

MANUAL TECNICO

**SOFTWARE PARA EL MANEJO DE USUARIOS Y LA REALIZACION DE RESERVAS
PERSONALIZADAS PARA EL CONJUNTO RESIDENCIAL RECODO DE CEDRO SUBA**

JOSEPH NICOLAS VARON VARGAS

JUAN SEBASTIAN CASTRO MARTINEZ

FICHA: 2771440V2

SERVICIO NACIONAL DE APRENDIZAJE SENA

CENTRO DE ELECTRICIDAD, ELECTRONICA Y TELECOMUNICACIONES CEET

TECNICO EN PROGRAMACION DE SOFTWARE

BOGOTA D.C., SEPTIEMBRE 2012

INTRODUCCION

Este manual describe una solución tecnológica para mejorar el proceso de agendamiento en el Conjunto Residencial Recodo de Cedro Suba. Actualmente, se utilizan métodos anticuados como hojas sueltas y libros de registro, lo que causa problemas de organización y eficiencia.

El nuevo sistema propuesto es una aplicación digital que centralizará toda la información de agendamientos. Sus principales características incluyen:

- Una base de datos unificada
- Una interfaz fácil de usar
- Acceso para múltiples usuarios desde diferentes dispositivos

Con esta herramienta, los administradores podrán crear y modificar citas, registrar asistencias y generar informes automáticamente. Esto ayudará a reducir errores, ahorrar tiempo y facilitar el análisis de datos.

La implementación se realizará gradualmente, empezando por instalar el software y trasladar la información existente. Luego, se capacitará al personal y se hará una prueba antes de usarlo definitivamente.

CONTENIDO

INTRODUCCION	2
1. REQUERIMIENTOS TÉCNICOS.....	5
2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	6
2.1.1 PHP.....	6
2.2.2 DISEÑO WEB.....	6
2.2.3 HTML.....	6
2.2.4 CSS.....	6
2.2.5 JAVASCRIPT.....	6
2.2.6 MYSQL	7
2.2.7 XAMPP.....	7
2.2.8 AZURE JUNTO A MYSQL WORKBENCH.....	8
3. CASOS DE USO.....	9
4. DIAGRAMA DE DESPLIEGUE	21
5. DIAGRAMA DE CLASES	22
6. MODELO RELACIONAL	23
7. SENTENCIAS DDL	24
8. SENTENCIAS DML	25
9. DICCIONARIO DE DATOS	26
10. EJECUCION DE SENTENCIAS EN WORKBENCH	28
10.1 EJECUCION DE SENTENCIAS EN WORKBENCH	28
10.2 EJECUCION DE SENTENCIAS EN WORKBENCH	29

TABLAS CU

Tabla de contenido

Escribir el título del capítulo (nivel 1)	1
Escribir el título del capítulo (nivel 2)	2
Escribir el título del capítulo (nivel 3)	3
Escribir el título del capítulo (nivel 1)	4
Escribir el título del capítulo (nivel 2)	5
Escribir el título del capítulo (nivel 3)	6
1. TABLA 1 INICIAR SESION	10
2. TABLA 2 CERRAR SESION	10
3. TABLA 3 RECUPERAR CONTRASEÑA	10
4. TABLA 4 REGISTRAR ROL	11
5. TABLA 5 CONSULTAR ROL	11
6. TABLA 6 CONSULTAR ROL POR ID	11
7. TABLA 7 ELIMINAR ROL	12
8. TABLA 8 ACTUALIZAR ROL	12
9. TABLA 9 REGISTRAR CASA	13
10. TABLA 10 CONSULTAR CASA	13
11. TABLA 11 CONSULTAR CASA POR ID	13
12. TABLA 12 ELIMINAR CASA	14
13. TABLA 13 ACTUALIZAR CASA	14
14. TABLA 14 REGISTRAR USUARIO	15
15. TABLA 15 CONSULTAR USUARIO	15
16. TABLA 16 CONSULTAR USUARIO POR ID	15
17. TABLA 17 ELIMINAR USUARIO	16
18. TABLA 18 ACTUALIZAR USUARIO	16
19. TABLA 19 REGISTRAR LUGAR	17
20. TABLA 20 CONSULTAR LUGAR	17
21. TABLA 21 ACTUALIZAR LUGAR	17
22. TABLA 22 ELIMINAR LUGAR	18
23. TABLA 23 REGISTRAR RESERVA	18
24. TABLA 24 CONSULTAR RESERVA	19
25. TABLA 25 APROBAR RESERVA	19

26. TABLA 26 RECHAZAR RESERVA	20
27. TABLA 27 ELIMINAR RESERVA	20

1. REQUERIMIENTOS TÉCNICOS.

- REQUERIMIENTOS TÉCNICOS

REQUERIMIENTOS MÍNIMOS DE HARDWARE

- ✓ Procesador: Intel Pentium 4 o equivalente en Amd
- ✓ Memoria RAM: Mínimo: 1 Gigabyte (GB)
- ✓ Disco Duro: 500 GB de espacio libre

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- ✓ Sistema Operativo: Windows 7 o posterior
- ✓ Navegador web: Google Chrome (versión 50 o posterior), Mozilla Firefox (versión 45 o posterior), o Microsoft Edge

Notas adicionales:

- Se recomienda una conexión a Internet de al menos 1 Mbps para un rendimiento óptimo.
- Aunque Gmail puede funcionar con especificaciones inferiores, estos requisitos aseguran una experiencia de usuario fluida.
- Para sistemas operativos más antiguos como los mencionados (Windows NT/98/Me/2000/2003/XP/Vista), es posible que algunas funciones de Gmail no estén disponibles o el rendimiento sea inferior.

2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

2.1 PHP

PHP es un lenguaje de programación diseñado para crear páginas web dinámicas. Se integra fácilmente con HTML, permitiendo a los desarrolladores combinar código PHP dentro del marcado HTML.

Las ejecuciones de PHP se realizan en el servidor, y el cliente (navegador web) recibe solo los resultados de estas ejecuciones. Cuando un cliente hace una petición, el servidor procesa el código PHP y genera contenido dinámico que se envía de vuelta al cliente.

Una de las principales ventajas de PHP es su capacidad para conectarse con diversos sistemas de gestión de bases de datos, incluyendo MySQL, Oracle, PostgreSQL y SQL Server. Esto permite crear aplicaciones web robustas y escalables.

PHP puede ejecutarse en la mayoría de los sistemas operativos modernos y es compatible con los servidores web más populares. Su versatilidad lo hace ideal para desarrollar desde sitios web simples hasta complejas aplicaciones empresariales, sistemas de gestión de contenidos (CMS) y plataformas de comercio electrónico.

Además, PHP cuenta con un amplio ecosistema de frameworks y bibliotecas que facilitan el desarrollo rápido y eficiente de aplicaciones web modernas.

2.2 DISEÑO WEB

2.2.1 HTML (HyperText Markup Language):

- Propósito: Es el lenguaje de marcado principal para estructurar el contenido en la web.

- **Características:** Utiliza etiquetas para definir elementos como encabezados, párrafos, enlaces, listas y otros componentes de la página. HTML establece la estructura básica de una página web.

2.2.2 CSS (Cascading Style Sheets):

- **Propósito:** Se usa para estilizar y diseñar la apariencia de los elementos HTML.
- **Características:** Permite aplicar estilos como colores, fuentes, márgenes y layout a los elementos HTML. CSS ayuda a hacer que las páginas web sean visualmente atractivas y coherentes con el diseño deseado.

2.2.3 JavaScript:

- **Propósito:** Es un lenguaje de programación que se usa para agregar interactividad y funcionalidad dinámica a las páginas web.
- **Características:** Permite manipular el contenido HTML y CSS, responder a eventos del usuario (como clics y desplazamientos), y realizar operaciones más avanzadas como validación de formularios y manipulación de datos.

2.2.4 MYSQL WORKBENCH

MySQL Workbench es una herramienta gráfica desarrollada por Oracle que se utiliza para el diseño, desarrollo y administración de bases de datos MySQL. Proporciona una interfaz visual para gestionar y modelar bases de datos, crear y ejecutar consultas SQL, y administrar usuarios y permisos. También permite la administración de instancias de MySQL, la realización de copias de seguridad y la restauración de datos, así como la generación de informes. Su objetivo es simplificar y agilizar las tareas relacionadas con la gestión de bases de datos MySQL.

2.2.5 XAMPP

XAMPP es una solución integral de software libre diseñada para simplificar el desarrollo web local. Incluye los siguientes componentes clave:

Apache: Un servidor web de código abierto que permite gestionar y servir contenido web. Es ampliamente utilizado debido a su flexibilidad y robustez.

MySQL/MariaDB: Sistemas de gestión de bases de datos relacionales. MySQL es conocido por su rendimiento y fiabilidad, mientras que MariaDB es una bifurcación de MySQL que ofrece mejoras y características adicionales.

XAMPP agrupa estos componentes en un solo paquete, facilitando la instalación y configuración de un entorno de desarrollo web completo en la computadora del usuario. Esto permite a los desarrolladores crear, probar y depurar aplicaciones web en un entorno local antes de su implementación en servidores de producción. La interfaz gráfica de XAMPP también simplifica la administración de los servicios, como iniciar y detener Apache o MySQL.

2.2.6 AZURE JUNTO A MYSQL WORKBENCH

Azure es la plataforma de servicios en la nube de Microsoft, que ofrece una amplia gama de servicios en la nube, incluyendo almacenamiento, cómputo, análisis y bases de datos. Uno de los servicios disponibles en Azure es Azure Database for MySQL, que es una base de datos relacional basada en MySQL completamente administrada y escalable.

MySQL Workbench es una herramienta gráfica para el diseño, desarrollo y administración de bases de datos MySQL. Puedes usar MySQL Workbench para conectarte y gestionar bases de datos MySQL que se ejecutan en Azure. Aquí te explico cómo se conectan ambos:

3. CASOS DE USO

En el desarrollo se realizaron 4 roles, uno de estos mas complementario que funcional puesto que cumple las mismas funciones que otro. Este rol es el de arrendatario ya que va a poder realizar las mismas funciones que un habitante del conjunto.

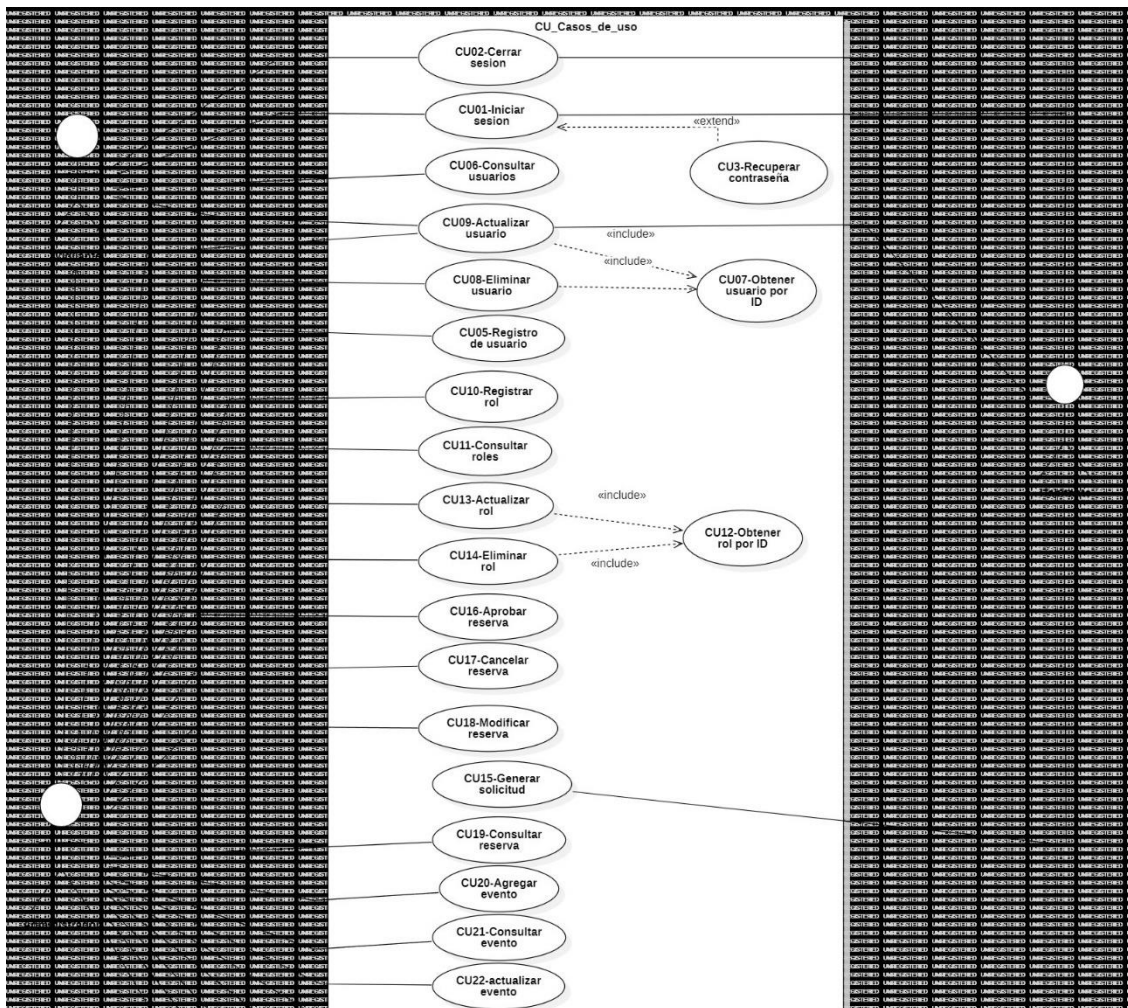


TABLA 1 – Inicio de sesión

NOMBRE: CU1 Inicio de sesión
Rol(es): Administrador, vigilante, habitante y arrendatario
Descripción: el usuario debe dirigirse al login para poder ingresar sus credenciales y poder así efectuar sus tareas posteriores
Flujo normal: <ol style="list-style-type: none"> 1. El usuario abre el login 2. Ingresa su usuario 3. Ingresa su contraseña 4. Inicia sesión
Pos condición: <ol style="list-style-type: none"> 1. Sesión iniciada

TABLA 2 – Cerrar sesión

NOMBRE: CU2 cerrar sesión
Rol(es): Administrador, vigilante, habitante y arrendatario
Descripción: el usuario debe dirigirse al logout para poder salir de su sesión
Flujo normal: <ol style="list-style-type: none"> 1. El usuario se dirige al botón de cerrar sesión 2. Presiona cerrar sesión
Pos condición: <ol style="list-style-type: none"> 1. Sesión cerrada

TABLA 3 – Recuperar/cambiar contraseña

NOMBRE: CU3 recuperar o cambiar contraseña
Rol(es): Administrador, vigilante, habitante y arrendatario
Descripción: el usuario debe dirigirse al login para poder ubicar el botón de recuperar contra seña.
Flujo normal: <ol style="list-style-type: none"> 1. El usuario abre el login 2. Ubica el botón de recuperar contraseña

3. Ingresar su código de usuario junto a su nueva contraseña 4. Inicia sesión de nuevo
Pos condición: 2. Contraseña recuperada

TABLA 4 – Registrar rol

NOMBRE: CU4 registro de roles
Rol(es): Administrador
Descripción: el administrador debe registrar los roles esto dirigiéndose a la barra lateral con el nombre de roles
Flujo normal: 1. El administrador entra al módulo de roles 2. Llena los datos del rol a registrar (nombre del rol) 3. Guarda el rol
Pos condición: 1. rol registrado

TABLA 5 – consultar roles

NOMBRE: CU5 Consultar roles
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todos los roles registrados
Flujo normal: 1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de actualizar o eliminar rol
Flujo alternativo: Debe haber roles registrados
Pos condición: 1. Se visualizan los roles

TABLA 6 – consultar roles por id

NOMBRE: CU6 Consultar roles por id
Rol(es): Administrador, vigilante

Descripción: el administrador y vigilante podrán revisar todos los roles registrados por id para hacer una búsqueda específica de los mismos
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de roles y se ubica el campo de consultar roles 2. Se visualizan los roles 3. Se encuentra el botón de actualizar o eliminar roles
Flujo alternativo: Debe haber roles registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan los roles

TABLA 7 – eliminar roles

NOMBRE: CU7 Eliminar roles
Rol(es): Administrador
Descripción: el administrador podrá eliminar los roles a registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de eliminar rol 4. Se oprime y se elimina el rol
Flujo alternativo: Debe haber roles registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan los roles 2. Se elimina el rol

TABLA 8 – actualizar roles

NOMBRE: CU8 Actualizar roles
Rol(es): Administrador
Descripción: el administrador podrá actualizar los roles a registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de actualizar rol 4. Se oprime y se envía a la vista de actualizar el rol
Flujo alternativo: Debe haber roles registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza la información del rol

2. Se actualiza el rol

TABLA 9– Registrar casa

NOMBRE: CU9 registro de casas
Rol(es): Administrador
Descripción: el administrador debe registrar las casas esto dirigiéndose a la barra lateral con el nombre de casas
Flujo normal: <ol style="list-style-type: none"> 1. El administrador entra al módulo de casas 2. Llena los datos de la casa y registra la registra (nombre de la casa) 3. Guarda la casa
Pos condición: <ol style="list-style-type: none"> 1. casa registrada

TABLA 10– consultar casas

NOMBRE: CU10 Consultar casas
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todas las casas registradas
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de casas y se ubica el campo de consultar casa 2. Se visualizan las casas 3. Se encuentra el botón de actualizar o eliminar casa
Flujo alternativo: Debe haber casas registradas
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan las casas

TABLA 11 – consultar casas por id

NOMBRE: CU10 Consultar casas por id
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todas las casas registradas por id y poder consultarlas mejor
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de casas y se ubica el campo de consultar casa Se visualizan las casas Se encuentra el botón de actualizar o eliminar casa
Flujo alternativo: Debe haber casas registradas
Pos condición: <ol style="list-style-type: none"> Se visualizan las casas

TABLA 12 – eliminar casas

NOMBRE: CU12 Eliminar casas
Rol(es): Administrador
Descripción: el administrador podrá eliminar las casas registradas
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de casas y se ubica el campo de consultar casa Se visualizan las casas Se encuentra el botón de eliminar casa Se oprime y se elimina la casa
Flujo alternativo: Debe haber casas registradas
Pos condición: <ol style="list-style-type: none"> Se visualizan las casas Se elimina las casas

TABLA 13 – actualizar casas

NOMBRE: CU13 Actualizar casas
Rol(es): Administrador
Descripción: el administrador podrá actualizar las casas registradas
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de casas y se ubica el campo de consultar casa Se visualizan las casas Se encuentra el botón de actualizar casa Se oprime y se envía a la vista de actualizar la casa
Flujo alternativo: Debe haber casas registradas

Pos condición:

3. Se visualiza la información de la casa
4. Se actualiza la casa

TABLA 14 – registrar usuarios

NOMBRE: CU14 registro de usuarios
Rol(es): Administrador
Descripción: el administrador debe registrar los usuarios esto dirigiéndose a la barra lateral con el nombre de usuarios
Flujo normal: <ol style="list-style-type: none">1. El administrador entra al módulo de usuarios2. Llena los datos del usuario a registrar (rol, casa, nombres, apellidos, id...)3. Se le asigna un correo y contraseña
Pos condición: <ol style="list-style-type: none">2. Usuario registrado

TABLA 15 – consultar usuarios

NOMBRE: CU15 Consultar usuarios
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todos los usuarios registrados
Flujo normal: <ol style="list-style-type: none">3. Se abre el modulo de usuarios y se ubica el campo de consultar usuarios4. Se visualizan los usuarios5. Se encuentra el botón de actualizar o eliminar usuario
Flujo alternativo: Debe haber usuarios registrados
Pos condición: <ol style="list-style-type: none">2. Se visualizan los usuarios

TABLA 16 – consultar usuarios por id

NOMBRE: CU16 Consultar usuarios por id
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todos los usuarios registrados por id para hacer una búsqueda específica de los mismos
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de usuarios y se ubica el campo de consultar usuarios Se visualizan los usuarios Se encuentra el botón de actualizar o eliminar usuario
Flujo alternativo: Debe haber usuarios registrados
Pos condición: <ol style="list-style-type: none"> Se visualizan los usuarios

TABLA 17 – eliminar usuarios

NOMBRE: CU17 Eliminar usuarios
Rol(es): Administrador
Descripción: el administrador podrá eliminar los usuarios a registrados
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de usuarios y se ubica el campo de consultar usuarios Se visualizan los usuarios Se encuentra el botón de eliminar usuario Se oprime y se elimina el usuario
Flujo alternativo: Debe haber usuarios registrados
Pos condición: <ol style="list-style-type: none"> Se visualizan los usuarios Se elimina el usuario

TABLA 18 – actualizar usuario

NOMBRE: CU18 Actualizar usuario
Rol(es): Administrador
Descripción: el administrador podrá actualizar los usuarios a registrados
Flujo normal: <ol style="list-style-type: none"> Se abre el módulo de usuarios y se ubica el campo de consultar usuarios Se visualizan los usuarios Se encuentra el botón de actualizar usuario Se oprime y se envía a la vista de actualizar el usuario
Flujo alternativo: Debe haber usuarios registrados
Pos condición: <ol style="list-style-type: none"> Se visualiza la información del usuario

6. Se actualiza el usuario

TABLA 19 – agregar lugar

NOMBRE: CU19 agregar evento/lugar
Rol(es): Administrador
Descripción: el administrador podrá realizar el registro de algún lugar especial para las reservas
Flujo normal: <ol style="list-style-type: none"> 1. se abre el módulo de lugares 2. se registra en el campo de agregar lugar
Flujo alternativo: N/A
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza el modulo de lugares 2. Se agrega el lugar

TABLA 20 – consultar lugares

NOMBRE: CU20 Consultar lugares
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todos los lugares registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de lugares y se ubica el campo de consultar lugar 2. Se visualizan los lugares 3. Se encuentra el botón de actualizar o eliminar lugar
Flujo alternativo: Debe haber lugares registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan los lugares

TABLA 21 – actualizar lugar

NOMBRE: CU21 Actualizar lugar
Rol(es): Administrador

Descripción: el administrador podrá actualizar los lugares registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de lugares y se ubica el campo de consultar lugar 2. Se visualizan los lugares 3. Se encuentra el botón de actualizar lugar 4. Se oprime y se envía a la vista de actualizar el lugar
Flujo alternativo: Debe haber lugares registradas
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza la información del lugar 2. Se actualiza el lugar

TABLA 22 – eliminar lugares

NOMBRE: CU22 Eliminar lugares
Rol(es): Administrador
Descripción: el administrador podrá eliminar los lugares a registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de lugares y se ubica el campo de consultar lugar 2. Se visualizan los lugares 3. Se encuentra el botón de eliminar lugar 4. Se oprime y se elimina el lugar
Flujo alternativo: Debe haber roles registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan los roles 2. Se elimina el rol

TABLA 23 – agregar reserva

NOMBRE: CU23 agregar reserva
Rol(es): Administrador, vigilante, habitante y arrendatario
Descripción: los usuarios podrán realizar sus reservas
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de las reservas 2. Se llenan los datos predispuestos como fecha, lugar y código de usuario
Flujo alternativo: debe haber lugares y usuarios registrados
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza el módulo de reservas 2. Se agrega la reserva

TABLA 24 – consultar reservas

NOMBRE: CU24 Consultar reserva
Rol(es): Administrador, vigilante
Descripción: el administrador y vigilante podrán revisar todas las reservas registradas
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de reservas y se ubica el campo de consultar reserva 2. Se visualizan las reservas 3. Se encuentra el botón de aprobar o rechazar la reserva
Flujo alternativo: Debe haber reservas registradas
Pos condición: <ol style="list-style-type: none"> 1. Se visualizan las reservas registradas

TABLA 25 – aprobar reserva

NOMBRE: CU25 Aprobar reserva
Rol(es): Administrador
Descripción: el administrador podrá aprobar o rechazar la reserva. Esto actualizando el campo de aprobado o rechazado
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de reservas y se ubica el campo de consultar reserva 2. Se visualizan las reservas 3. Se encuentra el botón de aprobar 4. Se oprime y se actualiza a aprobada la reserva
Flujo alternativo: Debe haber reservas registradas
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza la información de la reserva 2. Se aprueba la reserva

TABLA 26 – rechazar reserva

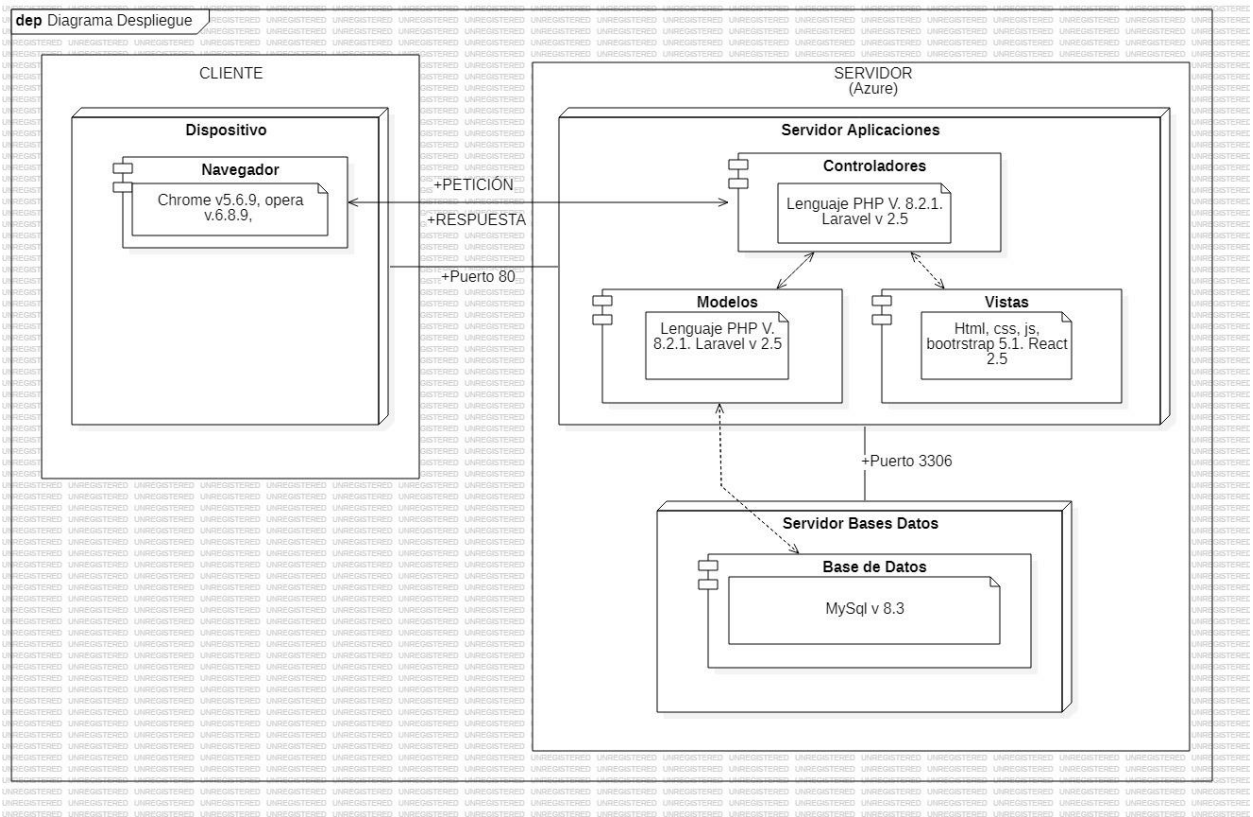
NOMBRE: CU26 Rechazar reserva
Rol(es): Administrador
Descripción: el administrador podrá aprobar o rechazar la reserva. Esto actualizando el campo de aprobado o rechazado
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de reservas y se ubica el campo de consultar reserva 2. Se visualizan las reservas 3. Se encuentra el botón de aprobar 4. Se oprime y se actualiza a rechazada la reserva
Flujo alternativo: Debe haber reservas registradas
Pos condición: <ol style="list-style-type: none"> 1. Se visualiza la información de la reserva 2. Se rechaza la reserva

TABLA 27 – eliminar reserva

NOMBRE: CU27 Eliminar reservas
Rol(es): Administrador
Descripción: el administrador podrá eliminar las reservas a registrados
Flujo normal: <ol style="list-style-type: none"> 1. Se abre el módulo de lugares y se ubica el campo de consultar reserva 2. Se visualizan las reservas 3. Se encuentra el botón de eliminar reserva 4. Se oprime y se elimina la reserva
Flujo alternativo: Debe haber reservas registradas
Pos condición: <ol style="list-style-type: none"> 3. Se visualizan las reservas 4. Se elimina la reserva

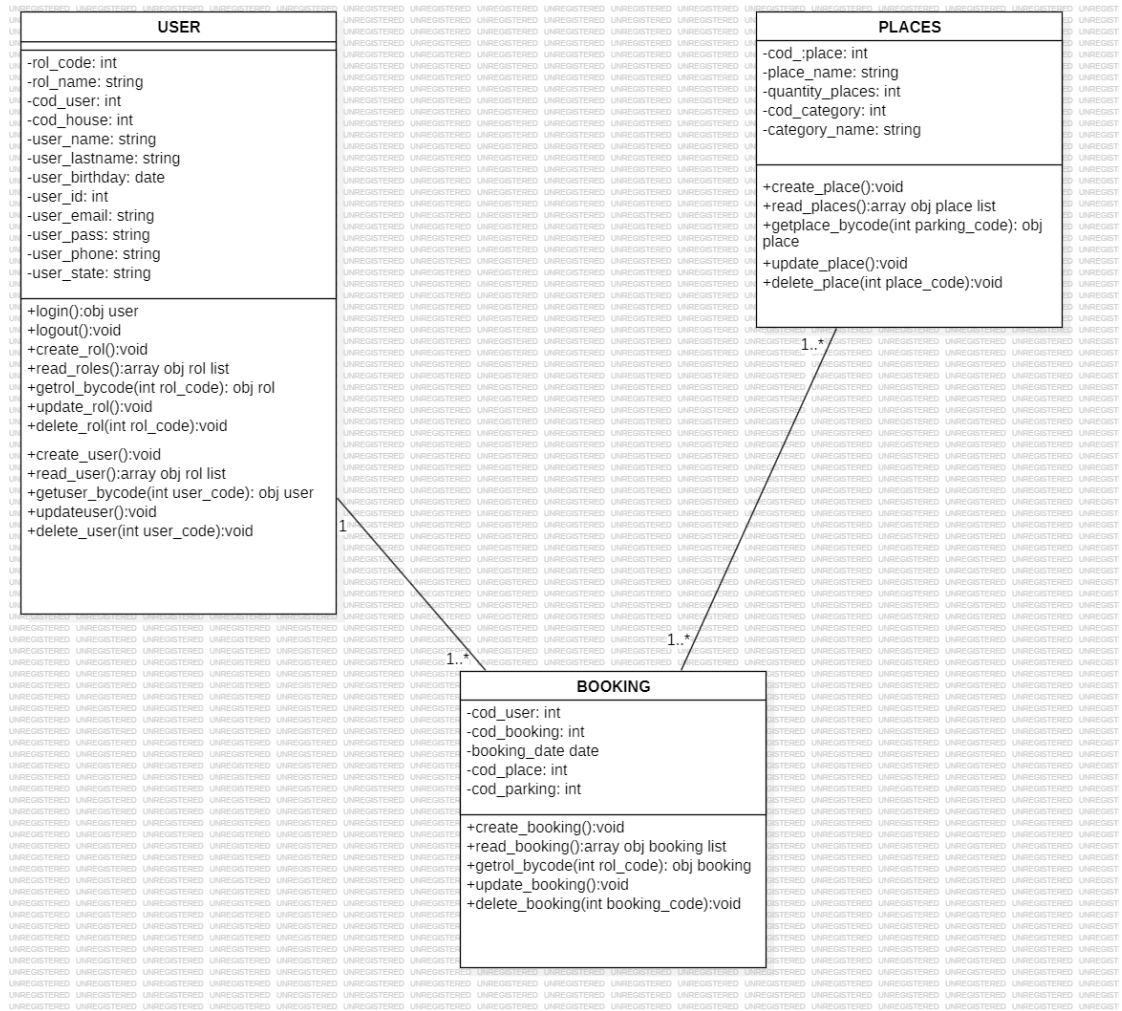
4. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue es una representación visual que muestra cómo los componentes de un sistema de software están distribuidos en el hardware y cómo interactúan entre sí. Permite entender la disposición física de los elementos del sistema, las conexiones entre ellos y la infraestructura necesaria para su funcionamiento. Es útil para planificar, implementar y gestionar el sistema en su entorno real, ayudando a asegurar un despliegue eficiente y efectivo.



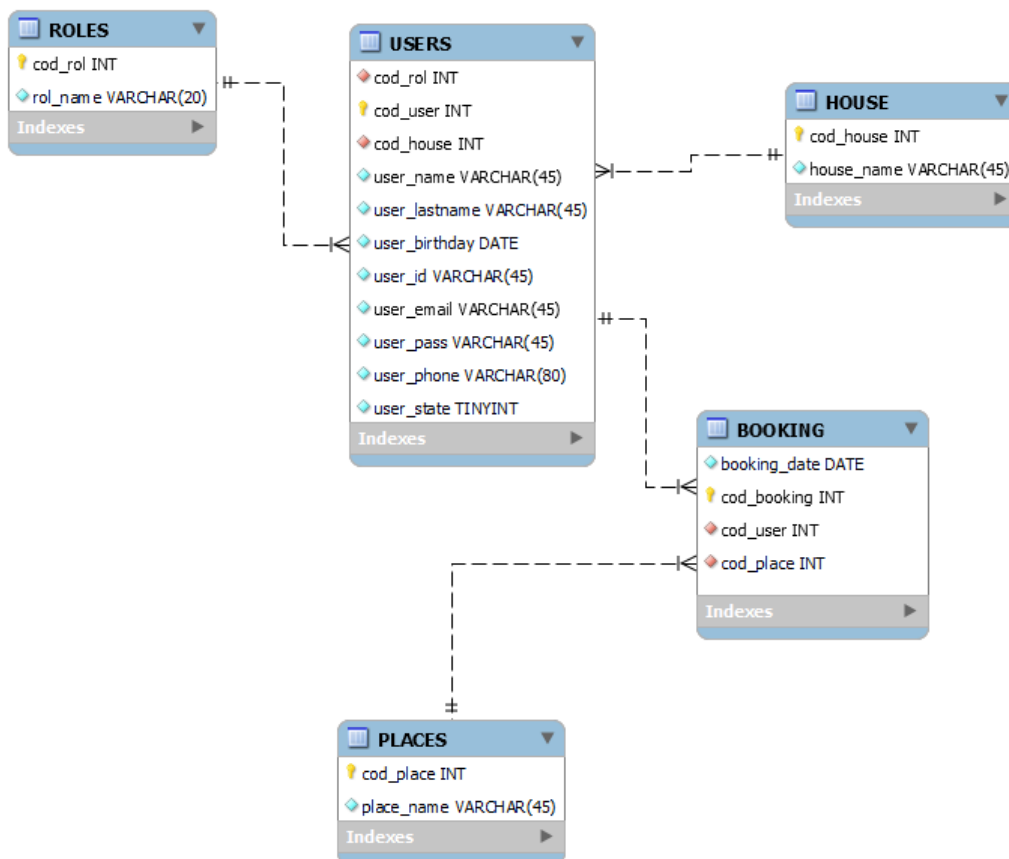
5. DIAGRAMA DE CLASES

El diagrama de clases es una representación visual que ilustra la estructura estática de un sistema de software, mostrando las clases, sus atributos, métodos y las relaciones entre ellas. Es esencial para entender y diseñar la arquitectura del sistema, facilitando la comunicación y la documentación del diseño de software.



6. MODELO RELACIONAL

el modelo relacional realizado en MySQL Workbench proporciona una visión estructurada y visual de cómo se organizan los datos en una base de datos, facilitando el diseño, la implementación y el mantenimiento del sistema de gestión de bases de datos.



7. SENTENCIAS DDL

Las sentencias DDL se utilizan para definir y modificar la estructura de las bases de datos y sus objetos (como tablas, índices y vistas). Estas sentencias afectan la estructura y el esquema de la base de datos, y suelen tener efectos permanentes que no se pueden deshacer fácilmente sin realizar una operación específica de reversión. En este caso se puede usar nuestra base de datos como referencia.

CREACION DE LAS TABLAS ROLES, USUARIOS, CASAS, LUGARES Y RESERVAS

```
16 -----
17 CREATE SCHEMA IF NOT EXISTS `conjuntos_reservas` DEFAULT CHARACTER SET utf8 ;
18 USE `conjuntos_reservas` ;
19
20 -----
21 -- Table `conjuntos_reservas`.`ROLES`
22 -----
23 CREATE TABLE IF NOT EXISTS `conjuntos_reservas`.`ROLES` (
24   `cod_rol` INT NOT NULL AUTO_INCREMENT,
25   `rol_name` VARCHAR(20) NOT NULL,
26   PRIMARY KEY (`cod_rol`))
27 ENGINE = InnoDB;
28
29
30 -----
31
32 -----
33 CREATE TABLE IF NOT EXISTS `conjuntos_reservas`.`HOUSE` (
34   `cod_house` INT NOT NULL AUTO_INCREMENT,
35   `house_name` VARCHAR(45) NOT NULL,
36   PRIMARY KEY (`cod_house`))
37 ENGINE = InnoDB;
38
39
40 -----
41 -- Table `conjuntos_reservas`.`USERS`
42 -----
43 CREATE TABLE IF NOT EXISTS `conjuntos_reservas`.`USERS` (
44   `cod_rol` INT NOT NULL,
45   `cod_user` INT NOT NULL AUTO_INCREMENT,
46   `cod_house` INT NOT NULL,
47   `user_name` VARCHAR(45) NOT NULL,
48   `user_lastname` VARCHAR(45) NOT NULL,
49   `user_birthday` DATE NOT NULL,
50   `user_id` VARCHAR(45) NOT NULL,
51   `user_email` VARCHAR(45) NOT NULL,
52   `user_pass` VARCHAR(45) NOT NULL,
53   PRIMARY KEY (`cod_user`))
54 ENGINE = InnoDB;
55
56 -----
57 -- Table `conjuntos_reservas`.`PLACES`
58 -----
59 CREATE TABLE IF NOT EXISTS `conjuntos_reservas`.`PLACES` (
60   `cod_place` INT NOT NULL AUTO_INCREMENT,
61   `place_name` VARCHAR(45) NOT NULL,
62   PRIMARY KEY (`cod_place`))
63 ENGINE = InnoDB;
64
65
66 -----
67
68 -----
69
```

```
81 -----
82 -- Table `conjuntos_reservas`.`BOOKING`
83 -----
84 CREATE TABLE IF NOT EXISTS `conjuntos_reservas`.`BOOKING` (
85   `booking_date` DATE NOT NULL,
86   `cod_booking` INT NOT NULL AUTO_INCREMENT,
87   `cod_user` INT NOT NULL,
88   `cod_place` INT NOT NULL,
89   `booking_status` ENUM('approved', 'pending', 'rejected') NOT NULL DEFAULT 'pending',
90   PRIMARY KEY (`cod_booking`),
91   INDEX `cod_user_idx` (`cod_user` ASC) VISIBLE,
92   INDEX `cod_place_idx` (`cod_place` ASC) VISIBLE,
93   CONSTRAINT `cod_user`
94     FOREIGN KEY (`cod_user`)
95     REFERENCES `conjuntos_reservas`.`USERS` (`cod_user`)
96     ON DELETE CASCADE
97     ON UPDATE CASCADE,
98   CONSTRAINT `cod_place`
99     FOREIGN KEY (`cod_place`)
100    REFERENCES `conjuntos_reservas`.`PLACES` (`cod_place`)
101    ON DELETE CASCADE
102    ON UPDATE CASCADE)
103 ENGINE = InnoDB;
104
105 -----
106
```


8. SENTENCIAS DML

Las sentencias DML se utilizan para manipular los datos dentro de las tablas. Estas sentencias afectan los datos almacenados en las tablas y permiten realizar operaciones como insertar, actualizar, eliminar y consultar datos.

Datos de la tabla casa, roles, usuarios y lugares

```
1  INSERT INTO `house` (`cod_house`, `house_name`) VALUES
2  (1, 'CASA UNO'),
3  (2, 'CASA DOS'),
4  (3, 'CASA TRES'),
5  (4, 'CASA CUATRO'),
6  (5, 'NO TIENE CASA');

INSERT INTO `roles` (`cod_rol`, `rol_name`) VALUES
(1, 'ADMIN'),
(2, 'VIGILANTE'),
(3, 'HABITANTE'),
(4, 'ARRENDATARIO');

INSERT INTO `places` (`cod_place`, `place_name`) VALUES
(1, 'BBQ'),
(2, 'SALON COMUNAL');

1  INSERT INTO `users` (`cod_rol`, `cod_user`, `cod_house`, `user_name`, `user_lastname`, `user_birthday`, `user_id`, `user_email`, `user_pass`, `user_phone`, `user_state`) VALUES
2  (1, 1, 5, 'maribel', 'perez', '1986-02-15', '1001010110', 'maribel@gmail.com', '8cb2237d0679ca88db6464eac60da96345513964', '31022541678', 1),
3  (2, 2, 5, 'Luis', 'Parra', '1977-10-06', '99884568', 'parra@gmail.com', '7c4a8d09ca3762af61e59520943dc26494f8941b', '3022548972', 1),
4  (3, 3, 1, 'joseph', 'varon', '2002-09-06', '1022524565', 'joseph@gmail.com', '7c4a8d09ca3762af61e59520943dc26494f8941b', '123456', 1),
5  (4, 4, 4, 'juan', 'lopez', '1999-09-27', '15151515', 'juan@gmail.com', '7c4a8d09ca3762af61e59520943dc26494f8941b', '3222156874', 1),
6  (3, 5, 2, 'luisa', 'martinez', '2001-01-01', '1111111111', 'luisa@gmail.com', '7c4a8d09ca3762af61e59520943dc26494f8941b', '123456', 1);
```

9. DICCIONARIO DE DATOS

Un diccionario de datos es una herramienta crucial para la gestión y diseño de bases de datos. Proporciona una descripción detallada de los datos, sus estructuras, relaciones y reglas, facilitando el diseño, desarrollo, mantenimiento y comprensión del sistema de datos. Es esencial para asegurar una gestión eficiente y coherente de la información en cualquier sistema de base de datos.

BOOKING

Column name	Data Type	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment
booking_date	DATE		✓							
cod_booking	INT	✓	✓					✓		
cod_user	INT		✓							
cod_place	INT		✓							
booking_status	ENUM('approved', 'pending', 'rejected')	✓								

HOUSE

Column name	Data Type	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment
cod_house	INT	✓	✓					✓		
house_name	VARCHAR(45)		✓							

PLACES

Column name	Data Type	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment
cod_place	INT	✓	✓					✓		
place_name	VARCHAR(45)		✓							

ROLES

Column name	Data Type	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment
cod_rol	INT	✓	✓					✓		
rol_name	VARCHAR(20)		✓							

users

Column name	Data Type	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment
cod_rol	INT(11)		✓							
cod_user	INT(11)	✓	✓							
cod_house	INT(11)		✓							
user_name	VARCHAR(45)	✓								
user_lastname	VARCHAR(45)	✓								
user_birthday	DATE	✓								
user_id	INT(11)	✓								
user_email	VARCHAR(45)	✓								
user_pass	VARCHAR(45)	✓								
user_phone	VARCHAR(80)	✓								
user_state	TINYINT(4)	✓								

10. EJECUCION DE SENTENCIAS EN WORKBENCH

En esta ejecución de las sentencias tanto ddl y dml buscamos poder agilizar el proceso de la carga de base de datos. Esto para que el administrador tenga un usuario universal registrado y poder así ingresar

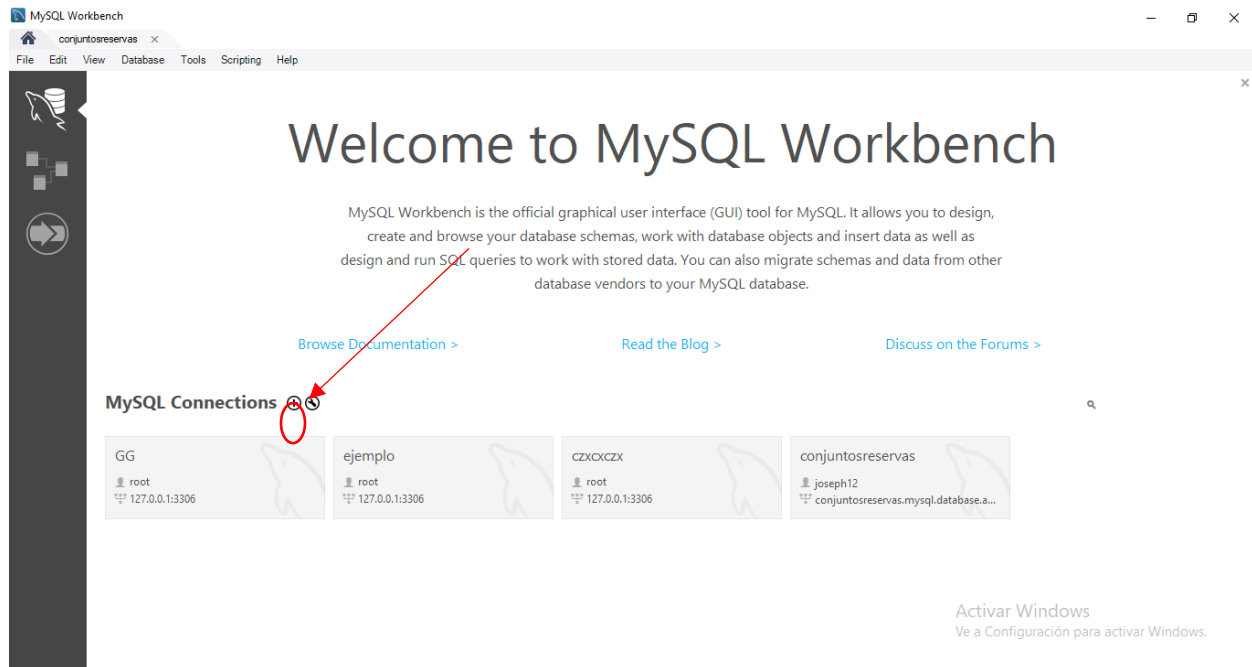
10.1 Instalación del repositorio de git

- Nos dirigimos a nuestro navegador de preferencia y entramos al repositorio con este url https://github.com/joseph12n/conjuntos_residenciales-main y clonamos el proyecto usando la herramienta de git con su comando de:

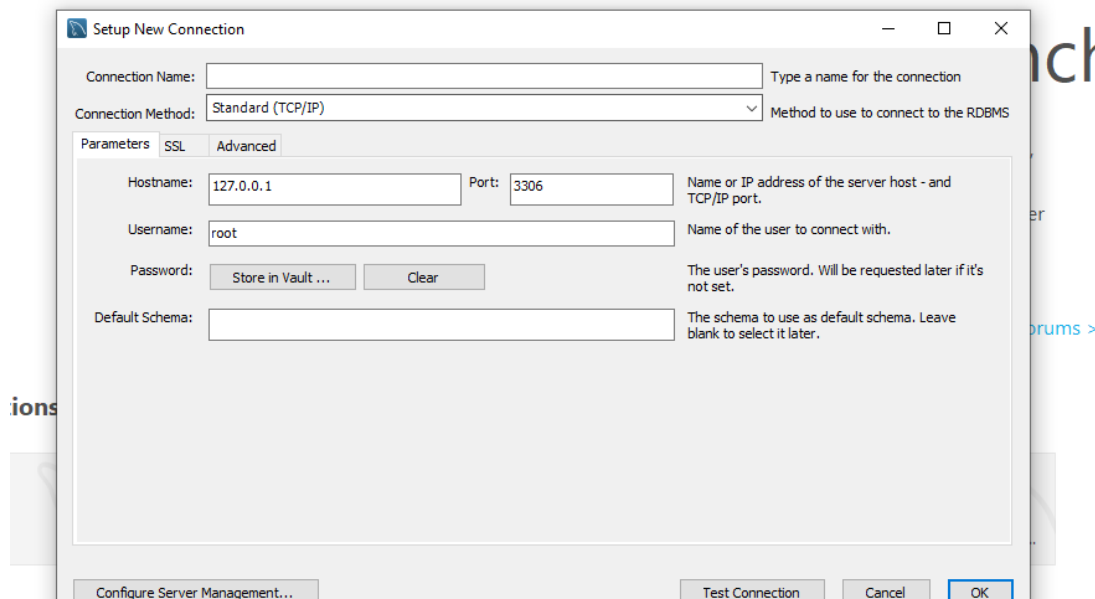
git clone https://github.com/joseph12n/conjuntos_residenciales-main.

10.2 CONEXIÓN EN MYSQL WORKBENCH V8.0CE

- Iniciamos el programa e ingresamos al icono del + para poder registrar una nueva conexión.



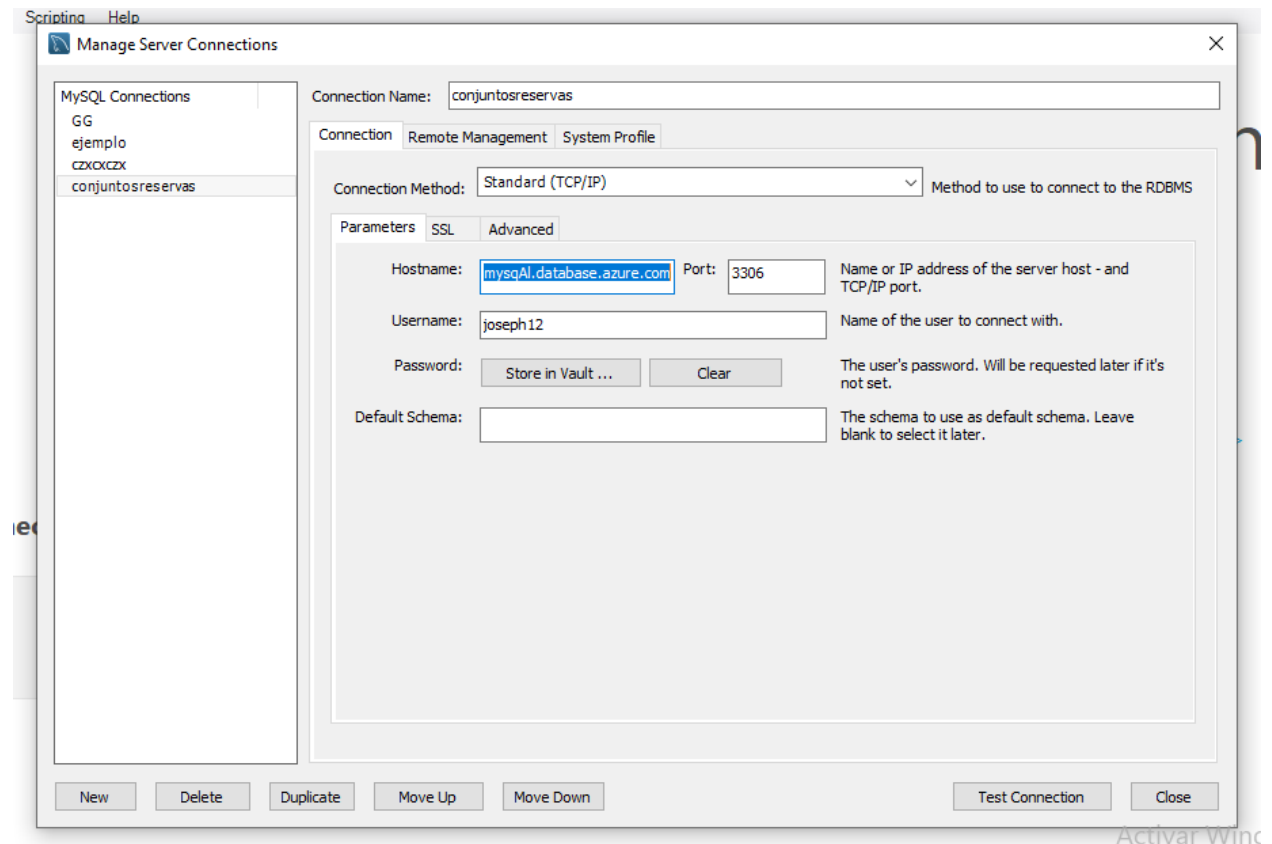
- Al darle clic se nos abrirá el modulo para realizar la conexión con. Esto ya que la base de datos esta alojada en Azure.



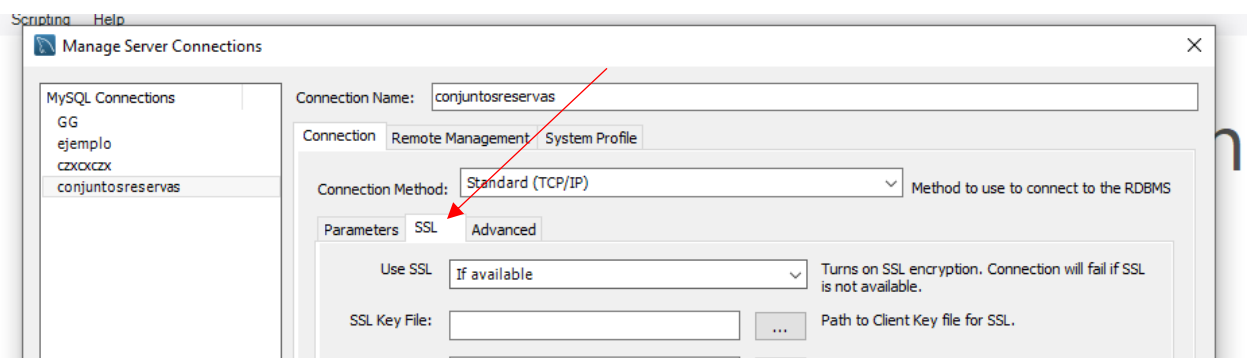
- Inicio de la conexión. Necesitaremos ingresar un nombre a la conexión a gusto del que vaya a iniciar la conexión.
- El método para usar se deja en Standart(TPC/IP). Es el que viene por defecto.
- En el hostname ingresaremos la siguiente url conjuntosreservas.mysql.database.azure.com

- El port se deja el que viene por defecto. En este caso. Port 3306
- En user name ingresamos el usuario admin. En este caso es joseph12
- En password ingresamos al botón que dice store in vault e ingresamos la contraseña que es conjuntosR12

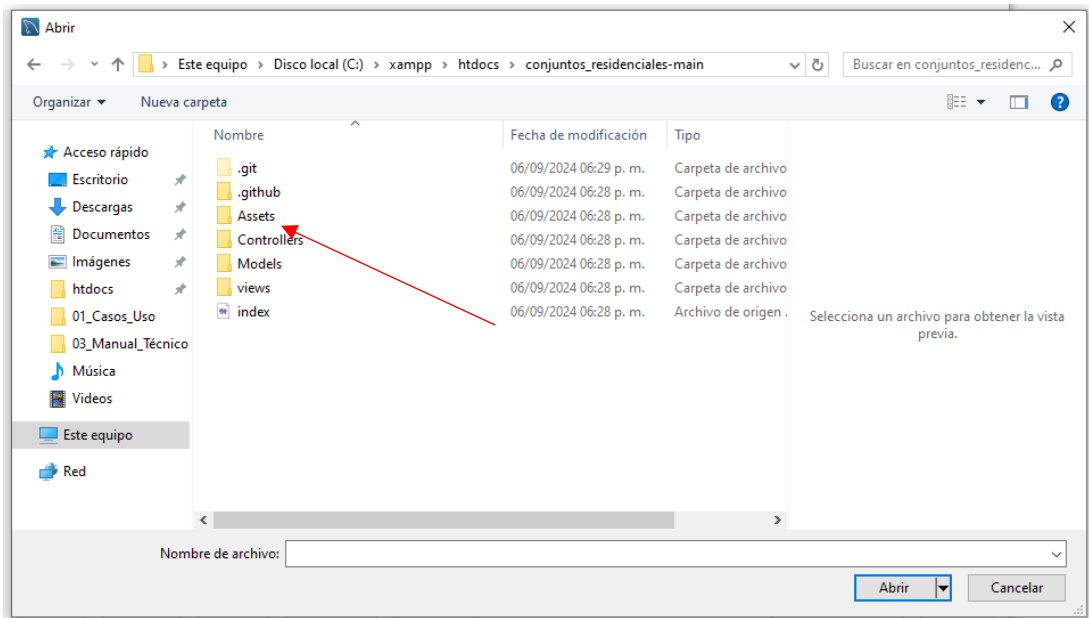
Así debería quedarnos por el momento.



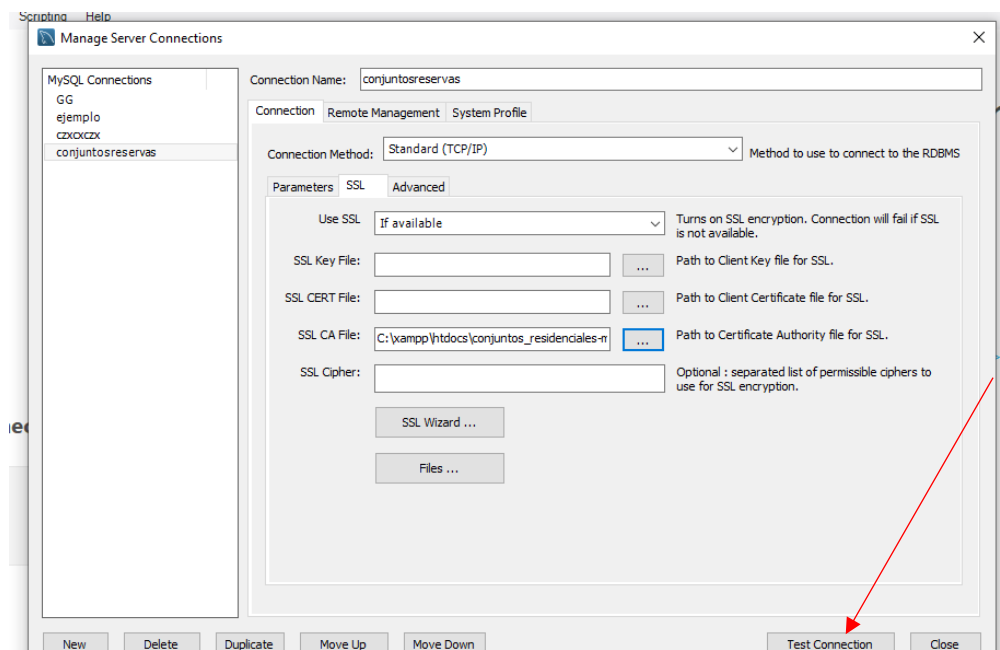
- Ahora ingresamos ahí dentro a la pestaña SSL y ubicamos el campo que dice SSL CA File.



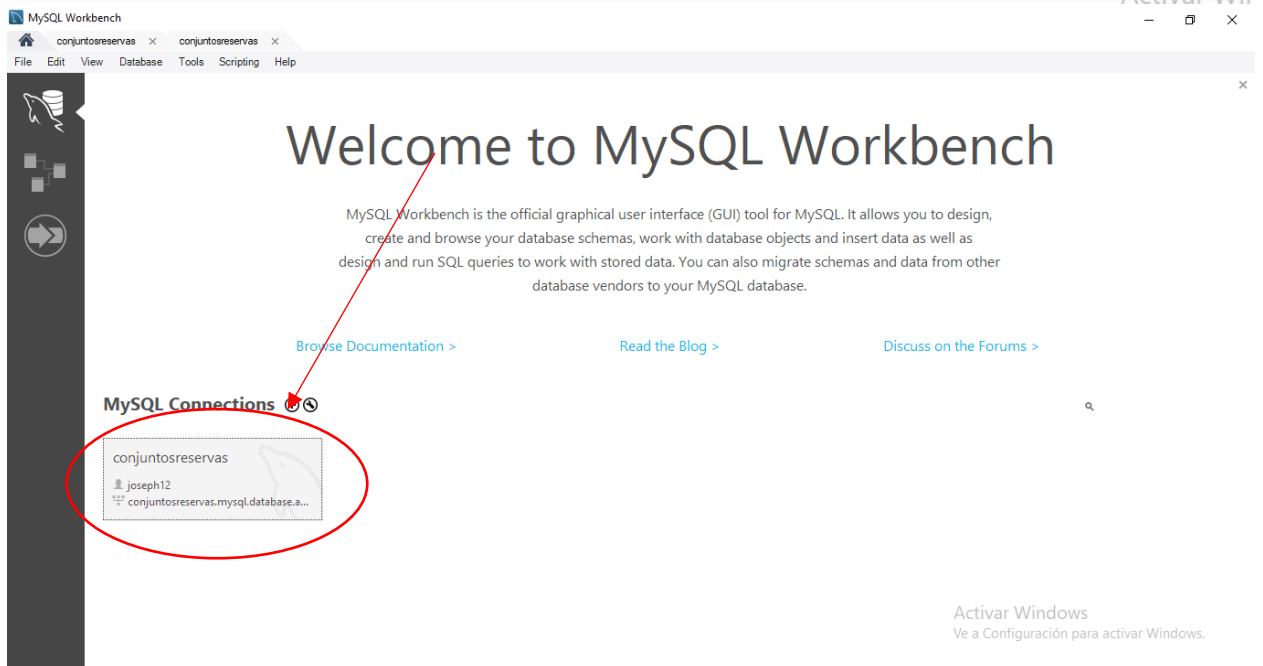
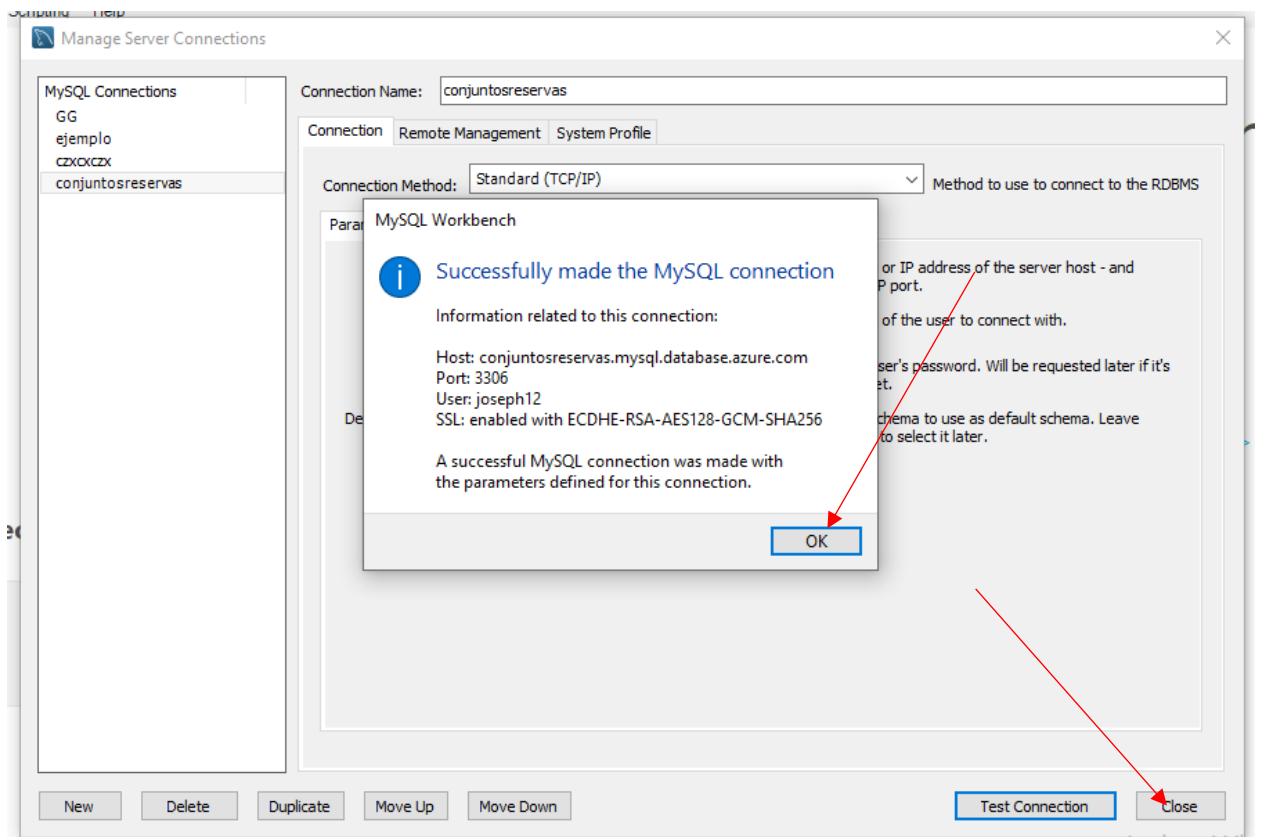
- Al ingresar debemos ubicar el repositorio clonado. Al ubicarlo ingresamos y deberíamos encontrar la carpeta llamada assets.



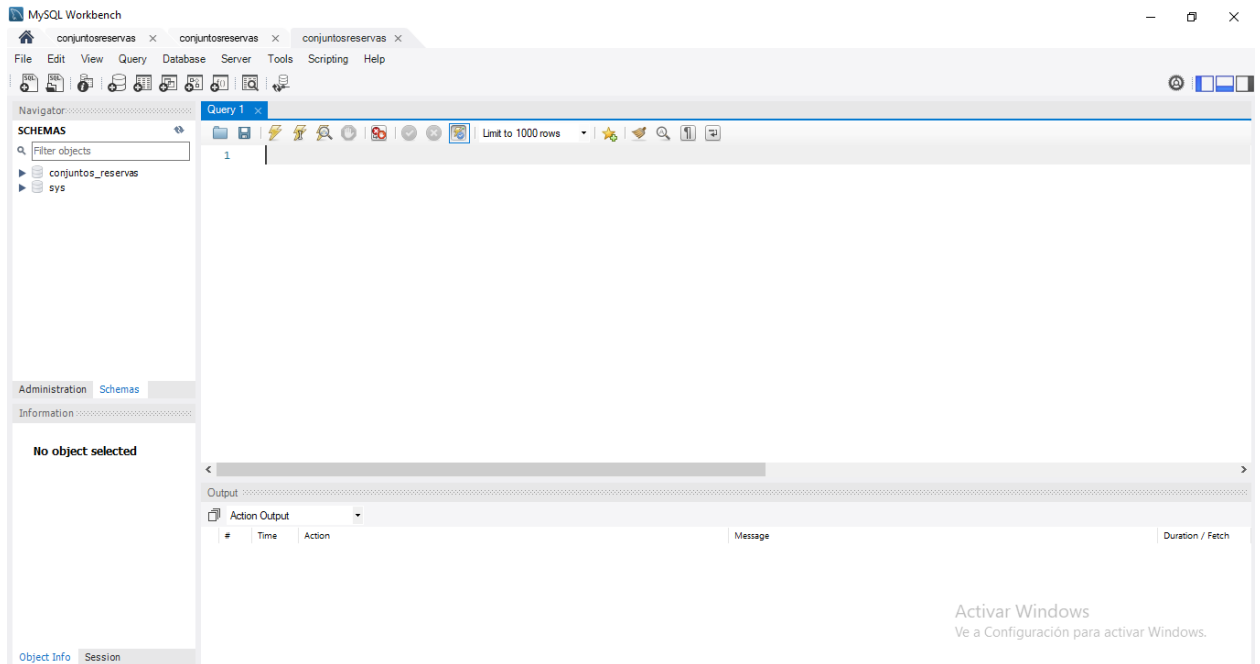
- Ingresamos y deberíamos encontrar un archivo llamado DigiCertGlobalRootCA.crt y lo seleccionamos. Ese es el archivo que debemos alojar en SSL CA File.
- Ahora se oprime el botón de test connection y esperamos a que se valide. Posibles errores concurrentes pueden ser por el nombre que dedicamos al nombre de la conexión o hacia la contraseña ingresada, pero siguiendo todos los pasos debería ser una conexión exitosa.



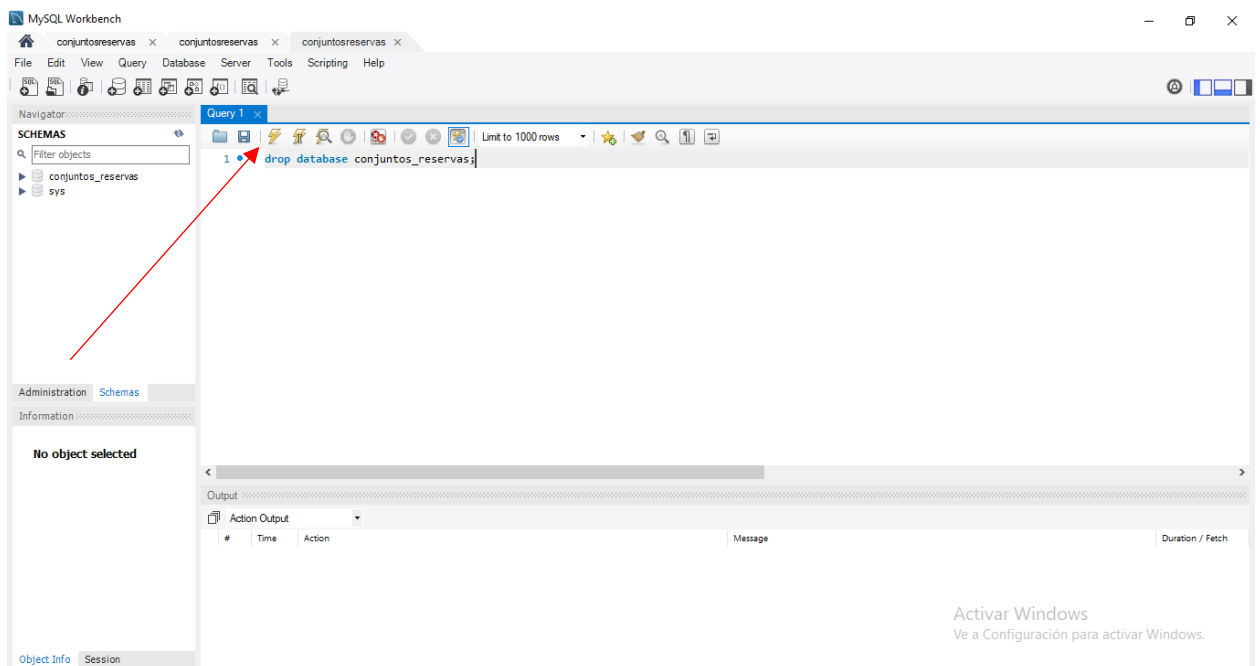
- Nos debería aparecer este mensaje. Seleccionamos ok y cerramos. Ahora la conexión existe y ya deberíamos ingresar. Nos pedirá la contraseña de nuevo pero podemos dejar checkeado el mensaje para que no nos la pida de nuevo



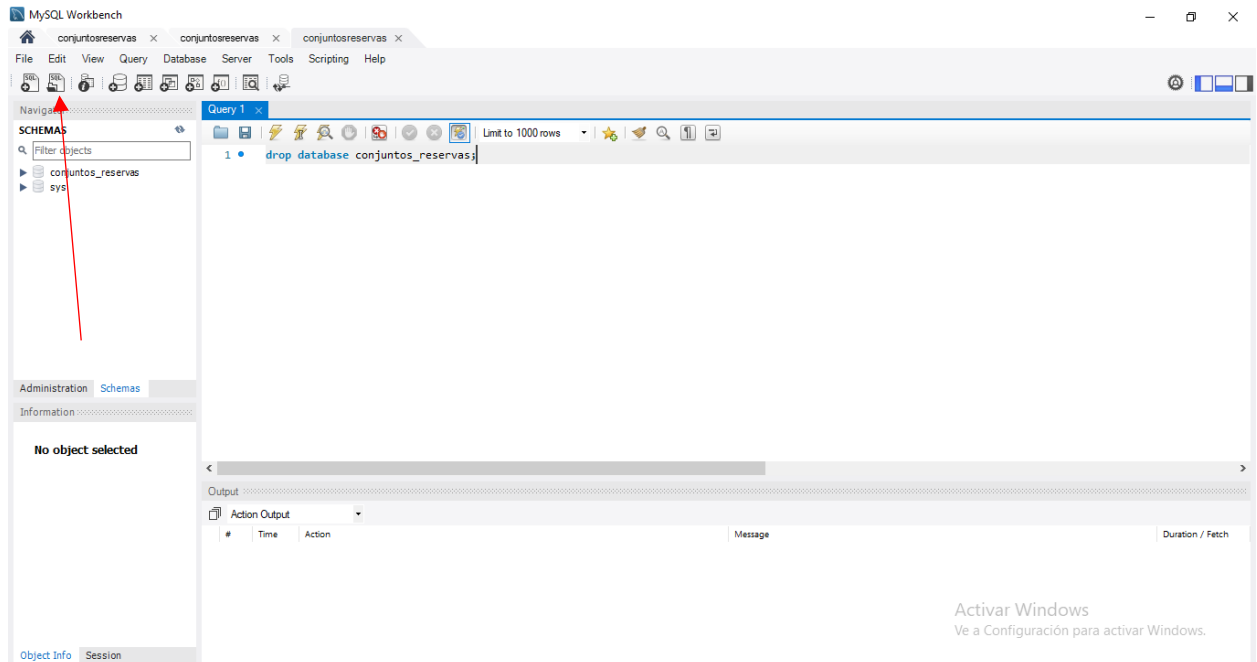
- Ingresamos y nos aparecerá la siguiente interfaz



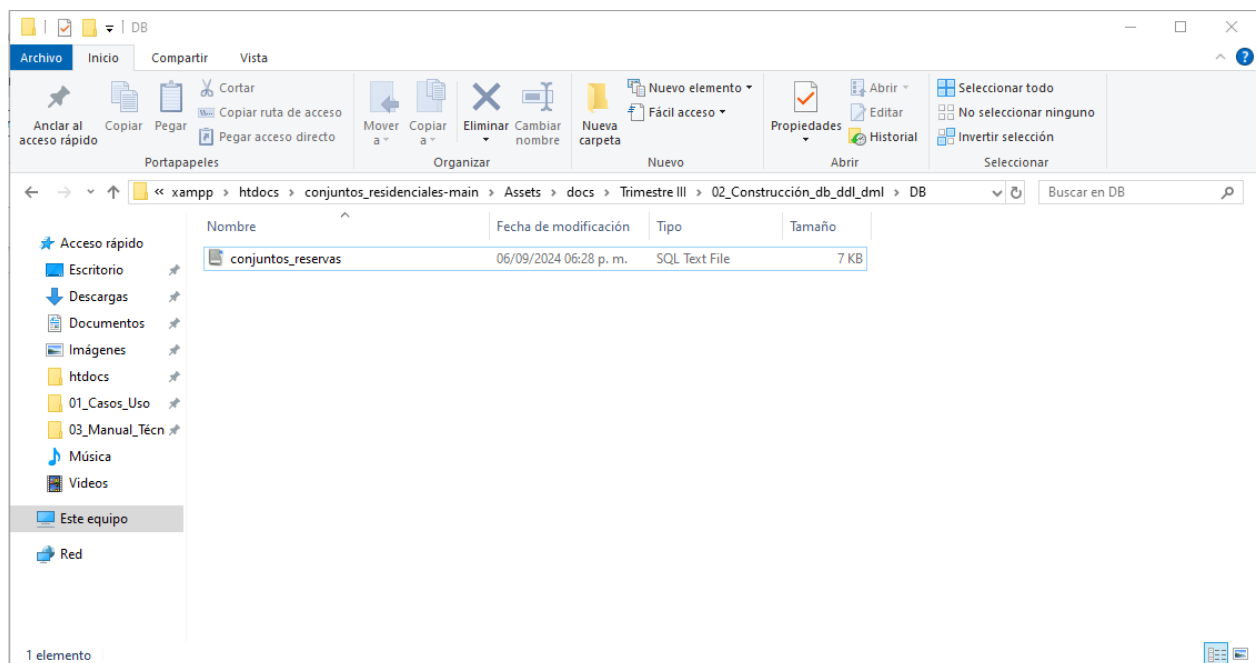
- Ingresamos la primera consulta para poder borrar la base de datos. Esto se hace por comodidad, en caso de urgencia en la que se deba eliminar. Esto se ejecutaría con `drop database conjuntos_reservas;` (para ejecutarlo debe estar la base de datos creada).
- Cuando se quiera ejecutar, simplemente se oprime el primer rayo que nos aparece en la parte superior izquierda. Así debería quedar.



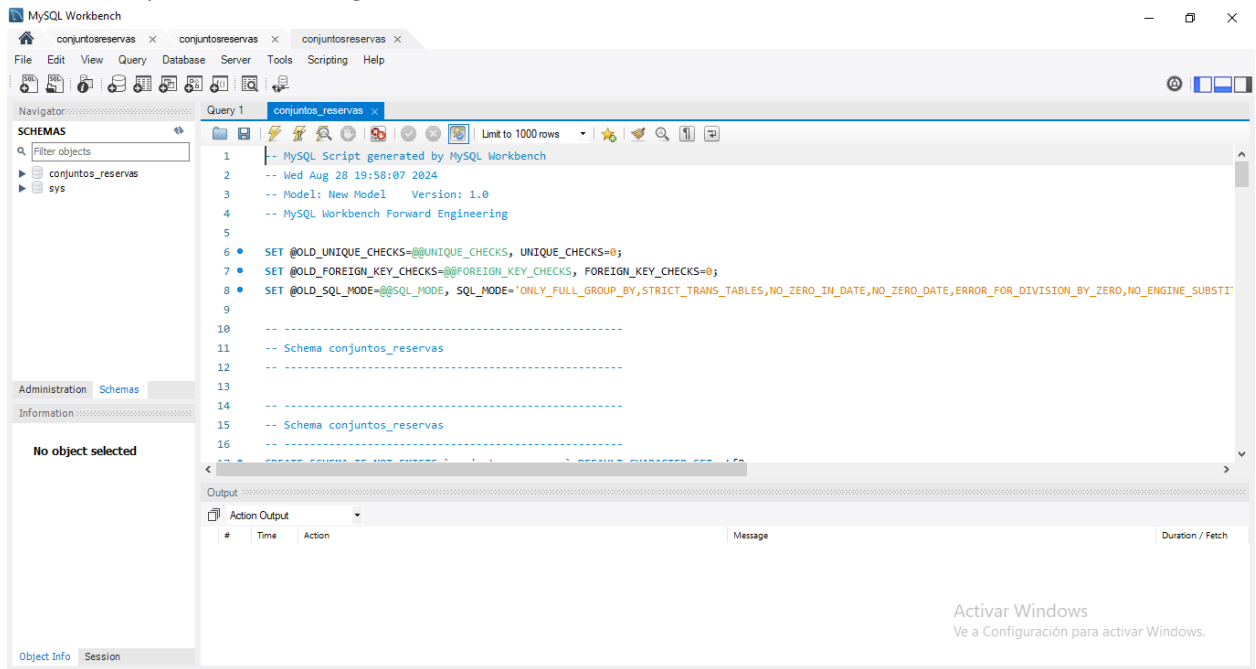
- Ya teniendo esa consulta ya iniciaríamos a cargar las sentencias ddl y dml en el icono que contiene una carpeta y el nombre sql.



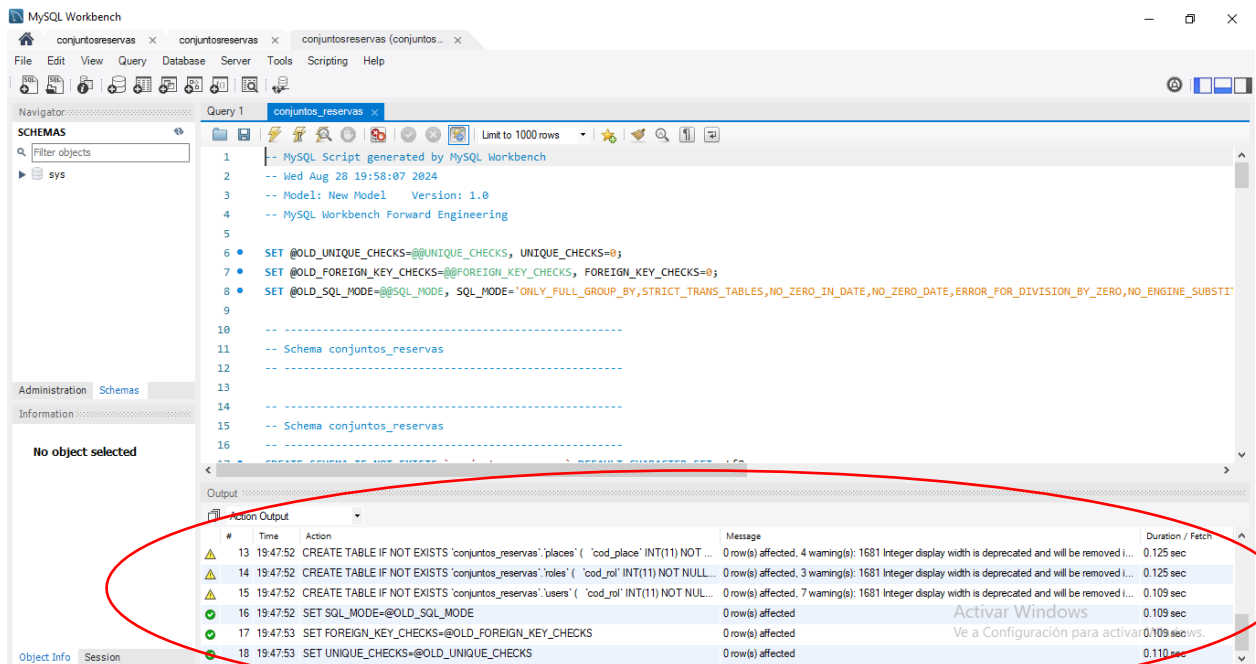
- Estas están alojadas en el mismo repositorio previamente clonado. Nos dirigimos a assets/docs/trimestre III y seleccionamos la carpeta que se llama 02_Construcción_db_ddl_dml y seleccionamos la carpeta que se llama DB Y abrimos el archivo que se llama conjuntos_reservas.



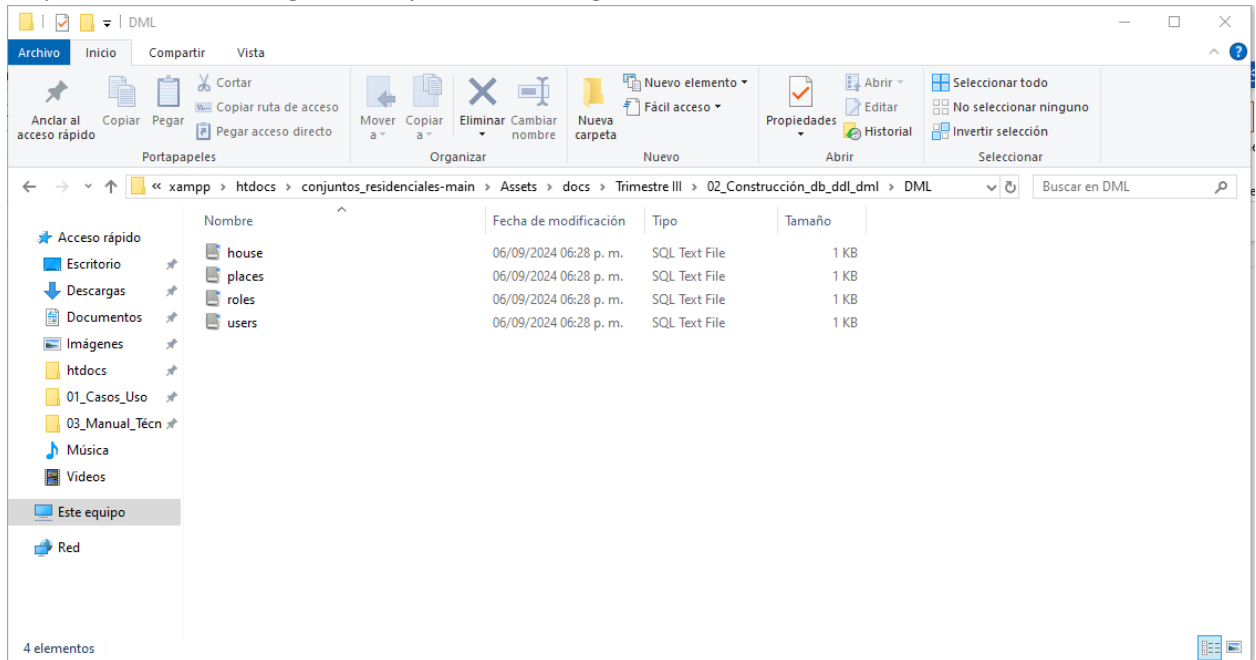
- Lo abrimos y se los verá lo siguiente.



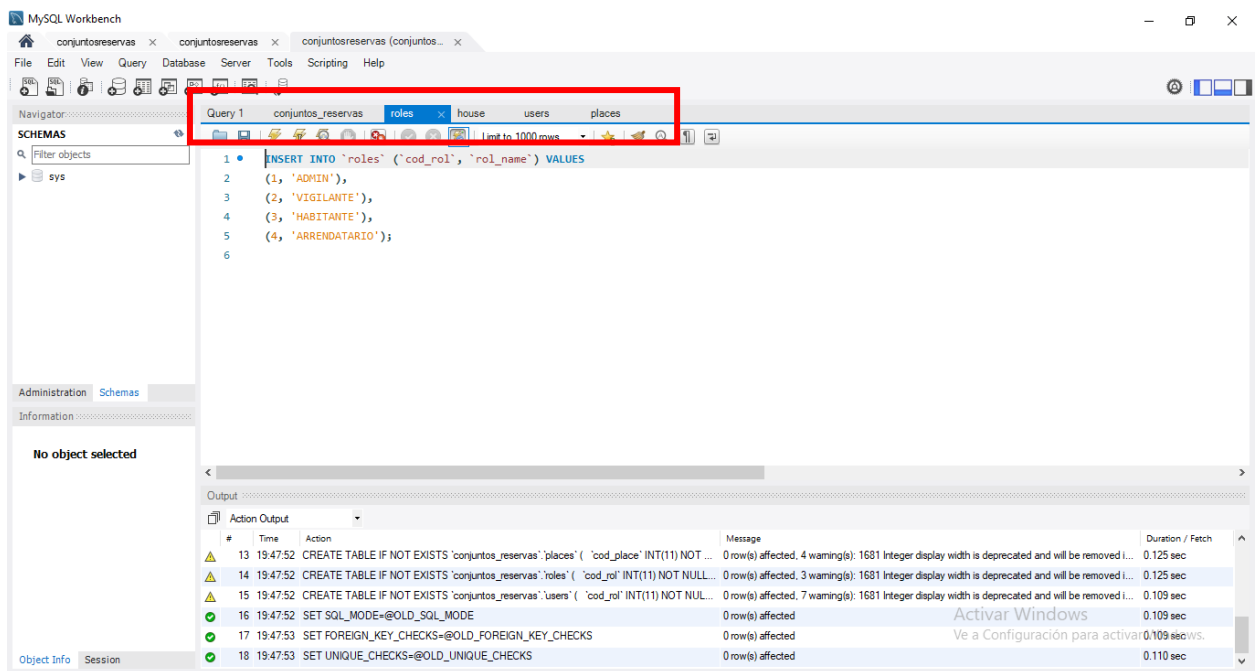
- Ya tendríamos la base de datos cargada. Para que se ejecute en azure simplemente le oprimimos al rayo y en la consola de la parte de abajo debería aparecer todos los iconos en verde y amarillo. Si hay uno en rojo lo mas probable es que se exporto mal la base de datos, se recomienda volverla a exportar.



- Ya tendríamos la bd cargada en azure. Ahora seria cargar los datos para estas tablas o bien conocido como sentencias dml. El proceso es el mismo con el que se cargo la base de datos solo que nos tendríamos que devolver y ubicar ahí mismo en 02_Construcción_db_ddl_dml la carpeta llamada dml, ingresamos y veríamos lo siguiente



- Realizamos el mismo proceso que con la carga de base de datos, abrimos uno por uno las carpetas en su preferencia en el orden de roles, house, users y places para no tener confusiones. Esto se debe ya que no se pueden registrar usuarios si no hay casas ni roles. Nos debería quedar así



- Ejecutamos en ese orden y ya deberíamos tener los datos llenos de cada tabla. En la tabla de reservas si se hace de manera manual en la vista ya que esta hecho para los usuarios. Este seria el paso a paso para la ejecución de la base de datos.