virtuales necesarias:

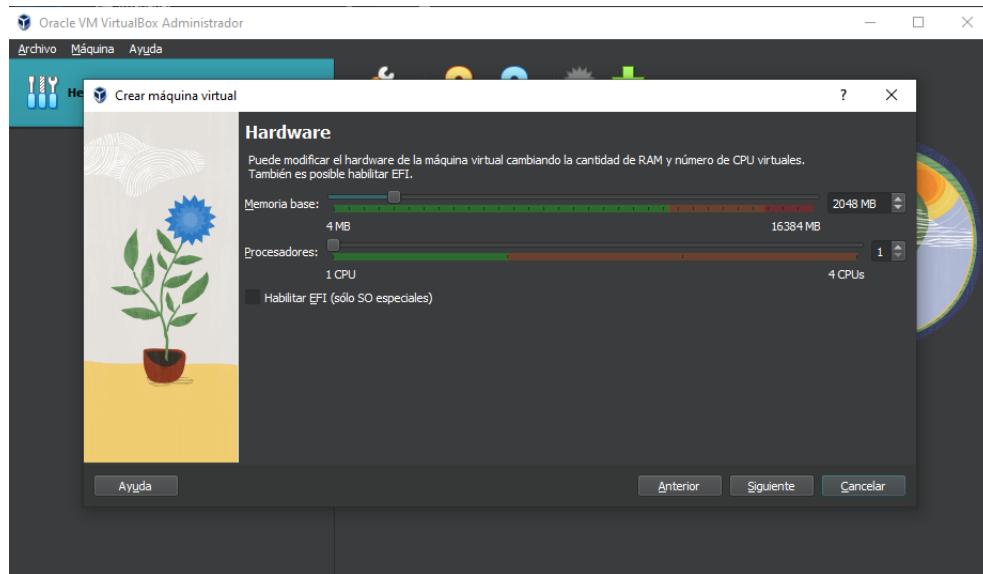
1. Descarga e instalación de VirtualBox: Visita el sitio web oficial de VirtualBox (<https://www.virtualbox.org>) y descarga la versión adecuada para tu sistema operativo. Sigue las instrucciones de instalación proporcionadas por el asistente de instalación.



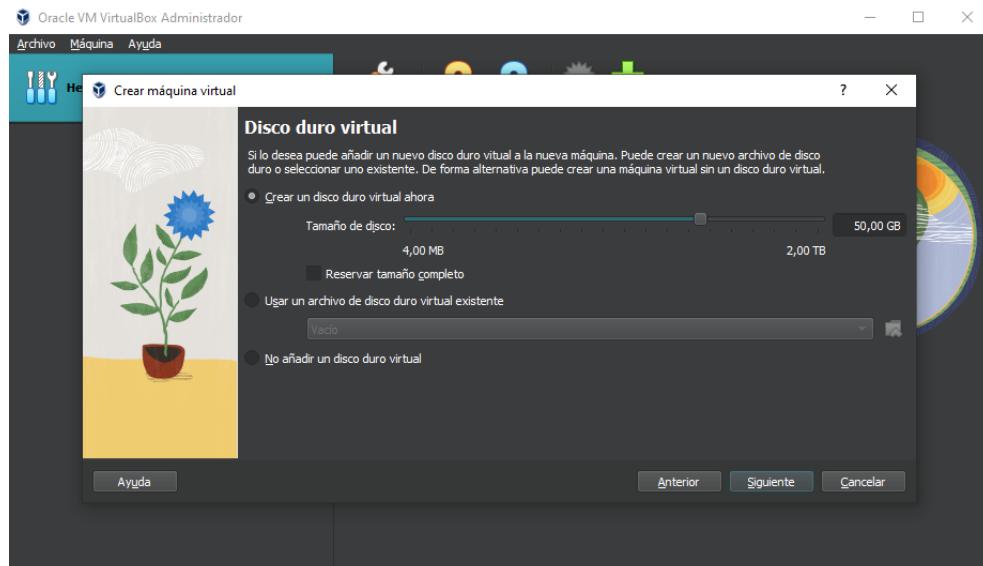
2. Creación de una nueva máquina virtual: Abre VirtualBox y haz clic en el botón "Nuevo" para crear una nueva máquina virtual. Ingresa un nombre descriptivo para la máquina virtual y selecciona el tipo de sistema operativo que se ejecutará en ella.



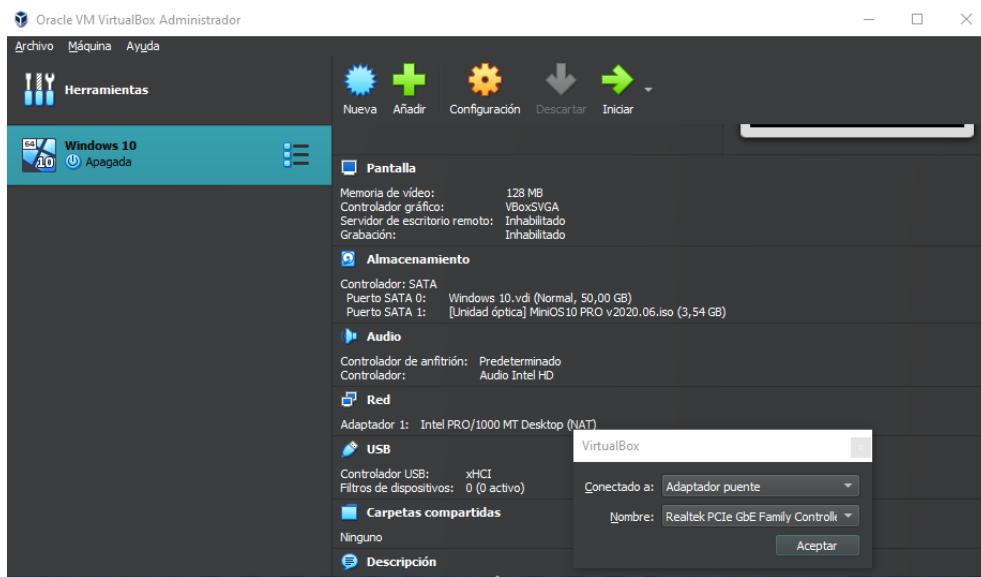
3. Asignación de recursos: Configura la cantidad de memoria RAM y espacio en disco que se asignarán a la máquina virtual. Asegúrate de asignar suficientes recursos para el funcionamiento óptimo de la base de datos y sus requerimientos específicos.



4. Configuración de almacenamiento: Crea un nuevo disco duro virtual o utiliza uno existente para almacenar el sistema operativo y los archivos de la base de datos. Selecciona el tipo de archivo de disco virtual más adecuado para tu entorno.



5. Configuración de la red: En la configuración de red de la máquina virtual, selecciona la opción "Adaptador puente" para permitir la conectividad de red entre las máquinas virtuales y con el host.



6. Configuración de otros parámetros: Explora las diferentes configuraciones disponibles en VirtualBox, como el sonido, las opciones de visualización y las carpetas compartidas, según las necesidades de tu proyecto.

2.3. Asignación de recursos a las Máquinas Virtuales

Es importante asignar recursos adecuados a las máquinas virtuales para garantizar un rendimiento óptimo del entorno de base de datos. Considera los siguientes aspectos:

- ✓ Memoria RAM: Asigna una cantidad suficiente de memoria RAM a cada máquina virtual para evitar cuellos de botella y asegurar un procesamiento eficiente de las consultas y transacciones de la base de datos.
- ✓ Almacenamiento: Asigna suficiente espacio en disco para las necesidades de almacenamiento de la base de datos y los archivos asociados. Considera el crecimiento esperado de la base de datos a lo largo del tiempo.
- ✓ Procesador: Asigna un número adecuado de núcleos de procesador a cada máquina virtual para aprovechar al máximo la capacidad de procesamiento disponible.
- ✓ Red: Configura la red de cada máquina virtual de manera que puedan comunicarse entre sí y con el host. Asegúrate de seleccionar una configuración de red que se adapte a tu entorno y necesidades específicas.

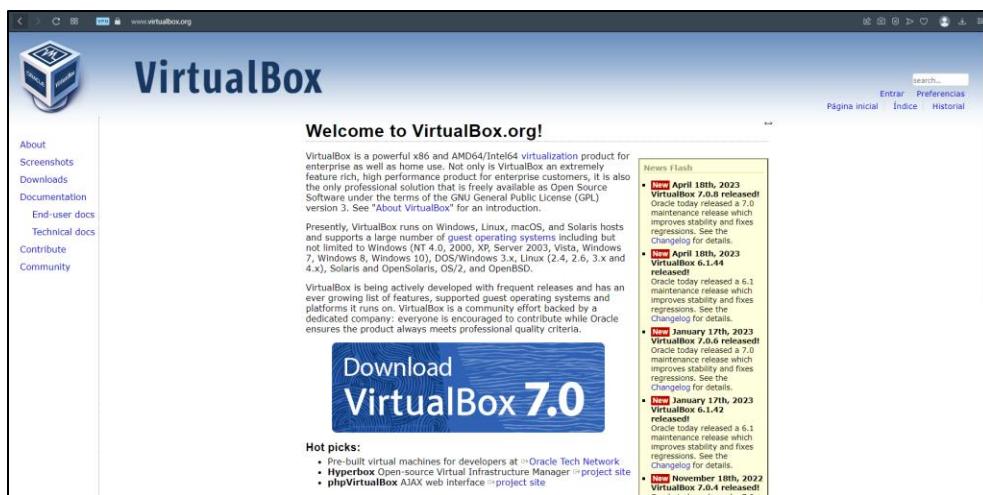
Siguiendo estos pasos, habrás creado las máquinas virtuales necesarias utilizando VirtualBox y asignando los recursos correspondientes a cada una. A continuación, se procederá con la instalación del sistema gestor de bases de datos en las máquinas virtuales.

3. Instalación del Sistema Gestor de Base de Datos

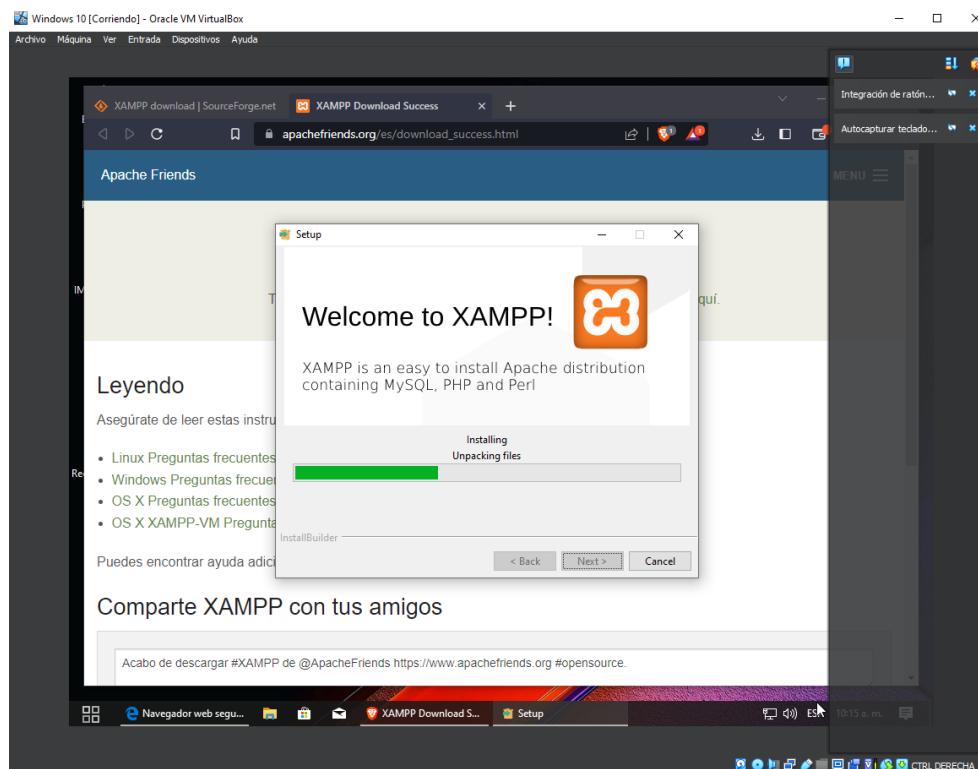
En esta sección, se describirá el proceso de instalación del sistema gestor de bases de datos en una máquina virtualizada con Windows 10. En este caso, utilizaremos XAMPP, una solución integrada que incluye Apache, MySQL, PHP y phpMyAdmin. A continuación, se detallarán los pasos para instalar XAMPP y configurar el entorno de base de datos.

3.1. Descarga e Instalación de XAMPP

1. Descarga de XAMPP: Accede al sitio web oficial de XAMPP (<https://www.apachefriends.org/es/index.html>) y descarga la versión compatible con Windows. Asegúrate de seleccionar la versión que incluye MySQL como componente.



2. Ejecución del instalador: Una vez completada la descarga, ejecuta el archivo de instalación de XAMPP. Aparecerá un asistente de instalación que te guiará a través del proceso.
3. Selección de componentes: Durante la instalación, se te presentará una lista de componentes disponibles. Asegúrate de seleccionar MySQL para instalar el sistema gestor de bases de datos.
4. Configuración de la instalación: El asistente te pedirá que selecciones la ubicación de instalación de XAMPP. Puedes dejar la ubicación predeterminada o elegir una diferente según tus preferencias.
5. Inicio de la instalación: Haz clic en el botón "Instalar" para comenzar el proceso de instalación. Espera a que se complete la instalación de XAMPP en tu máquina virtual.



6. Configuración adicional: Durante la instalación, se te pedirá que configures algunas opciones adicionales, como la configuración del servidor MySQL y la contraseña de root. Proporciona la información requerida según tus necesidades.

3.2. Verificación de la Conectividad y Funcionamiento

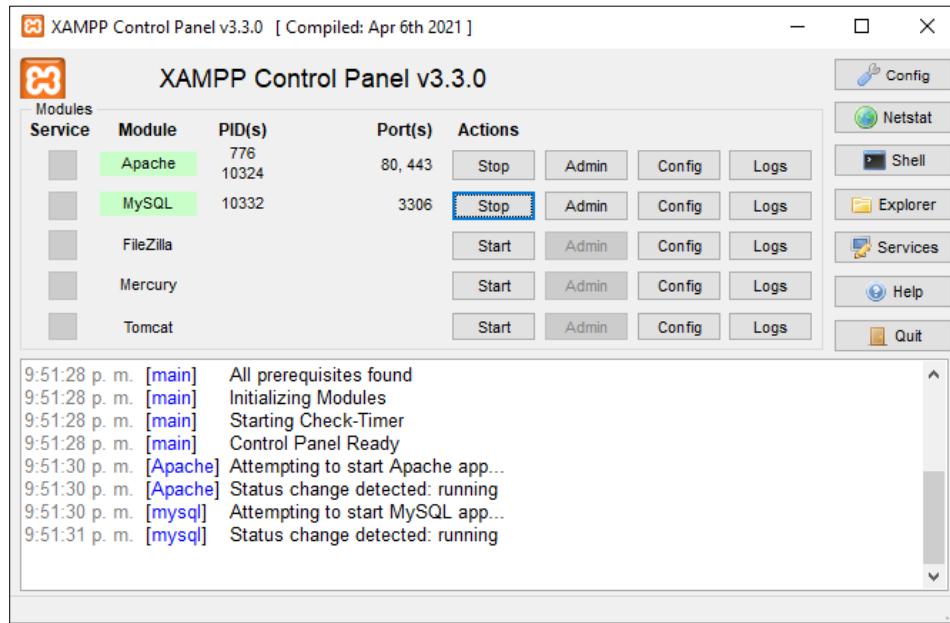
Una vez completada la instalación, es importante verificar la conectividad y el funcionamiento adecuado del sistema gestor de bases de datos. Sigue estos pasos:

1. Inicio de XAMPP: Abre XAMPP desde el menú de inicio o desde la ubicación donde se instaló. Asegúrate de ejecutarlo como administrador para evitar problemas de permisos.
2. Verificación de los servicios: En la interfaz de XAMPP, asegúrate de que los servicios de Apache y MySQL estén activos. Si alguno de los servicios no está en funcionamiento, intenta iniciarlos manualmente.

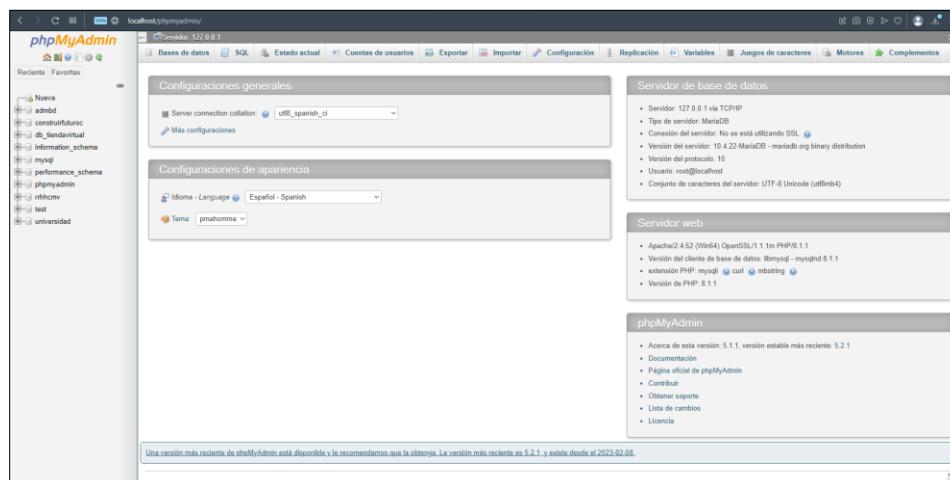
Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>



3. Acceso a phpMyAdmin: Abre un navegador web e ingresa la siguiente URL: <http://localhost/phpmyadmin>. Deberías poder acceder a la interfaz de phpMyAdmin, que te permitirá administrar y realizar consultas en la base de datos.
4. Prueba de conexión: En phpMyAdmin, verifica que puedas conectarte al servidor MySQL y acceder a la base de datos predeterminada. Intenta realizar algunas consultas básicas para asegurarte de que todo esté funcionando correctamente.



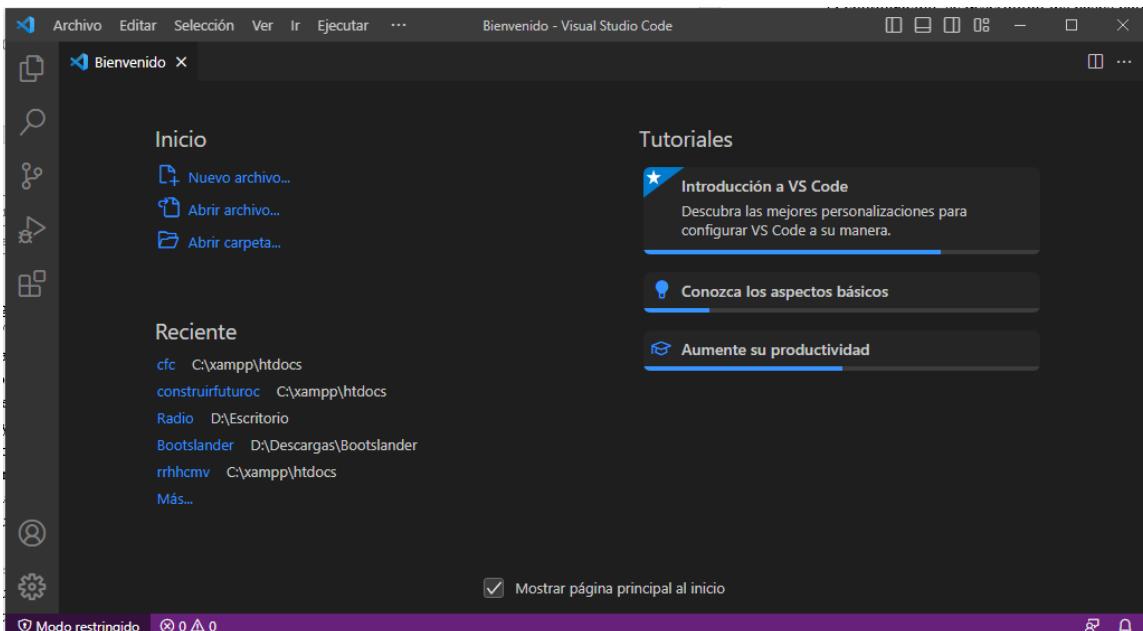
Al completar estos pasos, habrás instalado y configurado XAMPP en tu máquina virtualizada con Windows 10. Ahora, puedes proceder a la creación del script SQL de la base de datos en Visual Studio Code, como se detallará en la siguiente sección.

4. Creación del Script SQL de la Base de Datos

En esta sección, se detallarán los pasos para crear el script SQL de la base de datos en Visual Studio Code. El objetivo es definir la estructura de las tablas, los campos y las relaciones necesarias para almacenar y gestionar los datos de manera eficiente. A continuación, se presentan los pasos a seguir:

4.1. Preparación del entorno

1. **Abre Visual Studio Code:** Inicia Visual Studio Code desde el menú de inicio o desde la ubicación donde se encuentra instalado en tu máquina.

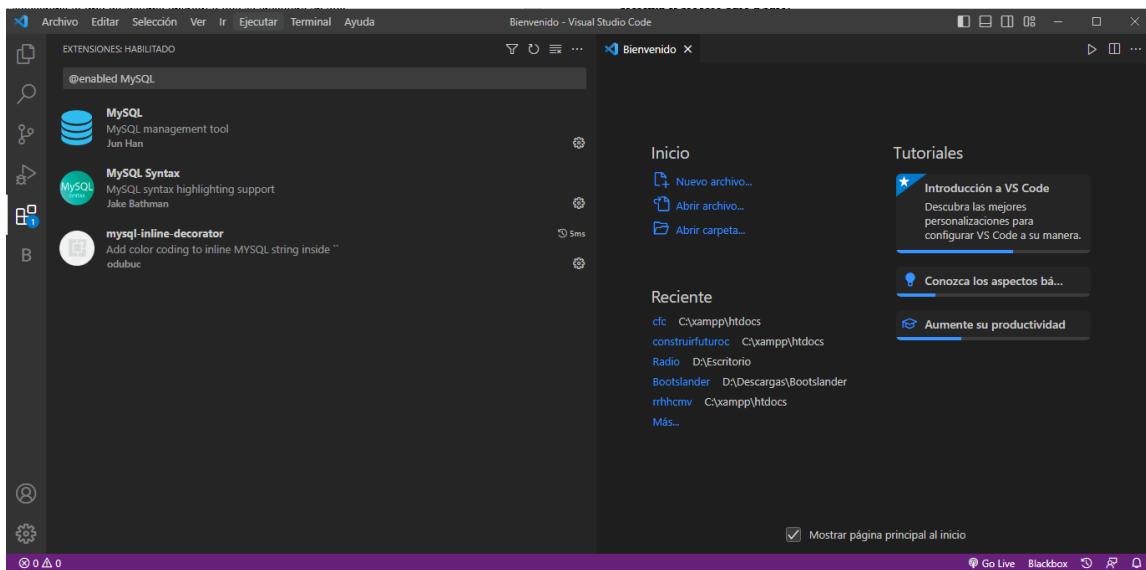


2. **Instalación de extensiones:** Verifica que las extensiones necesarias estén instaladas en Visual Studio Code. Algunas extensiones útiles para trabajar con bases de datos incluyen "SQL Server (MySQL)", "MySQL", "PostgreSQL" u otras según el sistema gestor de bases de datos utilizado, para este caso utilizaremos las extensiones "MySQL management tool", "MySQL Syntax" y "MySQL-inline-decorator" para agilizar el proceso de conexión y creación del script SQL.

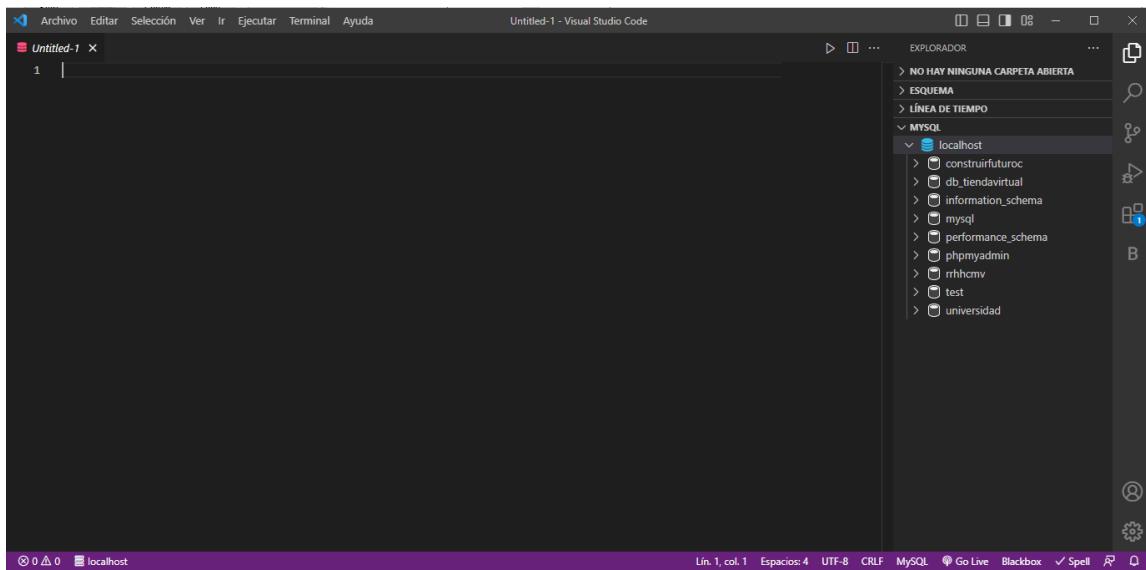
Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>



3. **Conexión a la base de datos:** Configura la conexión a la base de datos en Visual Studio Code utilizando la extensión correspondiente. Proporciona los detalles de conexión necesarios, como la dirección del servidor, el nombre de usuario y la contraseña hasta lograr visualizar en el explorador dentro de la sección MYSQL las bases de datos creadas en la dirección del servidor seleccionado.

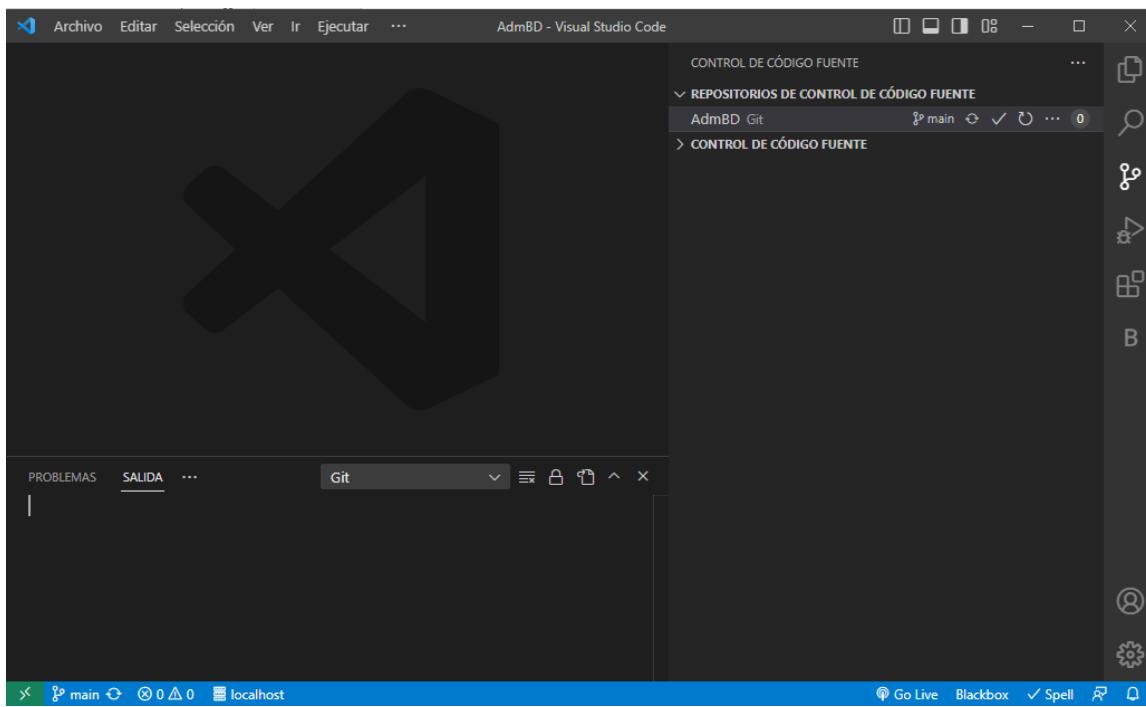


4. **Creación de un repositorio git (opcional):** Para utilizar las características de git como el control de código fuente en VS Code, se crea una carpeta para que contenga un repositorio o clonarlos desde una dirección URL, seleccionando la dirección de destino.

Documento de memoria técnica

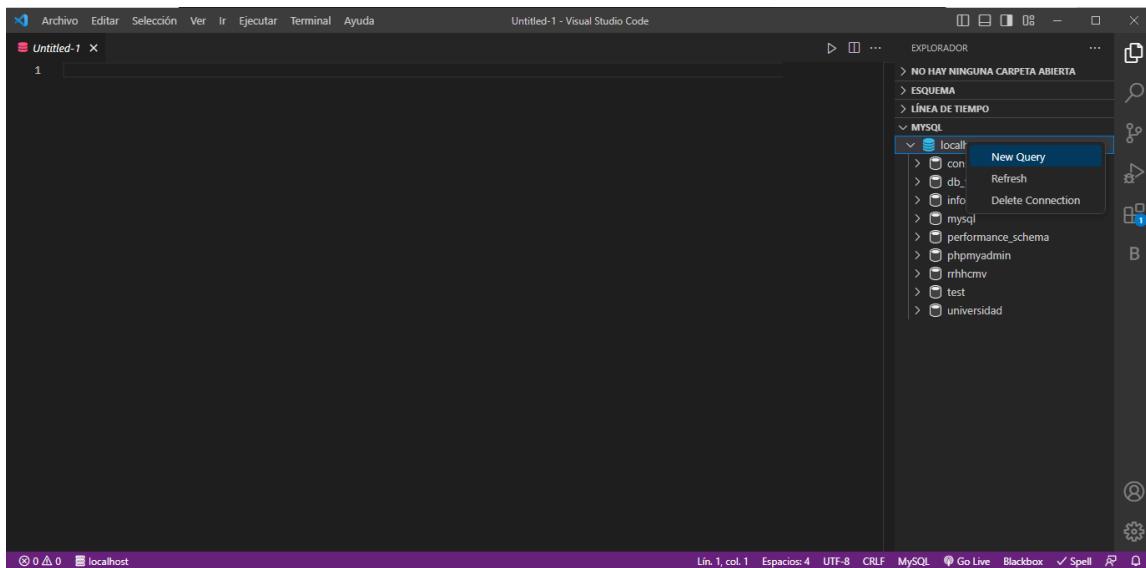
Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>



4.2. Creación del script SQL para la Base de datos

1. Se crea un nuevo archivo en Visual Studio Code haciendo clic derecho sobre la dirección del servidor “localhost” y seleccione la opción New Query.

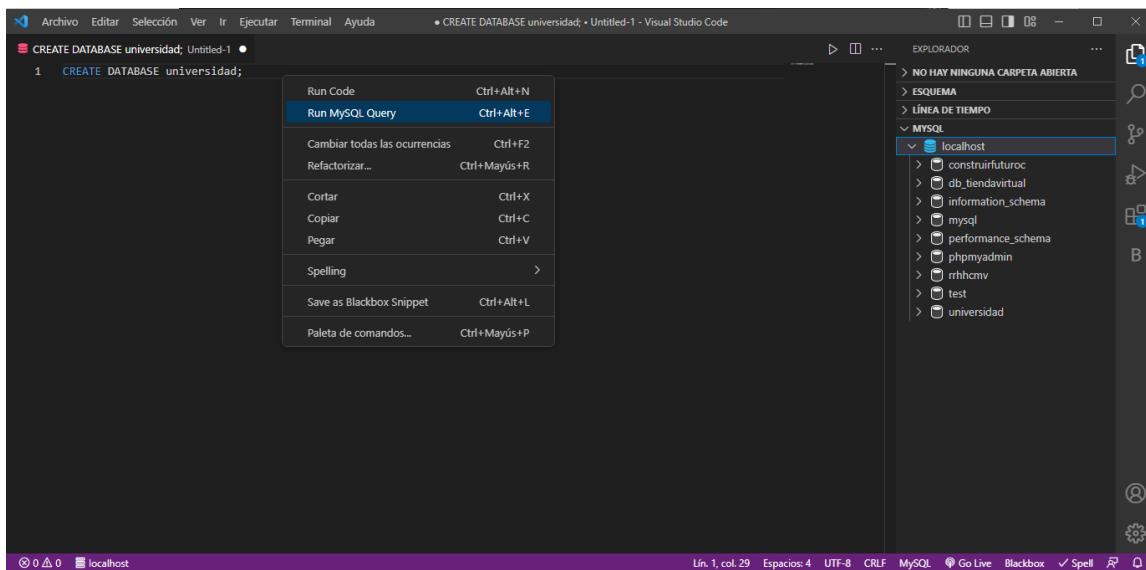


2. Se escribe la sentencia SQL “CREATE DATABASE nombre del proyecto” y presionando clic derecho se selecciona la opción “Run MySQL Query” para crear la base de datos.

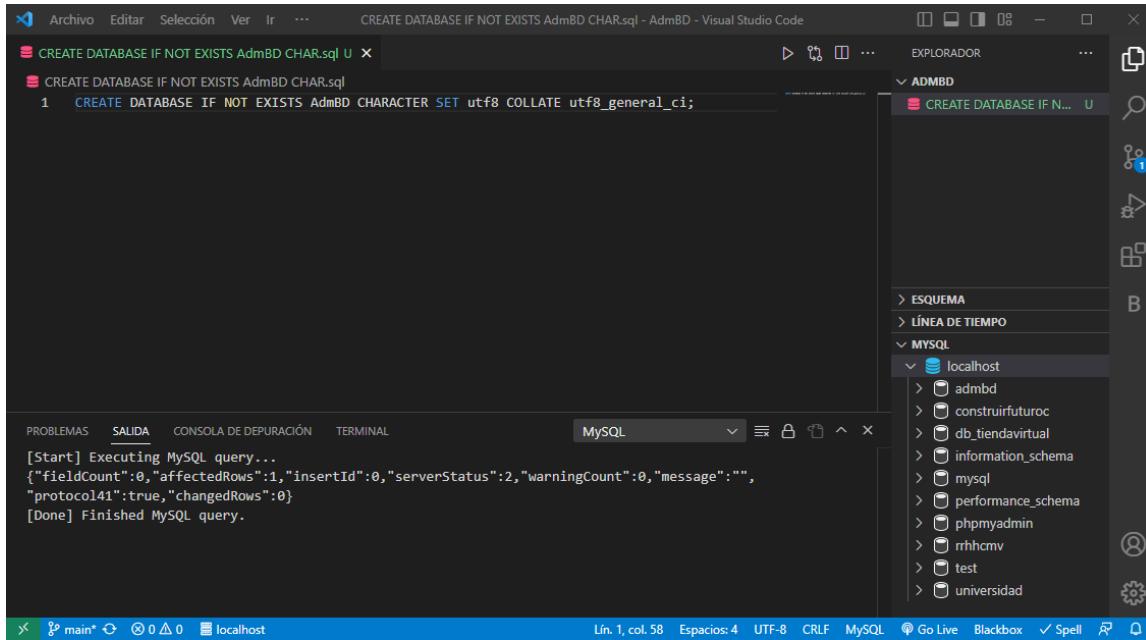
Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>



3. Se verifica el resultado (salida) mostrado en la consola de VS Code para constatar que no contiene ningún error, en este paso se procedió como evento educativo incorporar una sintaxis más estilizada para realizar el ejemplo que se muestra a continuación en la línea 1 del código.



4.3. Creación del script SQL para las tablas y campos

1. Se analiza los requerimientos y la estructura de los datos para identificar las entidades principales que se deben representar en la base de datos. Se utilizó la sintaxis SQL

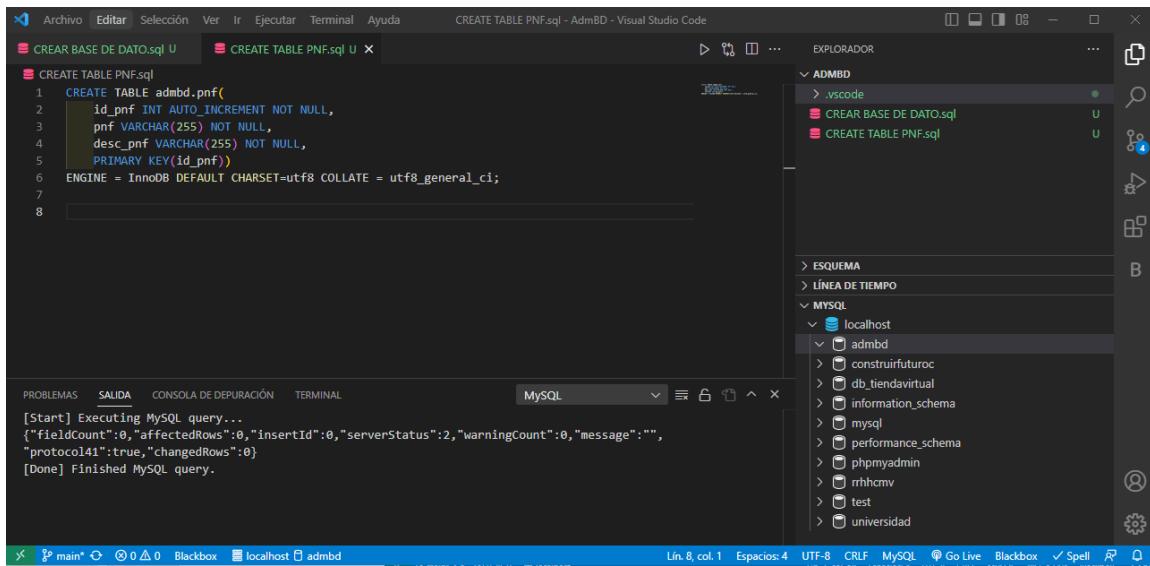
Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

“CREATE TABLE” para crear las tablas correspondientes a cada entidad, definiendo los campos adecuados, incluyendo el nombre, el tipo de dato y las restricciones necesarias, asignando una columna clave primaria a cada tabla para garantizar la unicidad de los registros.

Ilustración 1. Tabla-entidad "pnf"



The screenshot shows a Visual Studio Code interface with a MySQL extension. The left sidebar has tabs for 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL'. The 'TERMINAL' tab shows the execution of a MySQL query to create the 'PNF' table in the 'admbd' database. The query defines the table structure with columns: id_pnf (INT), pnf (VARCHAR(255)), and desc_pnf (VARCHAR(255)). The primary key is set to 'id_pnf'. The 'CREATE TABLE PNF.sql' file is open in the editor. The right sidebar shows the 'EXPLORADOR' (Explorer) view, which lists the database structure under 'ADMBD' and 'ESQUEMA' (Schema). The 'admbd' database is expanded, showing tables like 'construirfuturoc', 'db_tiendavirtual', 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'rthcmv', 'test', and 'universidad'. The 'pnf' table is also listed under the schema.

```
CREATE TABLE PNF.sql - AdmBD - Visual Studio Code
CREATE TABLE PNF.sql U
CREATE TABLE PNF.sql U

CREATE TABLE PNF.sql
1 CREATE TABLE admbd.pnf(
2     id_pnf INT AUTO_INCREMENT NOT NULL,
3     pnf VARCHAR(255) NOT NULL,
4     desc_pnf VARCHAR(255) NOT NULL,
5     PRIMARY KEY(id_pnf)
6 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;

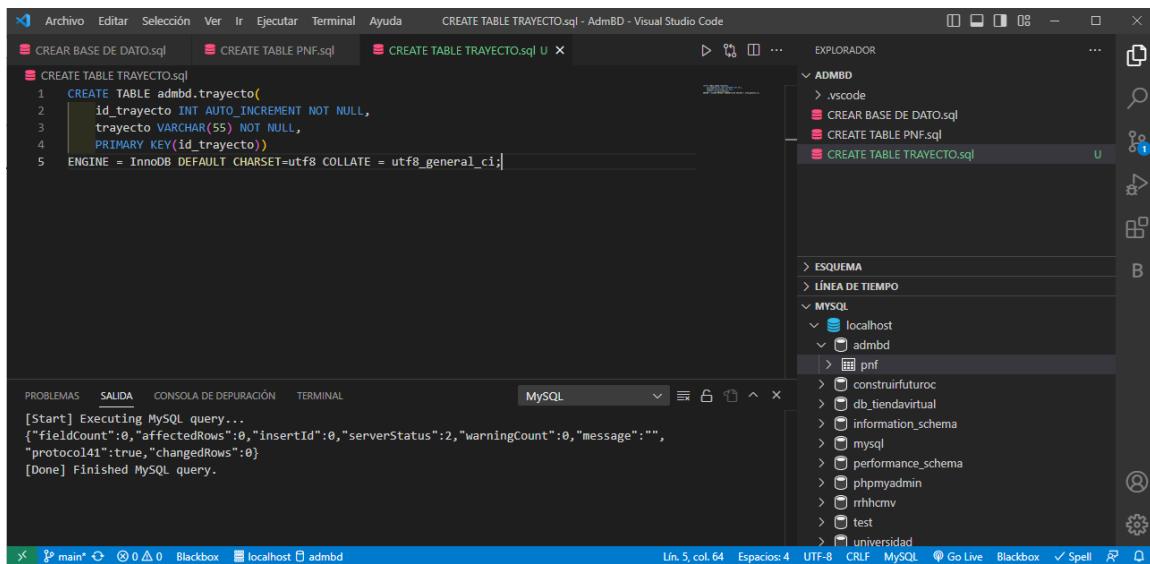
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL MySQL Lin. 8, col. 1 Espacios: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"",
"protocol41":true,"changedRows":0}
[Done] Finished MySQL query.

EXPLORADOR
ADMBD
    .vscode
        CREAR BASE DE DATO.sql
        CREATE TABLE PNF.sql

    ESQUEMA
    LÍNEA DE TIEMPO
    MySQL
        localhost
            admbd
                construirfuturoc
                db_tiendavirtual
                information_schema
                mysql
                performance_schema
                phpmyadmin
                rthcmv
                test
                universidad

B
```

Ilustración 2. Tabla-entidad "trayecto"



The screenshot shows a Visual Studio Code interface with a MySQL extension. The left sidebar has tabs for 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL'. The 'TERMINAL' tab shows the execution of a MySQL query to create the 'TRAYECTO' table in the 'admbd' database. The query defines the table structure with columns: id_trayecto (INT), trayecto (VARCHAR(55)), and PRIMARY KEY(id_trayecto). The primary key is set to 'id_trayecto'. The 'CREATE TABLE TRAYECTO.sql' file is open in the editor. The right sidebar shows the 'EXPLORADOR' (Explorer) view, which lists the database structure under 'ADMBD' and 'ESQUEMA' (Schema). The 'admbd' database is expanded, showing tables like 'construirfuturoc', 'db_tiendavirtual', 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'rthcmv', 'test', and 'universidad'. The 'trayecto' table is also listed under the schema.

```
CREATE TABLE TRAYECTO.sql - AdmBD - Visual Studio Code
CREATE TABLE TRAYECTO.sql U
CREATE TABLE TRAYECTO.sql U

CREATE TABLE TRAYECTO.sql
1 CREATE TABLE admbd.trayecto(
2     id_trayecto INT AUTO_INCREMENT NOT NULL,
3     trayecto VARCHAR(55) NOT NULL,
4     PRIMARY KEY(id_trayecto)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL MySQL Lin. 5, col. 64 Espacios: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"",
"protocol41":true,"changedRows":0}
[Done] Finished MySQL query.

EXPLORADOR
ADMBD
    .vscode
        CREAR BASE DE DATO.sql
        CREATE TABLE PNF.sql
        CREATE TABLE TRAYECTO.sql

    ESQUEMA
    LÍNEA DE TIEMPO
    MySQL
        localhost
            admbd
                trayecto
                    construirfuturoc
                    db_tiendavirtual
                    information_schema
                    mysql
                    performance_schema
                    phpmyadmin
                    rthcmv
                    test
                    universidad

B
```

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 3. Tabla-entidad "periodo_academico"

The screenshot shows the Visual Studio Code interface with several tabs open at the top: 'CREATE TABLE PNF.sql', 'CREATE TABLE TRAYECTO.sql', and 'CREATE TABLE PERIODO ACADEMICO.sql'. The 'CREATE TABLE PERIODO ACADEMICO.sql' tab is active, displaying the following SQL code:

```
1 CREATE TABLE admdb.periodo_academico(
2     id periodo_academico INT AUTO_INCREMENT NOT NULL,
3     periodo_academico VARCHAR(55) NOT NULL,
4     PRIMARY KEY(id_periodo_academico)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The 'EXPLORADOR' sidebar on the right shows a tree structure of database objects. Under the 'ADMBD' node, there is a 'CREATE BASE DE DATO.sql' file. Under the 'MySQL' node, there is a 'localhost' connection which contains the 'admbd' schema, the 'pnf' database, and the 'periodo_academico' table. The 'pnf' database is currently selected. The 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL' panes are visible at the bottom.

Ilustración 4. Tabla-entidad "ubicaciones"

The screenshot shows the Visual Studio Code interface with several tabs open at the top: 'CREATE TABLE PERIODO ACADEMICO.sql U X', 'CREATE TABLE UBILOCACIONES.sql U', and 'CREATE TABLE UBILOCACIONES.sql'. The 'CREATE TABLE UBILOCACIONES.sql' tab is active, displaying the following SQL code:

```
1 CREATE TABLE admdb.ubicaciones(
2     id ubicaciones INT AUTO_INCREMENT NOT NULL,
3     ubicaciones VARCHAR(255) NOT NULL,
4     PRIMARY KEY(id_ubicaciones)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The 'EXPLORADOR' sidebar on the right shows a tree structure of database objects. Under the 'ADMBD' node, there is a 'CREATE BASE DE DATO.sql' file. Under the 'MySQL' node, there is a 'localhost' connection which contains the 'admbd' schema, the 'pnf' database, and the 'trayecto' table. The 'pnf' database is currently selected. The 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL' panes are visible at the bottom.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 5. Tabla-entidad "titulo_grado"

The screenshot shows the Visual Studio Code interface with several tabs open. The active tab is 'CREATE TABLE TITULO_GRADO.sql'. The code in this tab creates a table 'admbd.titulo_grado' with columns 'id_titulo_grado' (INT AUTO_INCREMENT NOT NULL) and 'titulo_grado' (VARCHAR(100) NOT NULL), and a primary key constraint. The engine is set to InnoDB with utf8_general_ci collation.

```
CREATE TABLE admbd.titulo_grado(
    id_titulo_grado INT AUTO_INCREMENT NOT NULL,
    titulo_grado VARCHAR(100) NOT NULL,
    PRIMARY KEY(id_titulo_grado)
) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The Explorer sidebar on the right lists various database objects: ADMBD (CREATE TABLE TITULO_GRADO.sql, CREATE TABLE PERIODO_ACADEMICO.sql, CREATE TABLE PNF.sql, CREATE TABLE TRAYECTO.sql, CREATE TABLE UBICACIONES.sql), ESQUEMA (LÍNEA DE TIEMPO), and MYSQL (localhost, admdb). The 'admbd' folder under localhost is expanded, showing sub-tables like periodo_academico, pnf, titulo_grado, trayecto, ubicaciones, construirfuturoc, db_tiendavirtual, information_schema, and mysql.

Ilustración 6. Tabla-entidad "especialidad"

The screenshot shows a Visual Studio Code interface with several tabs open, all related to MySQL database schema creation:

- CREATE TABLE TITULO_GRADO.sql
- CREATE TABLE ESPECIALIDAD.sql
- CREATE TABLE ESPECIALIDAD.sql (selected)

The selected tab contains the following SQL code:

```
1 CREATE TABLE admdb.especialidad(
2     id_especialidad INT AUTO_INCREMENT NOT NULL,
3     especialidad VARCHAR(55) NOT NULL,
4     PRIMARY KEY(id_especialidad)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The Explorer sidebar on the right shows the following structure:

- ADMBD
 - > .vscode
 - > CREAR BASE DE DATO.sql
 - CREATE TABLE ESPECIALIDAD.sql (selected)
 - CREATE TABLE PERIODO_ACADEMICO.sql
 - CREATE TABLE PNF.sql
 - CREATE TABLE TITULO_GRADO.sql
 - CREATE TABLE TRAYECTO.sql
 - CREATE TABLE UBICACIONES.sql
- ESQUEMA
- LÍNEA DE TIEMPO
- MYSQL
 - localhost
 - > admdb
 - > periodo_academico
 - > pnf
 - > titulo_grado
 - > trayecto
 - > ubicaciones
 - > construirfuturoc
 - > db_tiendavirtual
 - > information_schema
 - > mysql
 - > performance_schema

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 7. Tabla-entidad "profesores"

The screenshot shows a Visual Studio Code interface with several tabs open. The active tab is 'CREATE TABLE PROFESORES.sql'. The code in this tab creates a table named 'admbd.profesores' with columns: 'id_profesor' (INT, AUTO_INCREMENT, NOT NULL), 'key_titulo' (INT, NOT NULL), 'key_especialidad' (INT, NOT NULL), and a primary key constraint on 'id_profesor'. The engine is set to InnoDB, default charset is utf8, and collation is utf8_general_ci.

The Explorer sidebar on the right lists various database objects:

- ADMDB:
 - > .vscode
 - > CREAR BASE DE DATO.sql
 - > CREATE TABLE ESPECIALIDAD.D.sql
 - > CREATE TABLE PERIODO ACADEMICO.sql
 - > CREATE TABLE PNF.sql
 - > CREATE TABLE PROFESORES.sql
 - > CREATE TABLE TITULO GRADO.sql
 - > CREATE TABLE TRAYECTO.sql
 - > CREATF TARIFURACIONES.sal
- ESQUEMA
- LÍNEA DE TIEMPO
- MYSQL
 - > localhost
 - > admbd
 - > especialidad
 - > periodo_academico
 - > pnf
 - > titulo_grado
 - > trayecto
 - > ubicaciones
 - > construirfuturoc
 - > db.tiendavirtual
 - > information_schema
 - > mysql

Ilustración 8. Tabla-entidad "maya_curricular"

```
CREATE TABLE MAYA CURRICULAR.sql x
CREATE TABLE MAYA CURRICULAR.sql
1 CREATE TABLE admbd.maya_curricular(
2     id_maya INT AUTO_INCREMENT NOT NULL,
3     maya_curricular VARCHAR(100) NOT NULL,
4     fech_creacion datetime NOT NULL DEFAULT current_timestamp(),
5     key_pnf INT NOT NULL,
6     PRIMARY KEY(id_maya)
7 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

MySQL

```
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
```

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 9. Tabla-entidad "materias"

The screenshot shows the Visual Studio Code interface with the 'CREATE TABLE MATERIAS.sql' file open in the editor. The code defines a table 'admbd.materias' with columns: id_materia (INT AUTO_INCREMENT NOT NULL), cod_materia (VARCHAR(10) NOT NULL), key_trayecto (INT NOT NULL), key_maya (INT NOT NULL), and a primary key (id_materia). The engine is set to InnoDB, default charset to utf8, and collate to utf8_general_ci. The right side of the screen displays the MySQL Explorer, showing the database structure including tables like especialidad, maya_curricular, periodo_academico, prf, profesores, titulo_grado, trayecto, ubicaciones, and construirfuturo. The bottom status bar indicates the query was executed successfully on MySQL at line 4, column 31.

```
CREATE TABLE MATERIAS(
    id_materia INT AUTO_INCREMENT NOT NULL,
    cod_materia VARCHAR(10) NOT NULL,
    key_trayecto INT NOT NULL,
    key_maya INT NOT NULL,
    PRIMARY KEY(id_materia)
)
ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

Ilustración 10. Tabla-entidad "materia_profesor"

The screenshot shows the Visual Studio Code interface with the 'CREATE TABLE MATERIA PROFESOR.sql' file open in the editor. The code defines a table 'admbd.materia_profesor' with columns: id_materia_profesor (INT AUTO_INCREMENT NOT NULL), key_profesor (INT NOT NULL), key_materia (INT NOT NULL), and key_periodo_academico (INT NOT NULL), with a primary key (id_materia_profesor). The engine is set to InnoDB, default charset to utf8, and collate to utf8_general_ci. The right side of the screen displays the MySQL Explorer, showing the database structure including tables like especialidad, maya_curricular, periodo_academico, prf, profesores, titulo_grado, trayecto, ubicaciones, and construirfuturo. The bottom status bar indicates the query was executed successfully on MySQL at line 5, column 40.

```
CREATE TABLE MATERIA PROFESOR(
    id_materia_profesor INT AUTO_INCREMENT NOT NULL,
    key_profesor INT NOT NULL,
    key_materia INT NOT NULL,
    key_periodo_academico INT NOT NULL,
    PRIMARY KEY(id_materia_profesor)
)
ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 11. Tabla-entidad "sede"

The screenshot shows the Visual Studio Code interface with two tabs open: 'CREATE TABLE MATERIA PROFESOR.sql' and 'CREATE TABLE SEDE.sql'. The 'CREATE TABLE SEDE.sql' tab contains the following SQL code:

```
1 CREATE TABLE admdb.sede(
2     id_sede INT AUTO_INCREMENT NOT NULL,
3     sede VARCHAR(255) NOT NULL,
4     PRIMARY KEY(id_sede)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The right side of the screen shows the 'CONTROL DE CÓDIGO FUENTE' (Source Control) panel, which displays a Git repository named 'AdmBD' with a single branch 'main'. A confirmation message 'Mensaje (Ctrl+Enter para confirmar en "main")' is shown, with a '✓ Confiración' button.

Ilustración 12. Tabla-entidad "usuario"

The screenshot shows the Visual Studio Code interface with two tabs open: 'CREATE TABLE SEDE.sql' and 'CREATE TABLE USUARIOS.sql'. The 'CREATE TABLE USUARIOS.sql' tab contains the following SQL code:

```
1 CREATE TABLE admdb.usuario(
2     id_usuario INT AUTO_INCREMENT NOT NULL,
3     key_sede INT NOT NULL,
4     PRIMARY KEY(id_usuario)
5 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The right side of the screen shows the 'EXPLORADOR' (Explorer) panel, which displays a file tree for the 'ADMB' project. The 'USUARIOS.sql' file is selected. The 'MySQL' section of the explorer shows a database named 'admb' containing tables such as 'especialidad', 'materias', 'materia_profesor', 'maya_curricular', 'periodo_academico', 'pnf', 'profesores', 'sede', 'titulo_grado', and 'trabajo'.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 13. Tabla-entidad "espacios"

The screenshot shows the Visual Studio Code interface with two tabs open: 'CREATE TABLE USUARIOS.sql' and 'CREATE TABLE ESPACIOS.sql'. The 'CREATE TABLE ESPACIOS.sql' tab contains the following SQL code:

```
1 CREATE TABLE admdb.espacios(
2     id_espacio INT AUTO_INCREMENT NOT NULL,
3     key_sede INT NOT NULL,
4     key_ubicaciones INT NOT NULL,
5     PRIMARY KEY(id_espacio)
6 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The right side of the screen shows the 'EXPLORADOR' sidebar with a tree view of database objects under 'ADMDB' and 'MySQL'. Under 'ADMDB', 'CREATE TABLE ESPACIOS.sql' is selected. Under 'MySQL', 'espacios' is selected. The bottom status bar shows 'Lín. 6, col. 64 Espacios: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell'.

Ilustración 14. Tabla-entidad "horarios"

The screenshot shows the Visual Studio Code interface with two tabs open: 'CREATE TABLE ESPACIOS.sql' and 'CREATE TABLE HORARIO.sql'. The 'CREATE TABLE HORARIO.sql' tab contains the following SQL code:

```
1 CREATE TABLE admdb.horario(
2     id_horario INT AUTO_INCREMENT NOT NULL,
3     key_materia INT NOT NULL,
4     key_espacios INT NOT NULL,
5     key_profesor INT NOT NULL,
6     hora_inicio time NOT NULL,
7     hora_fin time NOT NULL,
8     PRIMARY KEY(id_horario)
9 ) ENGINE = InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci;
```

The right side of the screen shows the 'EXPLORADOR' sidebar with a tree view of database objects under 'ADMDB' and 'MySQL'. Under 'ADMDB', 'CREATE TABLE HORARIO.sql' is selected. Under 'MySQL', 'horario' is selected. The bottom status bar shows 'Lín. 6, col. 31 Espacios: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell'.

2. Comentarios y documentación (**opcional**): Se agregaron comentarios descriptivos para facilitar la comprensión y el mantenimiento del código. Se recomienda *documentar* cualquier aspecto relevante sobre el diseño y funcionamiento de la base de datos, para este caso se ha realizado este proceso haciendo uso del repositorio (*guardado en git*).

4.4. Validación del script SQL

- Verificación de la sintaxis: Utiliza las herramientas proporcionadas por Visual Studio Code o por la extensión específica para verificar la sintaxis del script SQL y corregir cualquier error identificado.
- Ejecución de consultas: Utiliza Visual Studio Code para ejecutar consultas de prueba en la base de datos y asegurarte de que el script SQL funcione correctamente. Verifica que los datos se inserten correctamente, las consultas retornen los resultados esperados y las reglas de negocio se cumplan.

The screenshot shows the MySQL Workbench interface. The top bar has tabs for 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL'. The 'SALIDA' tab is active, displaying the following text:

```
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"",
"protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
```

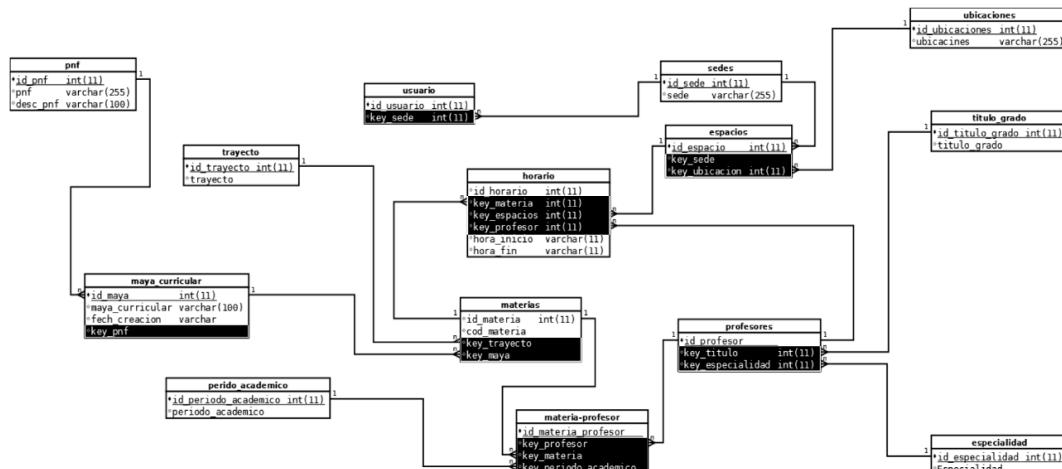
The right side of the interface shows a tree view of database objects under the 'MySQL' connection. The tree includes nodes for 'especialidad', 'materias', 'materia_profesor', 'maya_curricular', 'periodo_academico', 'pnf', 'profesores', 'sede', and 'titulo_grado'. Below the tree, there are several status indicators and icons.

- Optimización de consultas: Si es necesario, analiza y optimiza las consultas para mejorar su rendimiento. Utiliza las herramientas de análisis de consultas y ajusta la estructura de las tablas, los índices o las consultas mismas.

4.5. Definición de las relaciones entre las tablas

- Se analiza las interacciones entre las entidades y determina las relaciones que deben establecerse. Identifica las claves externas que se utilizarán para vincular las tablas.
- Se utiliza la cláusula "FOREIGN KEY" para establecer las claves externas en las tablas relacionadas. Define las restricciones referenciales para asegurar la integridad de los datos.
- Se asocia las tablas relacionadas utilizando las claves externas correspondientes. Asegúrate de que las claves externas apunten a las claves primarias correspondientes en las tablas relacionadas.

Ilustración 15. Identificación de las Foreign Key



4.6. Desarrollo de las claves foráneas (Foreign Keys)

1. Se analiza las relaciones establecidas entre las tablas y determina las entidades que actuarán como tablas padre y tablas hijas.
2. Se utilizó la sintaxis SQL “ALTER TABLE” para alterar las tablas correspondientes a cada entidad. Se utiliza la cláusula "FOREIGN KEY" para definir las claves foráneas en las tablas hijas especificando la columna que actuará como clave foránea. Finalmente, la cláusula "REFERENCES" para definir la tabla padre y la columna a la que hará referencia.

Ilustración 16. Relación maya_curricular - pnf

The screenshot shows a Visual Studio Code interface with a MySQL extension. The left pane displays the code for creating a foreign key:

```
ALTER TABLE admbd.maya_curricular ADD FOREIGN KEY (key_pnf) REFERENCES pnf (id_pnf);
```

The right pane shows the MySQL Explorer, which lists various database objects:

- ADMDB
 - .vscode
 - ALTER TABLE MAYA_CURRICULAR.sql
 - CREAR BASE DE DATO.sql
 - CREATE TABLE ESPACIOS.sql
 - CREATE TABLE ESPECIALIDAD.sql
 - CREATE TABLE HORARIO.sql
 - CREATE TABLE MATERIA PROFESOR.sql
 - CREATE TABLE MATERIAS.sql
 - CREATE TABLE TARI F MAYA CURRICULAR.sql
- LÍNEA DE TIEMPO
- MYSQL
 - localhost
 - admbd
 - espacios
 - especialidad
 - materias
 - materia_profesor
 - maya_curricular
 - periodo_academico
 - pnf
 - profesores
 - sede
 - titulo_grado

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 17. Relación usuario - sede

The screenshot shows a Visual Studio Code interface with the title bar "ALTER TABLE USUARIO.sql - AdmBD - Visual Studio Code". The Explorer sidebar on the right lists database objects under "ADMBD" and "MySQL". In the center, there are two tabs: "ALTER TABLE MAYA_CURRICULAR.sql" and "ALTER TABLE USUARIO.sql". The "ALTER TABLE USUARIO.sql" tab contains the following SQL code:

```
1 ALTER TABLE admdb.usuario ADD FOREIGN KEY (key_sede) REFERENCES sede (id_sede);
```

Below the tabs, the "PROBLEMAS", "SALIDA", and "CONSOLA DE DEPURACIÓN" sections are visible. The "CONSOLA DE DEPURACIÓN" section shows the execution of the query and its results:

```
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"&Records: 0
Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
```

The status bar at the bottom indicates "Lín. 1, col. 80 Espacios: 4 UTT-8 CRLF MySQL Go Live Blackbox Spell".

Ilustración 18. Relación espacios - sedes

The screenshot shows a Visual Studio Code interface with the title bar "ALTER TABLE ESPACIOS-SEDE.sql - AdmBD - Visual Studio Code". The Explorer sidebar on the right lists database objects under "ADMBD" and "MySQL". In the center, there are two tabs: "ALTER TABLE USUARIO-SEDE.sql" and "ALTER TABLE ESPACIOS-SEDE.sql". The "ALTER TABLE ESPACIOS-SEDE.sql" tab contains the following SQL code:

```
1 ALTER TABLE admdb.espacios ADD FOREIGN KEY (key_sede) REFERENCES sede (id_sede);
```

Below the tabs, the "PROBLEMAS", "SALIDA", and "CONSOLA DE DEPURACIÓN" sections are visible. The "CONSOLA DE DEPURACIÓN" section shows the execution of the query and its results:

```
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"&Records: 0
Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
```

The status bar at the bottom indicates "Lín. 1, col. 81 Espacios: 4 UTT-8 CRLF MySQL Go Live Blackbox Spell".

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 19. Relación espacios - ubicaciones

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE ESPACIOS-SEDE.sql' and 'ALTER TABLE ESPACIOS-UBICACIONES.sql'. The 'ALTER TABLE ESPACIOS-UBICACIONES.sql' tab contains the following SQL code:

```
1 ALTER TABLE admdb.espacios ADD FOREIGN KEY (key_ubicaciones)
2 REFERENCES ubicaciones (id_ubicaciones);
```

The 'EXPLORADOR' sidebar on the right shows a tree view of the database schema under 'ADMBD'. Under the 'admbd' database, there are tables: espacios, especialidad, horario, materias, materia_profesor, maya_curricular, periodo_academico, pf, profesores, and sede.

Ilustración 20. Relación profesores - titulo_grado

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE ESPACIOS-UBICACIONES.sql' and 'ALTER TABLE PROFESORES-TITULO_GRADO.sql'. The 'ALTER TABLE PROFESORES-TITULO_GRADO.sql' tab contains the following SQL code:

```
1 ALTER TABLE admdb.profesores ADD FOREIGN KEY (key_titulo)
2 REFERENCES titulo_grado (id_titulo_grado);
```

The 'EXPLORADOR' sidebar on the right shows a tree view of the database schema under 'ADMBD'. Under the 'admbd' database, there are tables: espacios, especialidad, horario, materias, materia_profesor, maya_curricular, periodo_academico, pf, profesores, and sede. A confirmation dialog box is visible in the center of the screen.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 21. Relación profesores - especialidad

The screenshot shows the Visual Studio Code interface with two SQL files open in tabs:

- ALTER TABLE PROFESORES-TITULO_GRADO.sql
- ALTER TABLE PROFESORES-TITULO_ESPECIALIDAD.sql

The second file contains the following code:

```
1 ALTER TABLE admbd.profesores ADD FOREIGN KEY (key_especialidad)
2 REFERENCES especialidad (id_especialidad);
```

The Explorer sidebar on the right shows the database schema structure under the 'ADMBD' connection, including tables like espacios, especialidad, horario, materias, materia_profesor, maya_curricular, periodo_academico, pf, profesores, and sede.

Ilustración 22. Relación materias - trayecto

The screenshot shows the Visual Studio Code interface with two SQL files open in tabs:

- ALTER TABLE PROFESORES-TITULO_ESPECIALIDAD.sql
- ALTER TABLE MATERIAS-TRAYECTO.sql

The second file contains the following code:

```
1 ALTER TABLE admbd.materias ADD FOREIGN KEY (key_trayecto)
2 REFERENCES trayecto (id_trayecto);
```

The Explorer sidebar on the right shows the database schema structure under the 'ADMBD' connection, including tables like espacios, especialidad, horario, materias, materia_profesor, maya_curricular, periodo_academico, pf, profesores, and sede.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 23. Relación *materias - maya_curricular*

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE MATERIAS-TRAYECTO.sql' and 'ALTER TABLE MATERIAS-MAYA_CURRICULAR.sql'. The second tab contains the following SQL code:

```
1 ALTER TABLE admbd.materias ADD FOREIGN KEY (key_maya)
2 REFERENCES maya_curricular (id_maya);
```

The right side of the screen shows the 'CONTROL DE CÓDIGO FUENTE' (Source Control) panel, which displays a repository named 'AdmBD Git' with a single branch 'main'. A confirmation message 'Mensaje (Ctrl+Enter para confirmar en ...)' is shown, with 'Confirmación' selected. The bottom status bar indicates the file is 'main' and the connection is 'localhost'.

Ilustración 24. Relación *materia_profesor - profesores*

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE MATERIAS-MAYA_CURRICULAR.sql' and 'ALTER TABLE MATERIA_PROFESOR-PROFESORES.sql'. The second tab contains the following SQL code:

```
1 ALTER TABLE admbd.materia_profesor ADD FOREIGN KEY (key_profesor)
2 REFERENCES profesores (id_profesor);
```

The right side of the screen shows the 'EXPLORADOR' (Explorer) panel, which displays a database structure for 'ADMBD'. Under 'ADMBD', there are several tables listed: 'espacios', 'especialidad', 'horario', 'materias', 'materia_profesor', 'maya.curricular', 'periodo_academico', 'pnf', 'profesores', and 'sede'. The 'localhost' section also lists the same tables. The bottom status bar indicates the file is 'main' and the connection is 'localhost'.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 25. Relación *materia_profesor - materias*

The screenshot shows a Visual Studio Code interface with an open file named 'ALTER TABLE MATERIA_PROFESOR-MATERIAS.sql'. The code contains the following SQL command:

```
1 ALTER TABLE admbd.materia_profesor ADD FOREIGN KEY (key_materia)
2 REFERENCES materias (id_materia);
```

The 'EXPLORADOR' sidebar on the right shows a tree view of the database schema under the 'ADMBD' node, including tables like 'espacios', 'especialidad', 'horario', 'materias', 'materia_profesor', 'maya_curricular', 'periodo_academico', 'pnf', 'profesores', and 'sede'.

Ilustración 26. Relación *materia_profesor - periodo_academico*

The screenshot shows a Visual Studio Code interface with two open files: 'ALTER TABLE MATERIA_PROFESOR-MATERIAS.sql' and 'ALTER TABLE MATERIA_PROFESOR-PERIODO_ACADEMICO.sql'. The second file contains the following SQL command:

```
1 ALTER TABLE admbd.materia_profesor ADD FOREIGN KEY (key_período_academico)
2 REFERENCES periodo_academico (id_período_academico);
```

The 'EXPLORADOR' sidebar on the right shows a tree view of the database schema under the 'ADMBD' node, including tables like 'espacios', 'especialidad', 'horario', 'materias', 'materia_profesor', 'maya_curricular', 'periodo_academico', 'pnf', 'profesores', and 'sede'.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 27. Relación horario - materias

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE MATERIA_PROFESOR-PERIODO_ACADEMICO.sql' and 'ALTER TABLE HORARIO-MATERIAS.sql'. The 'ALTER TABLE HORARIO-MATERIAS.sql' tab contains the following SQL code:

```
1 ALTER TABLE admbd.horario ADD FOREIGN KEY (key_materia)
2 REFERENCES materias (id_materia);
```

The 'PROBLEMAS' (Problems) tab shows no errors. The 'SALIDA' (Output) tab displays MySQL command-line output indicating successful execution of the ALTER TABLE statements. The 'CONSOLA DE DEPURACIÓN' (Debug Console) and 'TERMINAL' tabs are empty. The 'EXPLORADOR' (Explorer) sidebar on the right shows the database schema structure under 'ADMDB' and 'localhost'.

Ilustración 28. Relación horario - espacios

The screenshot shows a Visual Studio Code interface with two tabs open: 'ALTER TABLE HORARIO-MATERIAS.sql' and 'ALTER TABLE HORARIO-ESPACIOS.sql'. The 'ALTER TABLE HORARIO-ESPACIOS.sql' tab contains the following SQL code:

```
1 ALTER TABLE admbd.horario ADD FOREIGN KEY (key_espacios)
2 REFERENCES espacios (id_espacio);
```

The 'PROBLEMAS' (Problems) tab shows no errors. The 'SALIDA' (Output) tab displays MySQL command-line output indicating successful execution of the ALTER TABLE statements. The 'CONSOLA DE DEPURACIÓN' (Debug Console) and 'TERMINAL' tabs are empty. The 'EXPLORADOR' (Explorer) sidebar on the right shows the database schema structure under 'ADMDB' and 'localhost'.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

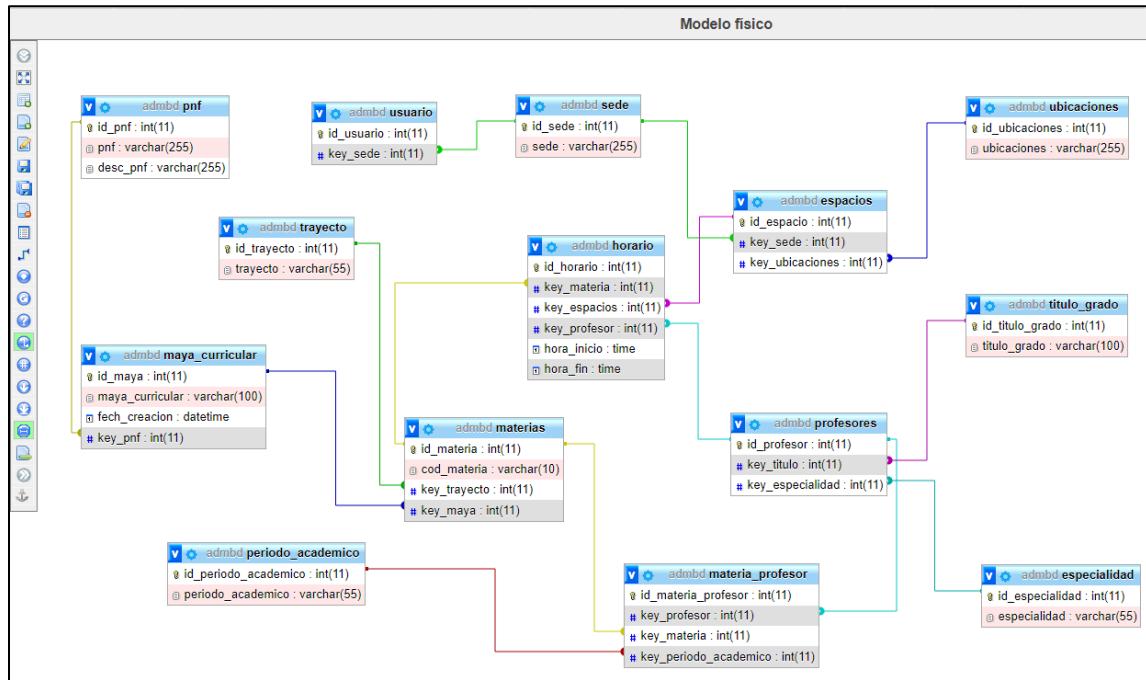
Ilustración 29. Relación horario - profesores

```
ALTER TABLE HORARIO-ESPACIOS.sql ALTER TABLE HORARIO-PROFESORES.sql
1 ALTER TABLE admbd.horario ADD FOREIGN KEY (key_profesor)
2 REFERENCES profesores (id_profesor);
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL MySQL LIN. 2, col. 37 Espacios:4 UTF-8 CRLF MySQL Go Live Blackbox Spell

```
{"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"&Records: 0 Duplicates: 0 Warnings: 0","protocol41":true,"changedRows":0} [Done] Finished MySQL query. [Start] Executing MySQL query... {"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"&Records: 0 Duplicates: 0 Warnings: 0","protocol41":true,"changedRows":0} [Done] Finished MySQL query. [Start] Executing MySQL query... {"fieldCount":0,"affectedRows":0,"insertId":0,"serverStatus":2,"warningCount":0,"message":"&Records: 0 Duplicates: 0 Warnings: 0","protocol41":true,"changedRows":0} [Done] Finished MySQL query.
```

Ilustración 30. Resultado final de las relaciones realizadas



5. Configuración de la Replicación de la Base de Datos

En esta sección, se detallará el proceso de configuración de la replicación de la base de datos. La replicación es fundamental para garantizar la disponibilidad y la redundancia de los datos en entornos críticos. A continuación, se presentan los pasos necesarios para configurar la replicación utilizando máquinas virtuales.

5.1. Configuración de las Máquinas Master y Slave

Preparación de la máquina virtual: Asegúrate de que la máquina destinada a la replicación esté configurada y funcionando correctamente, tal como se describió en la sección 2 y 3.

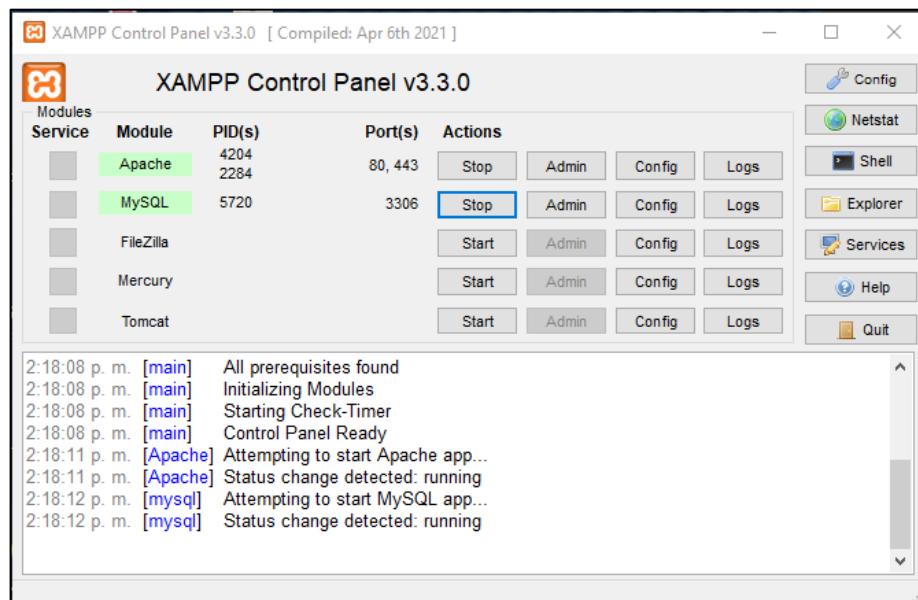
Asignación de roles: Identifica la maquina local como el servidor principal (master) y la máquina virtual como el servidor secundario (Slave). El servidor principal contendrá la base de datos principal y el servidor secundario se sincronizará con los cambios realizados en el servidor principal.

Configuración de la red: Verifica que las máquinas virtuales se puedan comunicar entre sí en la red. Configura la red de las máquinas virtuales para permitir la replicación, asegurándote de que los puertos necesarios estén abiertos y que las direcciones IP estén correctamente configuradas.

5.2. Configuración del Sistema Gestor de Base de Datos

Configuración del servidor principal: *Accede al servidor principal y realiza los siguientes pasos:*

1. Se abre el sistema gestor de base de datos (por ejemplo, MySQL) y se asegura de que esté funcionando correctamente.



Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

2. Se verifica la base de datos creada y se asegura de que esté funcionando correctamente en el servidor principal. Esta verificación incluye que las tablas, los campos y las relaciones estén correctamente definidos.

3. Antes de iniciar la replicación, en la interfaz de phpMyAdmin se realiza una copia de seguridad de la base de datos en el servidor principal. Esto asegurará que tengas una copia de seguridad de los datos en caso de cualquier problema durante el proceso de replicación.

4. Se abre el archivo de configuración en el servidor principal y realiza las siguientes configuraciones:

- ✓ Buscamos la configuración por default que contiene la siguiente línea

```
# Replication Master Server (default)
# binary logging is required for replication
# log-bin deactivated by default since XAMPP 1.4.11
#log-bin=mysql-bin
```

```
# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id = 1
```

- ✓ Reemplazamos las configuraciones anteriores por la siguiente línea:

```
# Replication Master Server (default)
# binary logging is required for replication
# log-bin deactivated by default since XAMPP 1.4.11
```

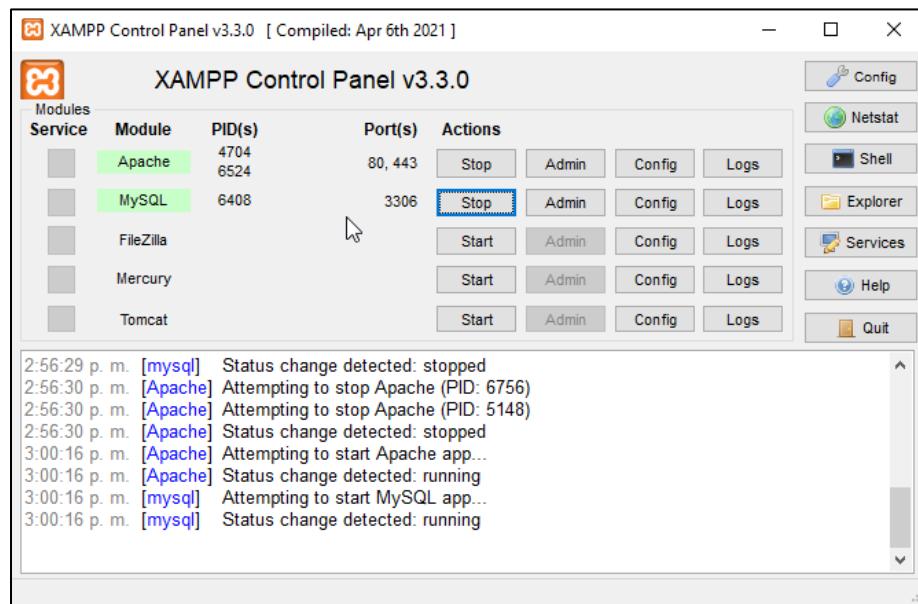
```
log-bin=mysql-bin
binlog_do_db = admbd
```

```
# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id    =1
```

- ✓ Se guarda los cambios y reinicia el servidor de base de datos principal.

Configuración del servidor secundario: *Accede al servidor secundario y realiza los siguientes pasos:*

1. Abre el sistema gestor de base de datos y asegúrate de que esté funcionando correctamente.

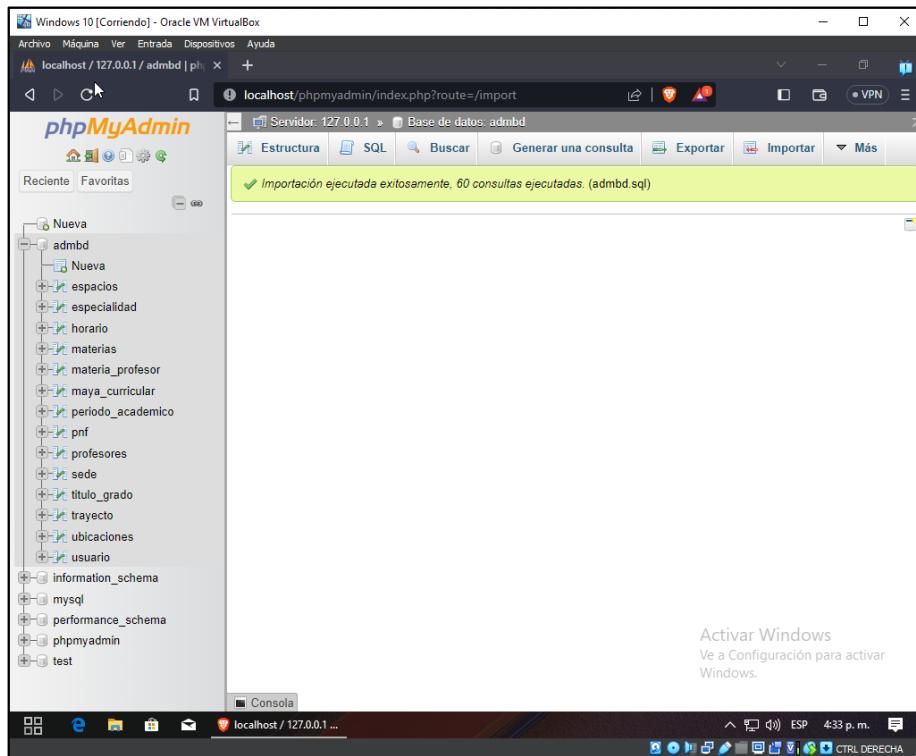


2. En la interfaz de phpMyAdmin, crea una nueva base de datos vacía que se utilizará para almacenar la réplica de la base de datos del servidor principal. Se importa la copia de seguridad de la base de datos creada. Esto asegurará que la base de datos en el servidor secundario sea una réplica exacta de la base de datos en el servidor principal.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>



3. Se abre el archivo de configuración en el servidor secundario y realiza las siguientes configuraciones:

- ✓ Buscamos la configuración por default que contiene la siguiente línea

```
# Replication Master Server (default)
# binary logging is required for replication
# log-bin deactivated by default since XAMPP 1.4.11
#log-bin=mysql-bin

# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id =1
```

- ✓ Comentamos la siguiente línea marcada en negrita:

```
# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
#server-id =1
```

- ✓ Ahora buscamos la configuración por default que contiene la siguiente línea

```
# Replication Slave (comment out master section to use this)
#
# To configure this host as a replication slave, you can choose between
# two methods :
#
# 1) Use the CHANGE MASTER TO command (fully described in our manual) -
#    the syntax is:
#
#    CHANGE MASTER TO MASTER_HOST=<host>,MASTER_PORT=<port>,
#    MASTER_USER=<user>,MASTER_PASSWORD=<password>;
#
#    where you replace <host>, <user>, <password> by quoted strings and
#    <port> by the master's port number (3306 by default).
#
# Example:
#
#    CHANGE MASTER TO MASTER_HOST='125.564.12.1',MASTER_PORT=3306,
#    MASTER_USER='joe',MASTER_PASSWORD='secret';
#
# OR
#
# 2) Set the variables below. However, in case you choose this method, then
#    start replication for the first time (even unsuccessfully, for example
#    if you mistyped the password in master-password and the slave fails to
#    connect), the slave will create a master.info file, and any later
#    change in this file to the variables' values below will be ignored and
#    overridden by the content of the master.info file, unless you shutdown
#    the slave server, delete master.info and restart the slaver server.
#    For that reason, you may want to leave the lines below untouched
#    (commented) and instead use CHANGE MASTER TO (see above)
#
# required unique id between 2 and 2^32 - 1
# (and different from the master)
# defaults to 2 if master-host is set
# but will not function as a slave if omitted
#server-id      = 2
```

- ✓ Retiramos el # y agregamos binlog_do_db = admdb en la siguiente línea marcada en negrita:

```
# required unique id between 2 and 2^32 - 1
# (and different from the master)
# defaults to 2 if master-host is set
# but will not function as a slave if omitted
server-id      = 2
binlog_do_db = admdb
```

4. Se guardan los cambios y reinicia el servidor de base de datos secundaria.

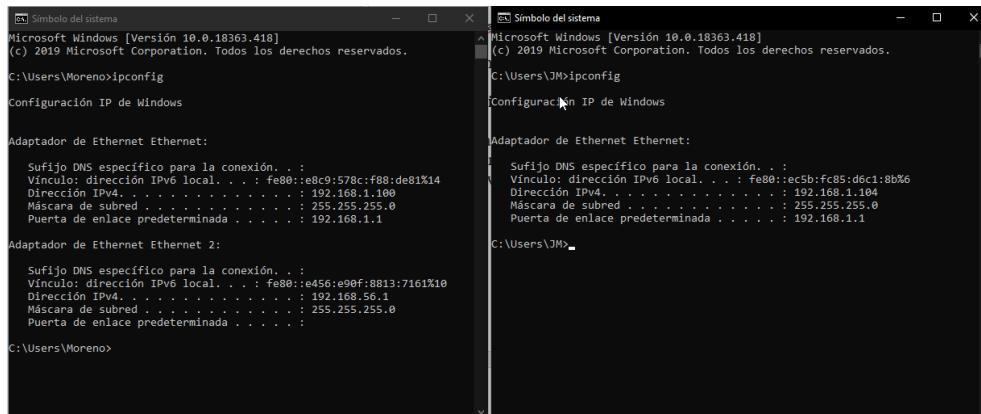
Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

5.3. Inicio de la Replicación

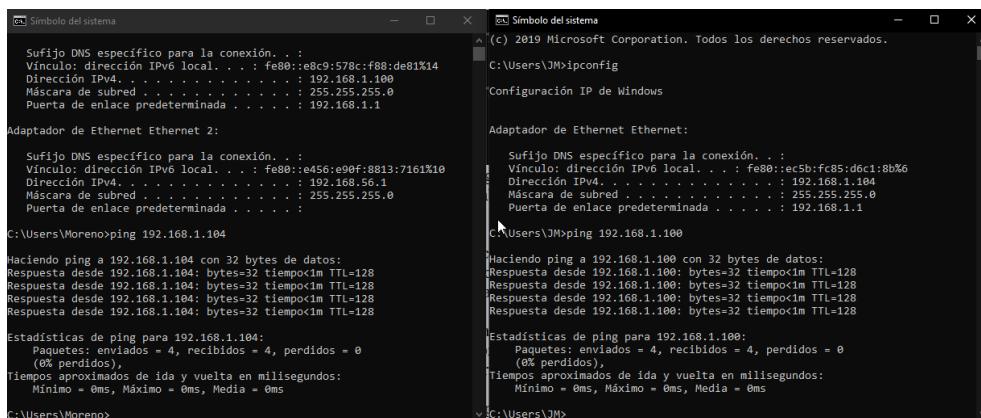
1. Se inicia la replicación en este apartado consultando las ip de las máquinas master y Slave utilizando el comando “ipconfig” para desplegar la información de la dirección ip de ambas maquinas respectivamente.



```
Administrator: Símbolo del sistema Microsoft Windows [Versión 10.0.18363.418] (c) 2019 Microsoft Corporation. Todos los derechos reservados. C:\Users\Moreno>ipconfig Configuración IP de Windows Adaptador de Ethernet Ethernet: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::e8c9:578c:f88:de81%14 Dirección IPv4. . . . . : 192.168.1.100 Máscara de subred . . . . . : 255.255.255.0 Puerta de enlace predeterminada . . . . . : 192.168.1.1 Adaptador de Ethernet Ethernet 2: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::e456:e90f:8813:7161%10 Dirección IPv4. . . . . : 192.168.56.1 Máscara de subred . . . . . : 255.255.255.0 Puerta de enlace predeterminada . . . . . : C:\Users\Moreno>

Administrator: Símbolo del sistema Microsoft Windows [Versión 10.0.18363.418] (c) 2019 Microsoft Corporation. Todos los derechos reservados. C:\Users\JM>ipconfig Configuración IP de Windows Adaptador de Ethernet Ethernet: Sufijo DNS específico para la conexión. . . : Vínculo: dirección IPv6 local. . . : fe80::ec5b:fc85:d6c1:8b%6 Dirección IPv4. . . . . : 192.168.1.104 Máscara de subred . . . . . : 255.255.255.0 Puerta de enlace predeterminada . . . . . : 192.168.1.1 C:\Users\JM>
```

2. Se realiza un ping en cada una de las maquinas con el objetivo de determinar ambas esté en comunicación y funcionando correctamente.



```
Administrator: Símbolo del sistema Microsoft Windows [Versión 10.0.18363.418] (c) 2019 Microsoft Corporation. Todos los derechos reservados. C:\Users\Moreno>ping 192.168.1.104 Haciendo ping a 192.168.1.104 con 32 bytes de datos: Respuesta desde 192.168.1.104: bytes=32 tiempo<1ms TTL=128 Estadísticas de ping para 192.168.1.104: Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 0ms, Máximo = 0ms, Media = 0ms C:\Users\Moreno>

Administrator: Símbolo del sistema Microsoft Windows [Versión 10.0.18363.418] (c) 2019 Microsoft Corporation. Todos los derechos reservados. C:\Users\JM>ping 192.168.1.100 Haciendo ping a 192.168.1.100 con 32 bytes de datos: Respuesta desde 192.168.1.100: bytes=32 tiempo<1ms TTL=128 Estadísticas de ping para 192.168.1.100: Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (% perdidos), Tiempos aproximados de ida y vuelta en milisegundos: Mínimo = 0ms, Máximo = 0ms, Media = 0ms C:\Users\JM>
```

3. Seleccionamos la base de datos del servidor master y haciendo uso de la opción SQL de phpMyAdmin agregamos la siguiente sentencia y presionamos en continuar.

```
USE admbd;
FLUSH TABLE WITH READ LOCK
```

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 31. Consulta sentencia USE

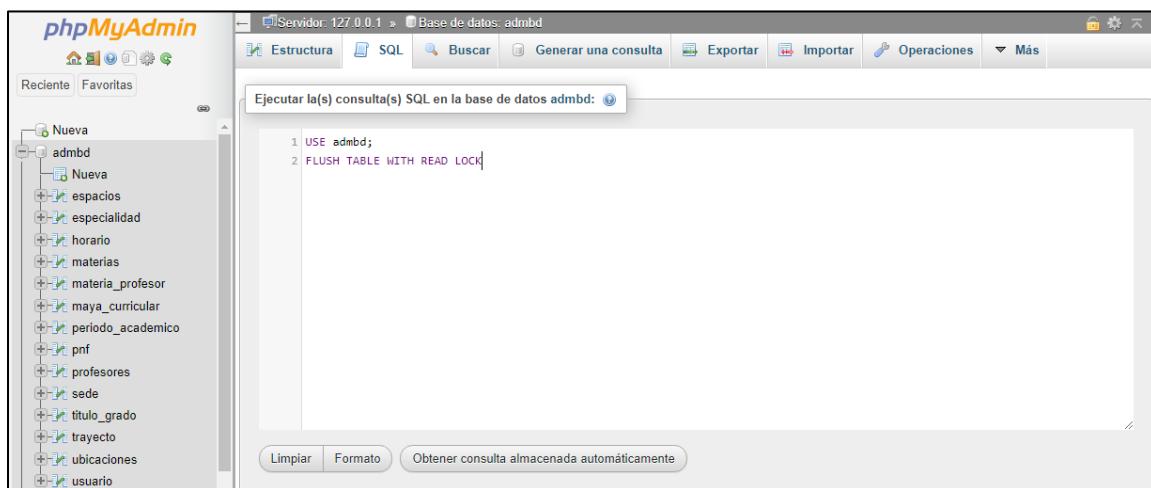
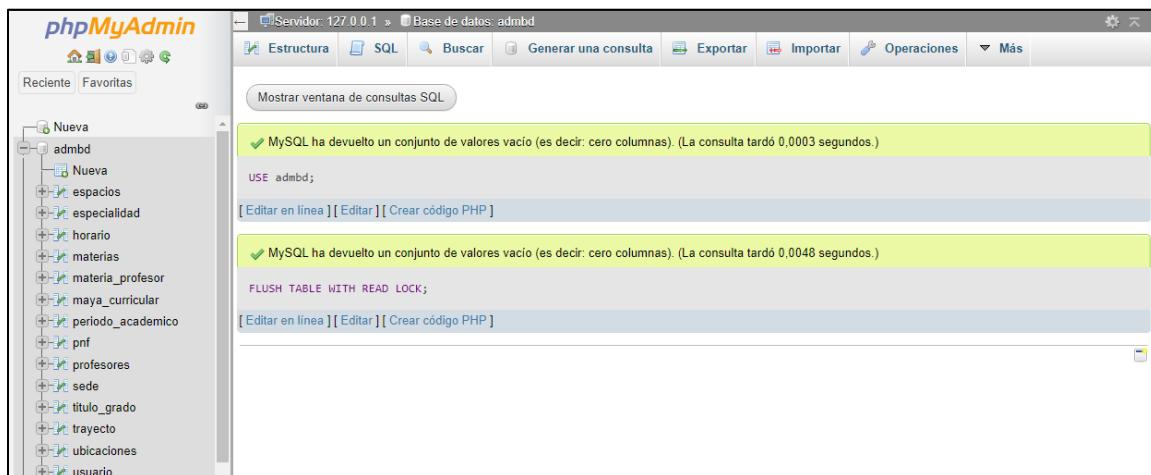
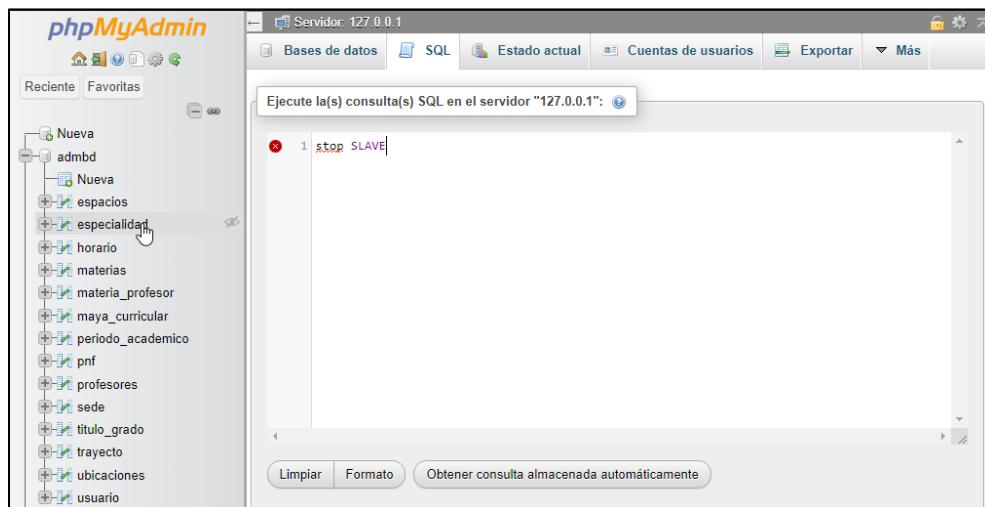
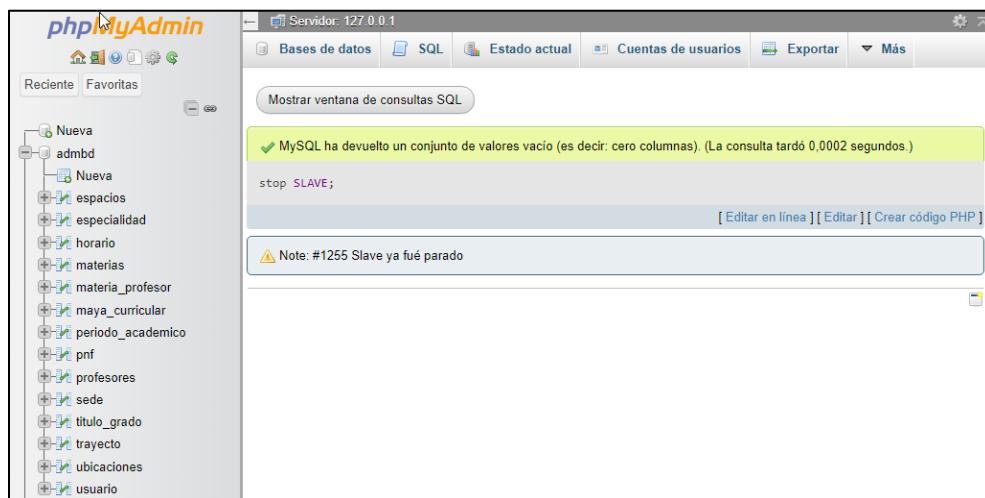


Ilustración 32. Resultado sentencia USE



4. Ahora seleccionamos la base de datos del servidor Slave y haciendo uso de la opción SQL de phpMyAdmin agregamos la siguiente sentencia y presionamos en continuar.

Stop SLAVE

Ilustración 33. Consulta de la sentencia stop SLAVE**Ilustración 34.** Resultado de la sentencia stop SLAVE

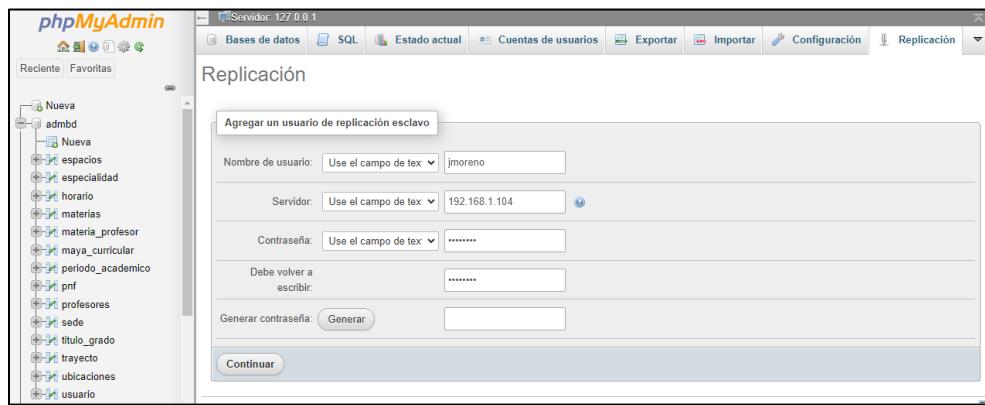
5. Ahora nos dirigimos a la base de datos del servidor principal y haciendo uso de la opción Replicación de phpMyAdmin seleccionamos “Agregar un usuario de replicación esclavo” y llenamos los siguientes datos:
 - ✓ Nombre de usuario: identifica el nombre de acceso a la BD master.
 - ✓ Servidor: ingresamos el ip de la máquina virtual.
 - ✓ Contraseña: clave de acceso que usaremos más adelante.

Finalizamos la operación presionando el botón continuar.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

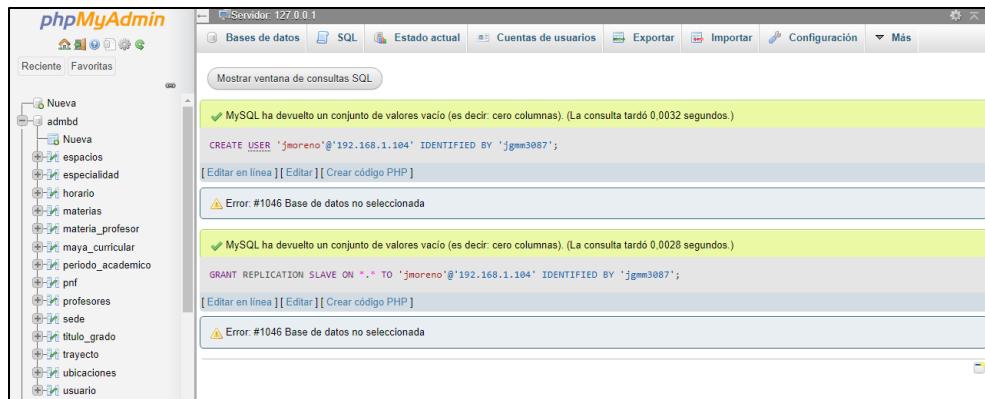


6. El resultado de la operación anterior nos genera un error que puede ser solucionado reemplazando la siguiente sintaxis.

```
CREATE USER 'jmoreno'@'192.168.1.104' USING '****';GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'jmoreno'@'192.168.1.104' REQUIRE NONE;
```

->

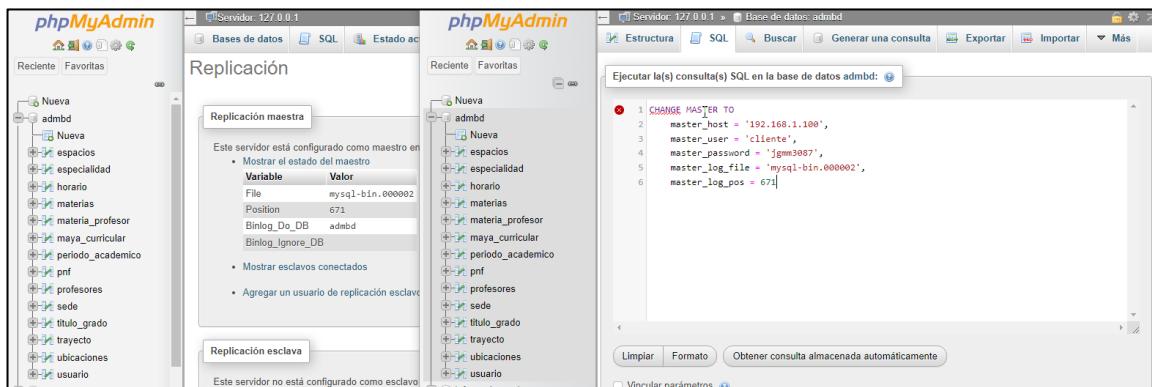
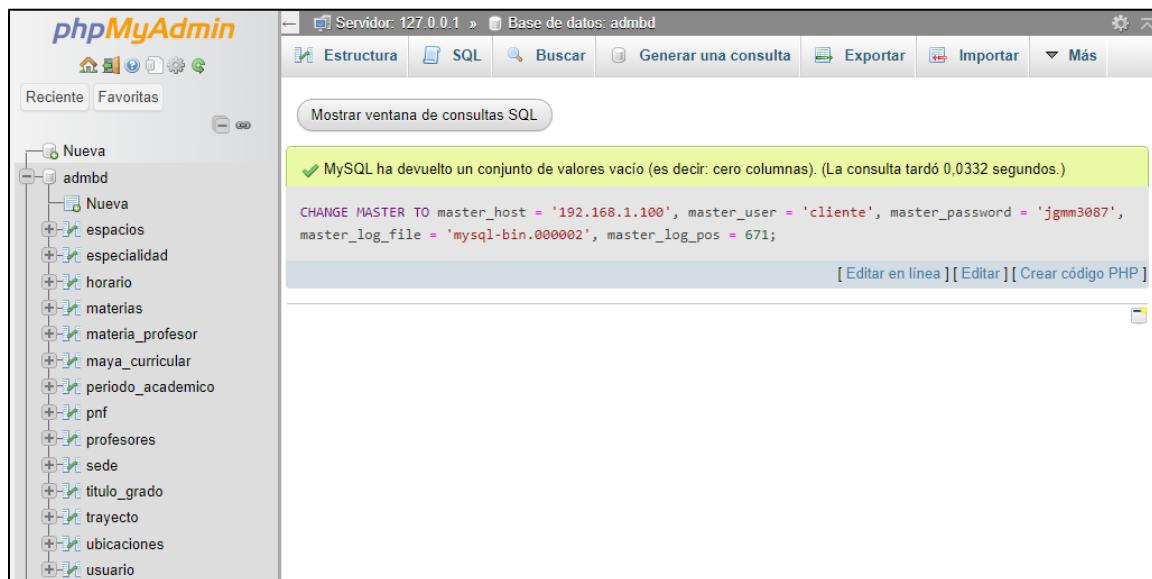
```
CREATE USER 'jmoreno'@'192.168.1.104' IDENTIFIED BY 'jgmm3087';GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'jmoreno'@'192.168.1.104' IDENTIFIED BY 'jgmm3087';
```



7. Luego, en la base de datos del servidor Slave y haciendo uso de la opción SQL de phpMyAdmin agregamos la siguiente sentencia validando que coincida con la información de la opción replicación del servidor master y presionamos en continuar.

CHANGE MASTER TO

```
master_host = '192.168.1.100'          //La ip de la máquina master
master_user = 'jmoreno'                 //El usuario definido en el paso anterior
master_password = 'jgmm3087'            //La contraseña definida en el paso anterior
master_log_file = 'mysql-bin.000002'     //Esta información que proviene del servidor master
master_log_pos = 671                   //Esta información que proviene del servidor master
```

Ilustración 35. Consulta de la sentencia *CHANGE MASTER TO***Ilustración 36.** Resultado de la sentencia *CHANGE MASTER TO*

8. Finalmente, procedemos a iniciar la base de datos en el servidor Slave, ingresamos a la opción de replicación y desplegamos “ver la tabla de estado de réplica” para visualizar los resultados y luego desplegamos la opción “controlar réplica o esclavo” y seleccionamos la opción “Full start” para que inicie automáticamente la replicación entre el master y el Slave.

Ilustración 37. Despliegue del estado de la réplica

The screenshot shows the phpMyAdmin interface for a MySQL database named 'admbd'. The left sidebar lists various databases and their tables. The main panel is titled 'Replicación de la réplica' (Replication Status). It contains two error messages: 'El hilo SQL replica no se está ejecutando!' (The SQL replication thread is not running!) and 'El hilo IO de réplica no se está ejecutando!' (The IO replication thread is not running!). Below the errors, it says: 'El servidor está configurado como réplica en un proceso de replicación. Deseas:' (The server is configured as a replica in a replication process. Do you want to:). A list item 'Ver la tabla de estado de réplica' (View the replication status table) is selected. To the right, there is a table showing replication variables and their values:

Variable	Valor
Slave_IO_State	
Master_Host	192.168.1.100
Master_User	cliente
Master_Port	3306
Connect_Retry	60
Master_Log_File	mysql-bin.000002
Read_Master_Log_Pos	671
Relay_Log_File	mysql-relay-bin.000001
Relay_Log_Pos	4
Relay_Master_Log_File	mysql-bin.000002
Slave_IO_Running	No
Slave_SQL_Running	No
Replicate_Do_DB	

At the bottom right, there is a note: 'Activar Windows' (Activate Windows) with the subtext 'Ve a Configuración para activar Windows.' (Go to Configuration to activate Windows.)

Ilustración 38. Despliegue de controlar réplica

The screenshot shows the phpMyAdmin interface for a MySQL database named 'admbd'. The left sidebar lists various databases and their tables. The main panel is titled 'Estado actual' (Current Status). On the right, there is a table listing replication variables:

Replicate_Wild_Ignore_Table	
Last_Error	0
Skip_Counter	0
Exec_Master_Log_Pos	671
Relay_Log_Space	256
Until_Condition	None
Until_Log_File	
Until_Log_Pos	0
Master_SSL_Allowed	
Master_SSL_CA_File	
Master_SSL_CA_Path	
Master_SSL_Cert	
Master_SSL_Cipher	
Master_SSL_Key	
Seconds_Behind_Master	

Below the table, there is a list of replication control options:

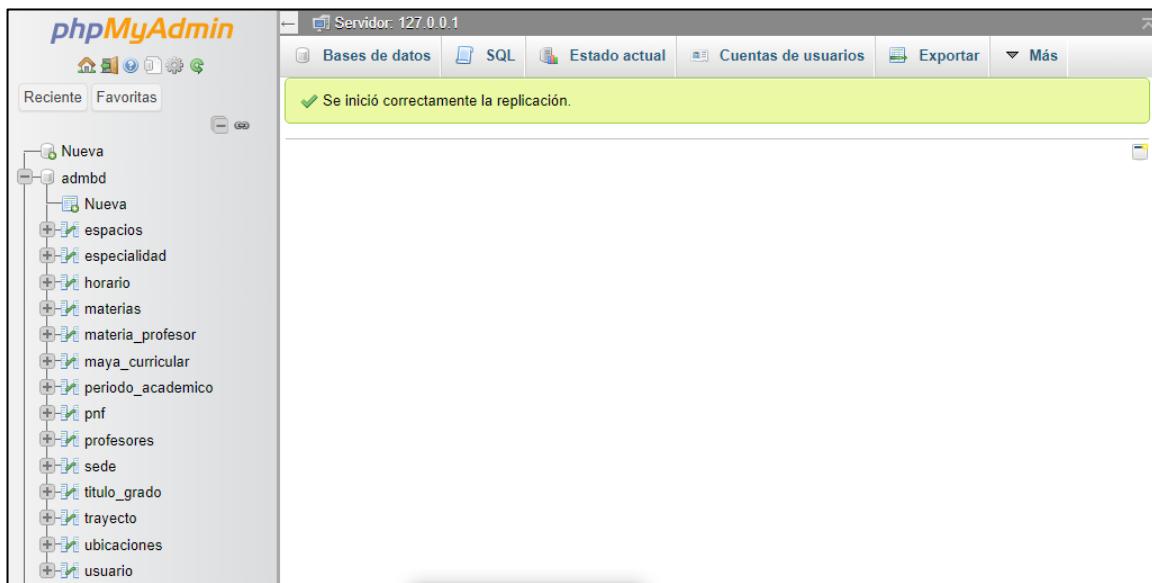
- Controlar réplica:
 - Full start
 - Reiniciar réplica
 - Iniciar hilo SQL sólamente
 - Iniciar hilo IO sólamente
- Manejo de errores:
- Cambiar o reconfigurar el servidor principal

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 39. Resultado de inicio de la replicación



9. Se monitorea que el proceso de replicación realizado en el servidor principal se replique correctamente en el servidor secundario. Asegúrate de que no se produzcan errores y que la sincronización sea exitosa.

Ilustración 40. Verificación Bases de datos (Master)

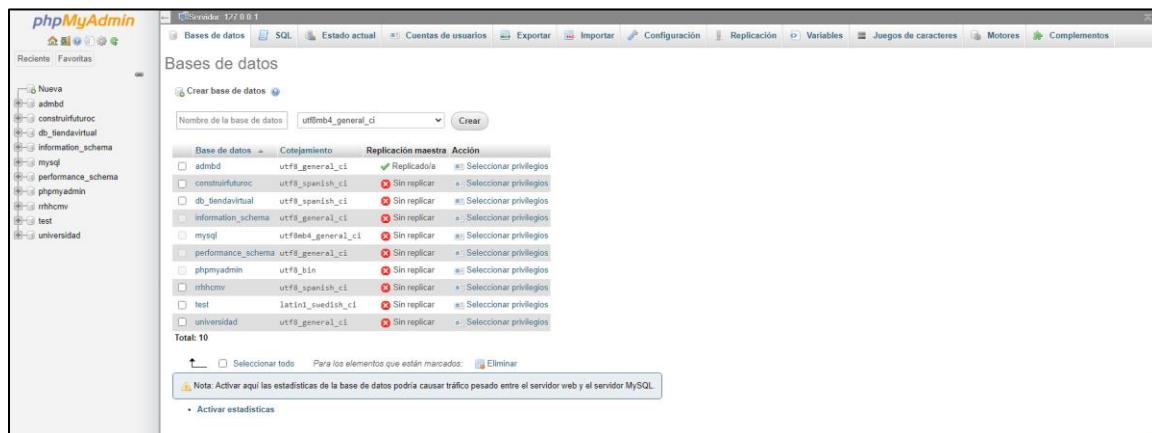


Ilustración 41. Verificación Bases de datos (Slave)

Base de datos	Cotejamiento	Replicación de la réplica	Acción
admbd	utf8_general_ci	Replicado/a	Seleccionar privilegios
information_schema	utf8_general_ci	Replicado/a	Seleccionar privilegios
mysql	utf8mb4_general_ci	Replicado/a	Seleccionar privilegios
performance_schema	utf8_general_ci	Replicado/a	Seleccionar privilegios
phpmyadmin	utf8_bin	Replicado/a	Seleccionar privilegios
test	latin1_swedish_ci	Replicado/a	Seleccionar privilegios

Total: 6

Activar Windows
Nota: Activar aquí las estadísticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor MySQL. Vea la configuración para activar Windows.

Al seguir estos pasos, se habrá configurado la replicación de la base de datos entre el servidor principal y el servidor secundario. Esto asegurará la disponibilidad y la redundancia de los datos, proporcionando un entorno más robusto y confiable.

6. Prueba y Validación del Sistema

En esta sección, se llevarán a cabo las pruebas y validaciones necesarias para asegurar el correcto funcionamiento de la base de datos y la replicación implementadas. Estas pruebas son fundamentales para garantizar la integridad de los datos, la respuesta del sistema y la confiabilidad de la replicación. A continuación, se presentan los pasos a seguir:

6.1. Pruebas de Funcionalidad

1. Se selecciona las siguientes tablas para realizar pruebas:

Entidades padres	Entidades hijas
“pnf”, “trayecto”	“maya_curricular”

2. Se realizan pruebas de inserción, actualización y eliminación de registros de las entidades padres en la base de datos haciendo uso de visual studio code.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 42. Insert de la tabla "pnf"

The screenshot shows a Visual Studio Code interface with a MySQL extension. The code editor contains an SQL script named 'INSERT INTO PNF.sql' which inserts 11 rows into the 'admbd.pnf' table. The rows describe various professional roles ('PNF') such as 'Administración', 'Agrícola', 'Contaduría Pública', etc. The terminal below shows the command being run and its successful completion.

```
INSERT INTO PNF.sql - AdmBD - Visual Studio Code
INSERT INTO PNF.sql
1  INSERT INTO admbd.pnf (id_pnf, pnf, desc_pnf) VALUES
2  (NULL, 'PNF Administración', 'Es un profesional que debe desempeñarse en tareas propias de la administración en el área contable y de costos industriales.'),
3  (NULL, 'PNF Agroalimentación', 'Es un profesional que debe desempeñarse en tareas propias de la Agroalimentación.'),
4  (NULL, 'PNF Contaduría Pública', 'Es un profesional que debe desempeñarse en tareas propias de la Contaduría Pública.'),
5  (NULL, 'PNF Electricidad', 'Es un profesional que debe desempeñarse en tareas propias de la Electricidad.'),
6  (NULL, 'PNF Electrónica', 'Es un profesional que debe desempeñarse en tareas propias de la Electrónica.'),
7  (NULL, 'PNF Informática', 'Es un profesional que debe desempeñarse en tareas propias de la Informática.'),
8  (NULL, 'PNF Instrumentación', 'Es un profesional que debe desempeñarse en tareas propias de la Instrumentación y Control.'),
9  (NULL, 'PNF Mantenimiento', 'Es un profesional que debe desempeñarse en tareas propias del Mantenimiento.'),
10 (NULL, 'PNF Mecánica', 'Es un profesional que debe desempeñarse en tareas propias de la Mecánica.'),
11 (NULL, 'PNF Sistemas de Calidad y Ambiente', 'Es un profesional que debe desempeñarse en tareas propias de Sistemas de Calidad y Ambiente.')

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL MySQL
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":10,"insertId":1,"serverStatus":2,"warningCount":0,"message":"Records: 10  Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.

SÍMBOLOS DE REVISIÓN: master 0 11 ✓ 0 0 △ 0 Blackbox localhost admbd LÍNEA: 11, col. 142 ESPACIOS: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell
```

Ilustración 43. Insert de la tabla "trayecto"

The screenshot shows a Visual Studio Code interface with a MySQL extension. The code editor contains an SQL script named 'INSERT INTO TRAYECTO.sql' which inserts 6 rows into the 'admbd.trayecto' table. The rows represent different types of journeys ('I', 'II', 'III', 'IV', 'V'). The terminal below shows the command being run and its successful completion.

```
INSERT INTO TRAYECTO.sql - AdmBD - Visual Studio Code
INSERT INTO TRAYECTO.sql
1  INSERT INTO admbd.trayecto (id_trayecto, trayecto) VALUES
2  (NULL, 'I'),
3  (NULL, 'II'),
4  (NULL, 'III'),
5  (NULL, 'IV'),
6  (NULL, 'V');

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL MySQL
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":5,"insertId":1,"serverStatus":2,"warningCount":0,"message":"5 Records: 5  Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.

SÍMBOLOS DE REVISIÓN: master 0 11 ✓ 0 0 △ 0 Blackbox localhost admbd LÍNEA: 6, col. 13 ESPACIOS: 4 UTF-8 CRLF MySQL Go Live Blackbox Spell
```

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 44. Insert de la tabla "maya_curricular"

The screenshot shows the Visual Studio Code interface with the MySQL extension. The Explorer sidebar on the right shows the database schema, including the 'maya_curricular' table with columns: id_maya (int(11)), maya_curricular (varchar(100)), fech_creacion (datetime), and key_pnf (int(11)). The terminal tab displays the execution of an INSERT query into the 'maya_curricular' table, which inserts 6 rows of data. The status bar at the bottom indicates the query was executed successfully.

```
INSERT INTO MAYA_CURRICULAR.sql - AdmBD - Visual Studio Code
INSERT INTO PNF.sql X INSERT INTO TRAYECTO.sql X INSERT INTO MAYA_CURRICULARsql X
INSERT INTO MAYA_CURRICULAR.sql
1  INSERT INTO maya_curricular (id_maya, maya_curricular, fech_creacion, key_pnf) VALUES
2  (NULL, '0', current_timestamp(), '1'),
3  (NULL, '1', current_timestamp(), '1'),
4  (NULL, '0', current_timestamp(), '2'),
5  (NULL, '1', current_timestamp(), '2'),
6  (NULL, '0', current_timestamp(), '4'),
7  (NULL, '1', current_timestamp(), '4');

[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":5,"insertId":1,"serverStatus":2,"warningCount":0,"message":"&Records:
5 Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":6,"insertId":1,"serverStatus":2,"warningCount":0,"message":"&Records:
6 Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.

Lín. 7, col. 39  Espacios: 4  UTF-8  CRLF  MySQL  Go Live  Blackbox  Spell  Q
```

Ilustración 45. Update de la tabla "maya_curricular"

The screenshot shows the Visual Studio Code interface with the MySQL extension. The Explorer sidebar on the right shows the database schema, including the 'maya_curricular' table with columns: id_maya (int(11)), maya_curricular (varchar(100)), fech_creacion (datetime), and key_pnf (int(11)). The terminal tab displays the execution of an UPDATE query on the 'maya_curricular' table, setting key_pnf to 3 where id_maya is 5. The status bar at the bottom indicates the query was executed successfully.

```
UPDATE MAYA_CURRICULAR.sql - AdmBD - Visual Studio Code
UPDATE INTO TRAYECTO.sql X UPDATE INTO MAYA_CURRICULAR.sql X UPDATE MAYA_CURRICULARsql U X
UPDATE MAYA_CURRICULAR.sql
1  UPDATE maya_curricular SET key_pnf = 3 WHERE maya_curricular.id_maya = 5;

[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":5,"insertId":1,"serverStatus":2,"warningCount":0,"message":"&Records:
5 Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":6,"insertId":1,"serverStatus":2,"warningCount":0,"message":"&Records:
6 Duplicates: 0  Warnings: 0","protocol41":true,"changedRows":0}
[Done] Finished MySQL query.
[Start] Executing MySQL query...
{"fieldCount":0,"affectedRows":1,"insertId":1,"serverStatus":2,"warningCount":0,"message":"(Rows
matched: 1  Changed: 1  Warnings: 0","protocol41":true,"changedRows":1}
[Done] Finished MySQL query.

Lín. 1, Col. 1 (73 seleccionada)  Espacios: 4  UTF-8  CRLF  MySQL  Go Live  Blackbox  Spell  Q
```

3. Se verifican que los cambios se reflejen correctamente en la replicación y que no se produzcan pérdidas de datos o conflictos.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 46. Verificación del insert pnf (Master)

The screenshot shows the phpMyAdmin interface for the 'admbd' database. The 'pnf' table is selected. The table structure is as follows:

	id_pnf	pnf	desc_pnf
1	1	PNF Administración	Es un profesional que debe desempeñarse en tareas ...
2	2	PNF Agroalimentación	Es un profesional que debe desempeñarse en tareas ...
3	3	PNF Contaduría Pública	Es un profesional que debe desempeñarse en tareas ...
4	4	PNF Electricidad	Es un profesional que debe desempeñarse en tareas ...
5	5	PNF Electrónica	Es un profesional que debe desempeñarse en tareas ...
6	6	PNF Informática	Es un profesional que debe desempeñarse en tareas ...
7	7	PNF Instrumentación y Control	Es un profesional que debe desempeñarse en tareas ...
8	8	PNF Mantenimiento	Es un profesional que debe desempeñarse en tareas ...
9	9	PNF Mecánica	Es un profesional que debe desempeñarse en tareas ...
10	10	PNF Sistemas de Calidad y Ambiente	Es un profesional que debe desempeñarse en tareas ...

Ilustración 47. Verificación del insert pnf (Slave)

The screenshot shows the phpMyAdmin interface for the 'admbd' database on a slave server. The 'pnf' table is selected. The table structure is as follows:

	id_pnf	pnf	desc_pnf
1	1	PNF Administración	Es un profesional que debe desempeñarse en tareas ...
2	2	PNF Agroalimentación	Es un profesional que debe desempeñarse en tareas ...
3	3	PNF Contaduría Pública	Es un profesional que debe desempeñarse en tareas ...
4	4	PNF Electricidad	Es un profesional que debe desempeñarse en tareas ...
5	5	PNF Electrónica	Es un profesional que debe desempeñarse en tareas ...
6	6	PNF Informática	Es un profesional que debe desempeñarse en tareas ...
7	7	PNF Instrumentación y Control	Es un profesional que debe desempeñarse en tareas ...
8	8	PNF Mantenimiento	Es un profesional que debe desempeñarse en tareas ...

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 48. Verificación del insert trayecto (Master)

The screenshot shows the phpMyAdmin interface for the 'admbd' database. The 'trayecto' table is selected. The table structure is shown with columns: id_trayecto and trayecto. Data rows are listed with IDs 1 through 5, corresponding to values I, II, III, IV, and V respectively. The interface includes standard phpMyAdmin navigation and search tools.

Ilustración 49. Verificación del insert trayecto (Slave)

The screenshot shows the phpMyAdmin interface for the 'admbd' database on a slave server (Windows 10). The 'trayecto' table is selected. The table structure and data are identical to the master server, showing five rows with IDs 1 through 5 and values I, II, III, IV, and V. The interface includes standard phpMyAdmin navigation and search tools.

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 50. Verificación del insert maya_curricular (Master)

The screenshot shows the phpMyAdmin interface for the 'admbd' database. The 'maya_curricular' table is selected. The table structure includes columns: id_maya, maya_curricular, fech_creacion, and key_pnf. The data shows 6 rows inserted at the same timestamp (2023-06-25 09:16:31). The 'key_pnf' column has values 1, 1, 2, 2, 4, and 4 respectively.

	id_maya	maya_curricular	fech_creacion	key_pnf
1	1	0	2023-06-25 09:16:31	1
2	2	1	2023-06-25 09:16:31	1
3	3	0	2023-06-25 09:16:31	2
4	4	1	2023-06-25 09:16:31	2
5	5	0	2023-06-25 09:16:31	4
6	6	1	2023-06-25 09:16:31	4

Ilustración 51. Verificación del insert maya_curricular (Slave)

The screenshot shows the phpMyAdmin interface for the 'admbd' database on a slave server. The 'maya_curricular' table is selected. The table structure and data are identical to the master server, with 6 rows inserted at 2023-06-25 09:16:31. The 'key_pnf' column values are 1, 1, 2, 2, 4, and 4.

	id_maya	maya_curricular	fech_creacion	key_pnf
1	1	0	2023-06-25 09:16:31	1
2	2	1	2023-06-25 09:16:31	1
3	3	0	2023-06-25 09:16:31	2
4	4	1	2023-06-25 09:16:31	2
5	5	0	2023-06-25 09:16:31	4
6	6	1	2023-06-25 09:16:31	4

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 52. Verificación del update maya_curricular (Master)

The screenshot shows the phpMyAdmin interface for the 'admbd' database. The 'maya_curricular' table is selected. The data grid shows the following rows:

	id_maya	maya_curricular	fech_creacion	key_pnf
<input type="checkbox"/>	1	0	2023-06-25 09:16:31	1
<input type="checkbox"/>	2	1	2023-06-25 09:16:31	1
<input type="checkbox"/>	3	0	2023-06-25 09:16:31	2
<input type="checkbox"/>	4	1	2023-06-25 09:16:31	2
<input type="checkbox"/>	5	0	2023-06-25 09:16:31	3
<input type="checkbox"/>	6	1	2023-06-25 09:16:31	4

Ilustración 53. Verificación del update maya_curricular (Slave)

The screenshot shows the phpMyAdmin interface for the 'admbd' database on the slave server. The 'maya_curricular' table is selected. The data grid shows the following rows:

	id_maya	maya_curricular	fech_creacion	key_pnf
<input type="checkbox"/>	1	0	2023-06-25 09:16:31	1
<input type="checkbox"/>	2	1	2023-06-25 09:16:31	1
<input type="checkbox"/>	3	0	2023-06-25 09:16:31	2
<input type="checkbox"/>	4	1	2023-06-25 09:16:31	2
<input type="checkbox"/>	5	0	2023-06-25 09:16:31	3
<input type="checkbox"/>	6	1	2023-06-25 09:16:31	4

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

6.2. Validación de Replicación

Ilustración 54. Verificación del servidor (Master)

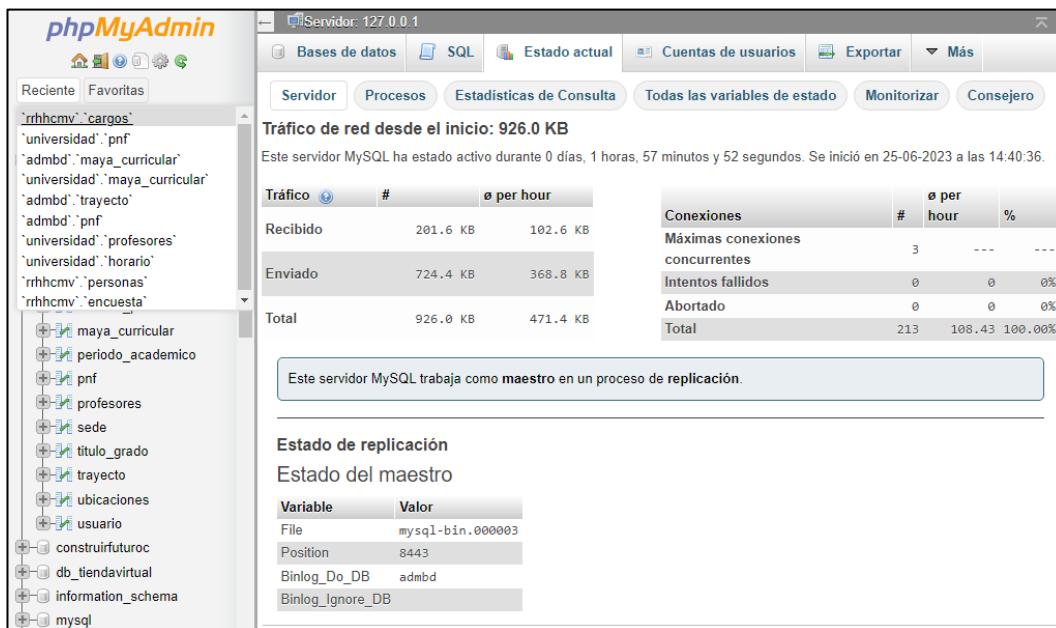


Ilustración 55. Verificación del servidor (Slave)

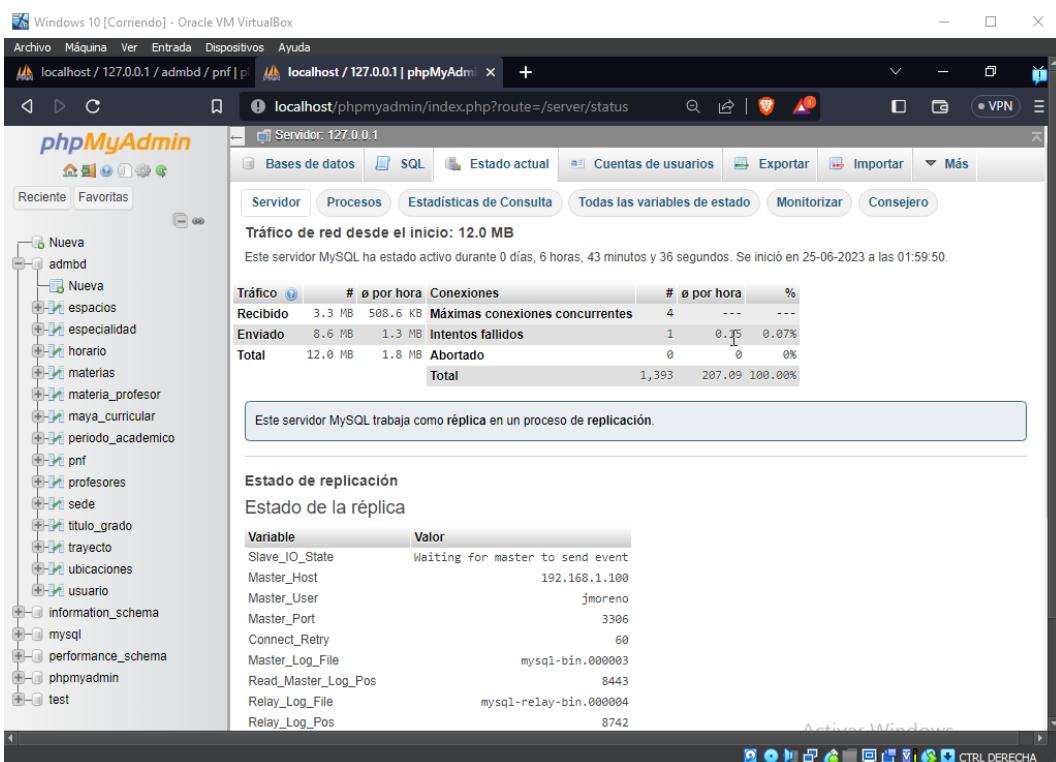
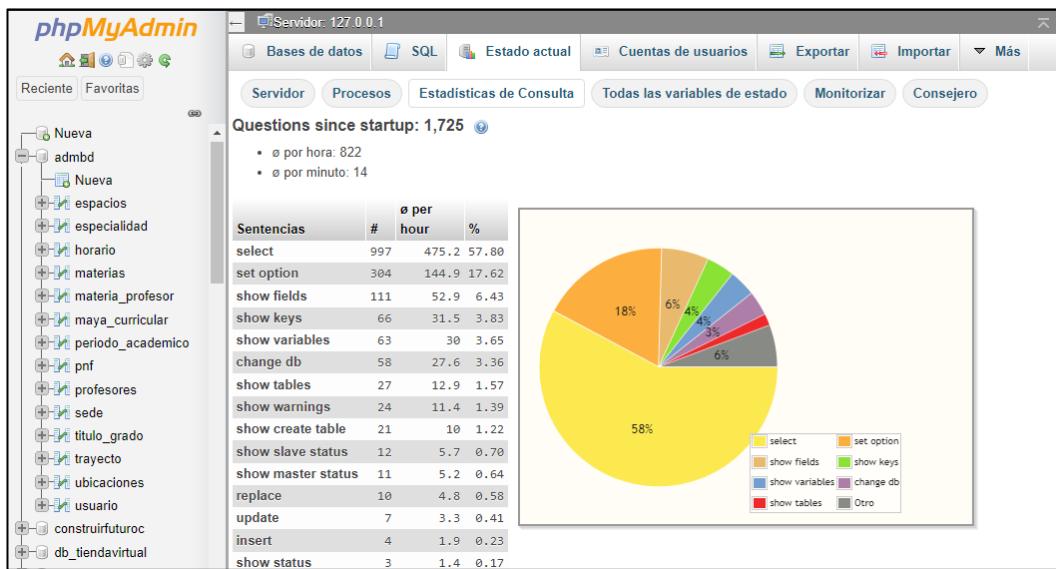
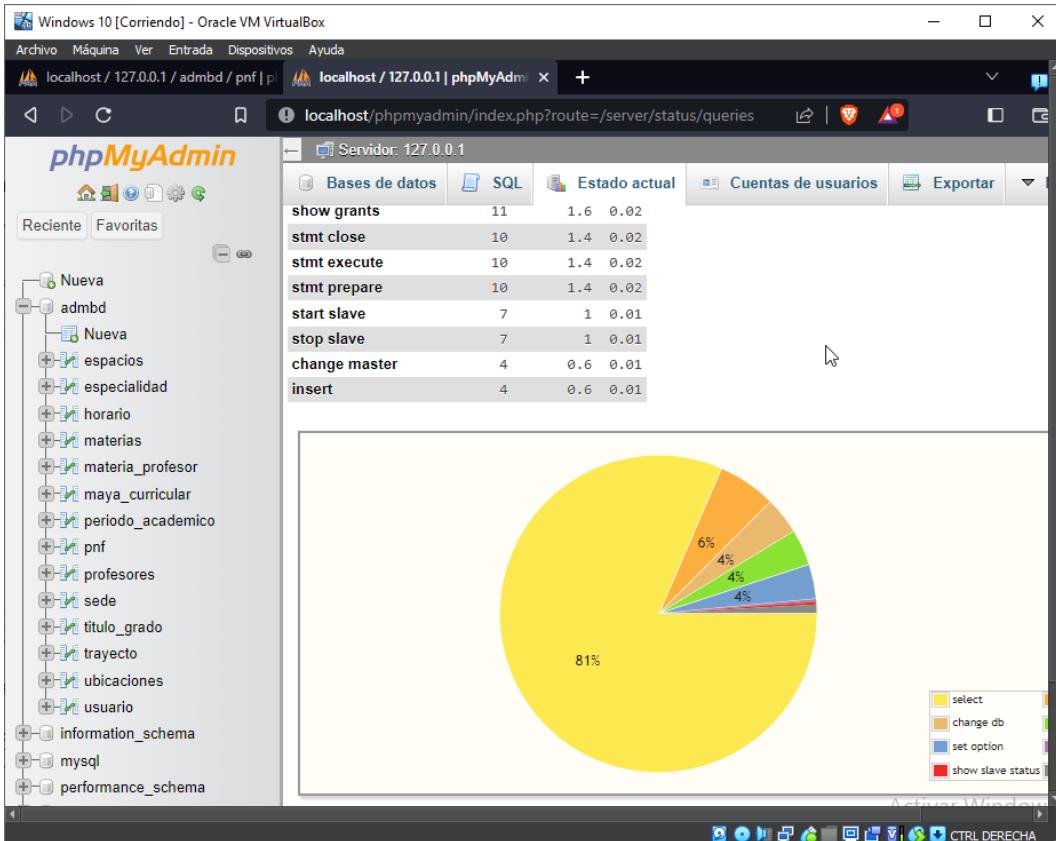


Ilustración 56. Verificación de estadísticas (Master)**Ilustración 57.** Verificación de estadísticas (Slave)

Documento de memoria técnica

Replicación de una base de datos ACT1_ABD_18608180

Versión: <1.0.0>

Ilustración 58. Verificación de variables de estados (Master)

Variable	Valor	Descripción
Rpl status	AUTH_MASTER	El estado de la replicación a prueba de fallos (aún no ha sido implementada).
Slave connections	1	
Slave heartbeat period	0	
Slave open temp tables	0	El número de tablas temporales actualmente abiertas por el proceso SQL esclavo.
Slave received heartbeats	0	
Slave retried transactions	0	Número total de veces (desde el arranque) que el proceso SQL esclavo de replicación ha reintentado hacer transacciones.
Slave running	OFF	Está ENCENDIDO si este servidor es un esclavo que está conectado a un master.
Slave skipped errors	0	

Ilustración 59. Verificación de variables de estados (Slave)

Variable	Valor	Descripción
Rpl status	AUTH_MASTER	El estado de la replicación a prueba de fallos (aún no ha sido implementada).
Slave connections	0	
Slave heartbeat period	30	
Slave open temp tables	0	El número de tablas temporales abiertas actualmente por el hilo SQL de réplica.
Slave received heartbeats	270	
Slave retried transactions	0	Número total (desde el inicio) de veces que el hilo SQL de réplica de replicación ha reintentado transacciones.
Slave running	ON	Está ACTIVADO si este servidor es una réplica que está conectada a un servidor principal.
Slave skipped errors	0	

Al finalizar esta etapa de prueba y validación, tendrás la confianza de que la base de datos y la replicación están funcionando correctamente y son capaces de gestionar las demandas del sistema de manera eficiente y confiable.

7. Conclusiones y Recomendaciones

En esta sección, se presentarán las conclusiones y recomendaciones derivadas del proceso de implementación del script SQL de la base de datos y la configuración de la replicación utilizando máquinas virtuales. A continuación, se presentan los puntos clave a considerar:

7.1. Conclusiones

El desarrollo del script SQL de la base de datos permitió establecer una estructura sólida y coherente para almacenar y gestionar los datos. La correcta definición de tablas, relaciones y restricciones garantiza la integridad y la eficiencia en el acceso a la información.

La configuración de la replicación entre el servidor principal y el servidor secundario mediante máquinas virtuales y/o locales proporciona una solución confiable y redundante para asegurar la disponibilidad de los datos. La sincronización continua de los cambios garantiza la integridad de los datos y la capacidad de respuesta del sistema.

Realizar las actividades en el ambiente de trabajo permite simular las condiciones reales en las que se utilizará la base de datos y la replicación. Esto facilita la validación de su funcionamiento y la identificación de posibles problemas o ajustes necesarios.

7.2. Recomendaciones

Mantenimiento regular: Es fundamental llevar a cabo un mantenimiento regular de la base de datos y el entorno de replicación. Esto incluye copias de seguridad periódicas, monitoreo del rendimiento, actualizaciones de software y hardware, y revisión de permisos y seguridad.

Pruebas exhaustivas: Realiza pruebas exhaustivas antes de implementar la base de datos y la replicación en un entorno de producción. Verifica el funcionamiento correcto de las consultas, la replicación de datos y la commutación por error.

Documentación actualizada: Mantén la documentación de la base de datos y la replicación actualizada a medida que se realicen modificaciones o mejoras. Esto facilitará el entendimiento y el mantenimiento del sistema en el futuro.

Capacitación y formación: Proporciona capacitación adecuada a los administradores y usuarios involucrados en la gestión de la base de datos y la replicación. Esto garantizará un uso correcto y eficiente del sistema, así como una respuesta adecuada en caso de incidentes.

Supervisión continua: Implementa una supervisión continua del sistema para detectar posibles problemas o necesidades de ajuste. Utiliza herramientas de monitoreo y establece alertas para recibir notificaciones de eventos relevantes.