



PT-MCBD-01.Manual de Configuración BD

SPORT FULL
1.0

HISTORIAL DE REVISIÓN

Versión	Fecha Elaboración	Responsable Elaboración	Fecha Aprobación	Responsable Aprobación
1.0	15/04/2024	Equipo		

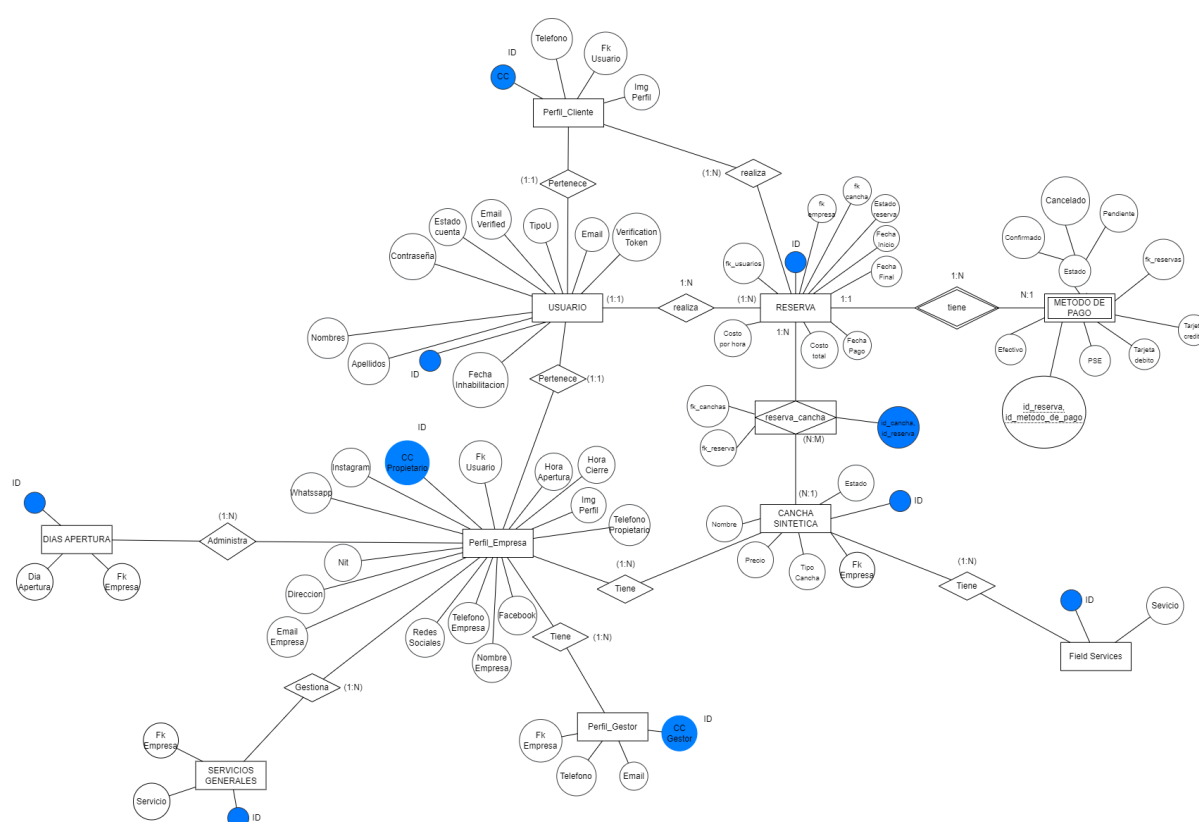
CAMBIOS RESPECTO A LA VERSIÓN ANTERIOR

VERSIÓN	MODIFICACIÓN RESPECTO VERSIÓN ANTERIOR
1.0	Creacion de documento

Tabla de Contenido.

1. Introducción
2. Alcance
3. Modelo Entidad Relación (MER)
4. Diccionario de Datos
5. Modelo Relacional o Estructura de Documentos

6. Justificación Sistema gestos de bases de datos seleccionado
7. Requisitos de configuración
8. Scripts
9. Configuración y ejecución de la base de datos
10. Otras consideraciones



1. Introducción

La creación de una aplicación dedicada a la gestión de canchas sintéticas surge como respuesta a la necesidad de optimizar y modernizar el proceso de reserva y administración de instalaciones deportivas. Este proyecto involucra a diversos actores, desde propietarios de canchas hasta usuarios finales, y busca abordar las limitaciones actuales mediante el aprovechamiento de la tecnología, incluyendo la integración de funcionalidades de GPS. La implementación de esta aplicación no solo mejorará la eficiencia en la gestión de las canchas, sino que también proporcionará a los usuarios una herramienta más conveniente y accesible para reservar y disfrutar de las instalaciones deportivas.

2. Alcance

La aplicación para gestión de canchas sintéticas busca optimizar la reserva y administración de estas instalaciones mediante funcionalidades como GPS, pagos en línea y gestión de torneos. Está dirigida a deportistas, equipos deportivos, instituciones educativas, empresas y propietarios de canchas. Ofrece reserva por hora o partido, visualización de disponibilidad, pagos en línea, geolocalización, calificación de canchas, historial de reservas, gestión de torneos y eventos, notificaciones y recordatorios. Se espera que la aplicación mejore la eficiencia, reduzca trámites, aumente la transparencia y controle para usuarios y propietarios, mejore la experiencia de los usuarios y promueva el deporte local. Se planea escalarla a nivel departamental, regional y nacional. La aplicación contribuirá a la modernización del sector deportivo local, impulsando la práctica del deporte y generando nuevas oportunidades para la gestión eficiente de las instalaciones deportivas.

3. Modelo Entidad Relación (MER)

4. Diccionario de Datos..

DDL : Lenguaje de definición de datos (Data Definition Language), define las estructuras que almacenarán los datos y los procedimientos o funciones que permitan consultarlos. CREATE, ALTER, DROP, TRUNCATE, RENAME.

SQL: (Structured Query Language) Lenguaje de consulta estructurado, Lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

Modelo de base de datos Lógicos: orientados a objetos o a registros. Un modelo de datos lógico describe los datos y las relaciones en detalle a un nivel muy alto. Básicamente incluye las entidades y las relaciones entre ellas junto con los atributos de cada entidad. (Es independiente del SGBD).

Modelo Relacional: Permite visualizar la estructura de una base de datos y las relaciones entre los datos. Consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos.

Diagrama o modelo de entidad relación: Permite representar gráficamente la estructura lógica de la base de datos.

Entidad: Es un objeto del mundo real, que contiene características, se escribe en singular y con MAYÚSCULA SOSTENIDA.

Entidad Concreta: PERSONA, EMPLEADO, CASA, AUTO...

Entidad Abstracta: CUENTA_BANCARIA, CURSO...

Atributos: Son las características o propiedades de una entidad, son representados por elipses.

Elipses: Representan a los atributos de una clase

Elipse simple: atributo simple, formado por un único valor.

Elipse doble: atributo multivalorado. (Puede tener varias direcciones, número de teléfonos...).

Elipse discontinua: atributo derivado (permite cálculos).

Uno a uno (1,1): una entidad de A se puede relacionar con una de B.

Uno a varios (1, m): una entidad A se puede relacionar con una o varias de B.

Varios a uno (m,1): varias entidades de A se pueden relacionar con una de B.

Varios a varios (m, n): varias entidades de A se pueden relacionar con varias entidades de B.

Rectángulo simple: conjunto de entidades fuerte

Rectángulo doble: conjunto de entidades débil.

Rombo simple: conjunto de relaciones.

Rombo doble: conjunto de relaciones en una entidad débil.

Rectángulos: Representan a las Entidades fuertes y cuando son dobles en Entidades débiles.

Rombos: Relación entre entidades.

Rombo simple: conjunto de relaciones. **Rombo doble:** conjunto de relaciones en una entidad débil.

DICCIONARIO DE DATOS			
Tabla	RESERVA		
Fecha	22-04-2024		
Proposito	Mostrar los datos de reservacion de las canchas		
Campo	Tipo	Longitud	Descripción
idReserva	INT	11	identificador unico de cada reserva
fecha_inicio	DATE	10	informacion fecha de inicio de la reserva
fecha_final	DATE	10	Información de la fecha fin reserva
metodo_pago	ENUM	13	Se describe los metodos de pago disponible
costo_hora	INT	11	Se declara el valor monetario de la reserva
fecha_pago	DATE	10	se indica cuando se paga ese valor.
estadUsuario_id	INT	17	Identificador unico de cada usuario

DICCIONARIO DE DATOS			
Tabla	USUARIOS		
Fecha	19/04/2024		
Proposito	Gestionar datos de usuarios para optimizar reservas, análisis y		
Campo	Tipo	Longitud	Descripción
Id	INT	11	Identificador único para cada usuario
Nombre	VARCHAR	45	Nombre del usuario
Email	VARCHAR	45	Dirección de correo electrónico del usuario

Diccionario de datos (canchas)			
Tabla	cancha_sintetica		
Fecha	20/04/2024		
Propósito	Guardar el nombre y dirección de cada cancha		
Campo	Tipo	Longitud	Descripción
idCancha	INT	11	Guardar el identificador de la cancha
nombre	VARCHAR	45	Guardar el nombre de la cancha
direccion_idCancha_direccion	VARCHAR	50	Guardar la dirección de la cancha
propietario_idpropietario	INT	11	Guardar la llave foranea única del propietario

DICCIONARIO DE DATOS			
Tabla	PROPIETARIO		
Fecha	19-04-2024		
Proposito	Identificar quién es responsable de la gestión y la integridad de		
Campo	Tipo	Longitud	Descripción
idPropietario	INT	11	Identificador unico del propietario
nombre	VARCHAR	45	Nombre del propietario
email	VARCHAR	45	Correo electronico del propietario
telefono	INT	11	Telefono del propietario

DICCIONARIO DE DATOS			
Tabla	RESERVA_HAS_CANCHA_SINTETICA		
Fecha	22/04/2024		
Proposito	Establecer una relación entre las reservas realizadas y las canchas sintéticas disponibles en un sistema o plataforma de gestión de reservas		
Campo	Tipo	Longitud	Descripción
idReserva	INT	11	Identificador unico de cada reserva
sintetica_id	INT	11	Identificador unico de la cancha sintetica
id_cancha_id_reserva	INT	11	Identificador unico de la cancha y la reserva

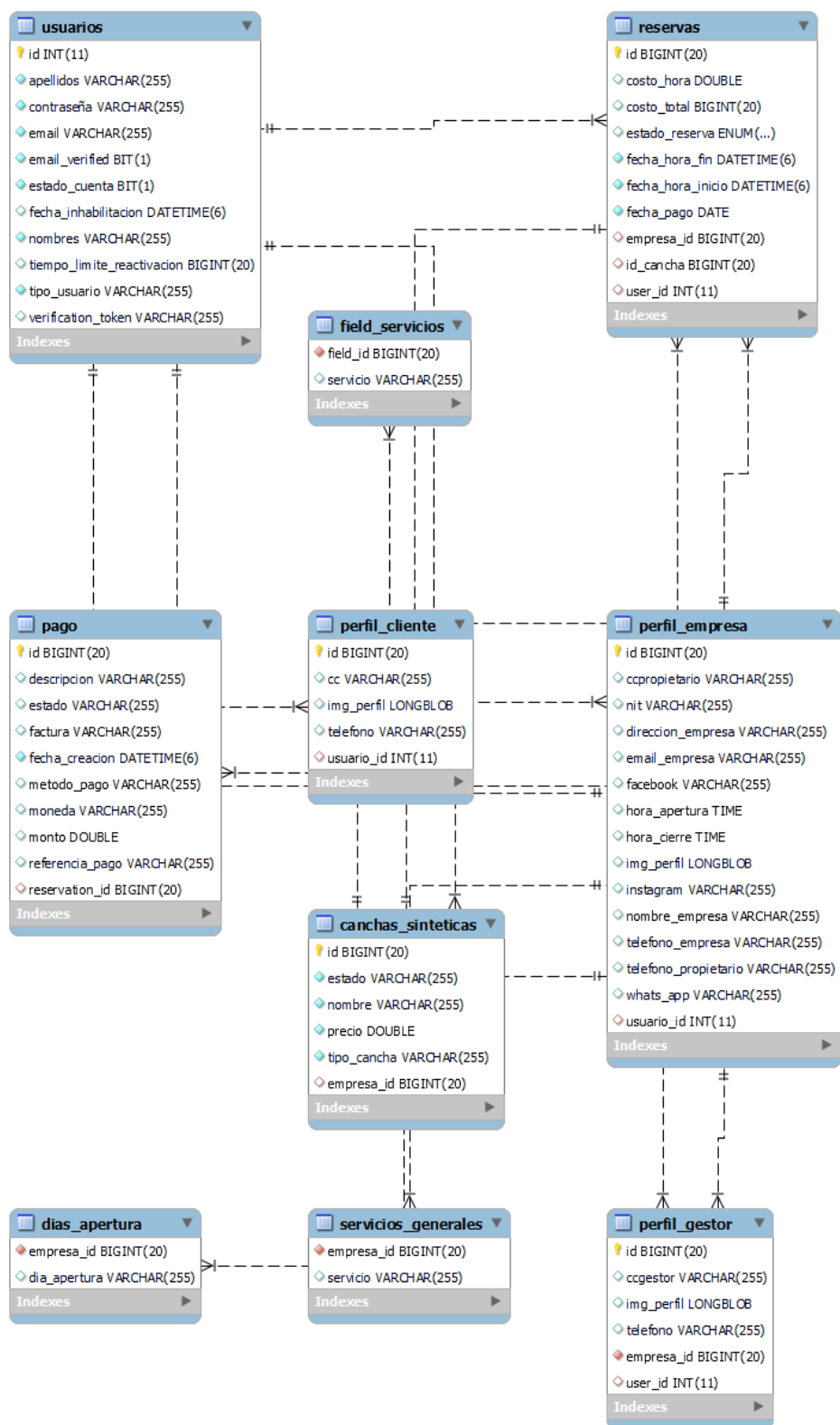
DICCIONARIO DE DATOS			
Tabla	Metodo_Pago		
Fecha	09-04-2024		
Proposito	Establecer la relacion entte los metodos de pago y las reservas de canchas		
Campo	Tipo	Longitud	Descripción
IdRerserva_Metodo_pago	INT	12	Identificador unico de reserva con el metodo de pago
Efectivo	VARCHAR	45	forma de pago
Nequi	VARCHAR	45	Metofa para hacer pagos
pse	VARCHAR	45	Metofa para hacer pagos
Reserva_IdrERVA	INT	11	identificador unico de cada reserva
Estado_idReserva_Estado 1	INT		identificadir unico de la reserva con el estado
Tarjeta_debito	VARCHAR	45	forma de pago
Tarjeta_credito	VARCHAR	45	forma de pago

DICCIONARIO DE DATOS			
Tabla	ESTADO		
Fecha	29/04/2024		
Proposito	Describir el estado o la condición de un objeto o entidad en un sistema.		
Campo	Tipo	Longitud	Descripción
IdReserva_Estado	INT	11	Identificador unico de cada estado de reserva
Cancelado	VARCHAR	45	Estado cancelado de la reserva
Pendiente	VARCHAR	45	Estado pendiente de la reserva
Confirmado	VARCHAR	45	Estado confirmado de la reserva

DICCIONARIO DE DATOS			
Tabla	direccion		
Fecha	29/04/2024		
Proposito	Almacenar información sobre la ubicación de las canchas para que esta pueda ser utilizada para diversos fines		
Campo	Tipo	Longitud	Descripción
idCancha_direccion	VARCHAR	50	Identificador único de la cancha
direccion	VARCHAR	45	Direccion de la cancha

5

. Modelo Relacional o Estructura de Documentos.



6. Justificación Sistema gestor de bases de datos seleccionado

MySQL sería una elección ideal como sistema gestor de base de datos para la aplicación de gestión de canchas sintéticas por diversas razones. En primer lugar, su robusta capacidad de escalabilidad permite manejar eficientemente grandes volúmenes de datos y tráfico, lo cual es fundamental para una plataforma que podría experimentar múltiples reservas simultáneas y un crecimiento rápido de usuarios. Además, su rendimiento rápido y eficiente garantiza tiempos de respuesta ágiles en las transacciones de reserva, mejorando así la experiencia del usuario. En cuanto a la seguridad, MySQL ofrece diversas características para proteger los datos, como la autenticación de usuarios, cifrado de datos en reposo y en tránsito, así como control de acceso a nivel de usuario y de base de datos. Estas medidas ayudan a salvaguardar la información confidencial de los usuarios y propietarios de las canchas, como detalles de reserva y datos de pago electrónico. Además, MySQL cuenta con una amplia comunidad de usuarios y un sólido soporte técnico, lo que proporciona recursos y ayuda para el desarrollo, mantenimiento y resolución de problemas de la base de datos. En resumen, MySQL ofrece una combinación de escalabilidad, rendimiento, seguridad y soporte que lo convierten en una opción confiable y adecuada para respaldar la gestión eficiente y segura de la aplicación de reservas de canchas sintéticas.

7. Requisitos de Configuración

1. Motor de Base de Datos (MySQL):

- Descarga e instala la última versión estable de MySQL Server desde el sitio web oficial de MySQL: [MySQL Downloads](#).
- Asegúrate de seleccionar la versión compatible con tu sistema operativo (Windows, Linux, macOS) y arquitectura (32-bit o 64-bit).

2. Cliente de Base de Datos:

- Se recomienda utilizar MySQL Workbench como cliente de base de datos para interactuar con el servidor MySQL.
- Descarga e instala MySQL Workbench desde el sitio web oficial de MySQL: [MySQL Workbench Downloads](#).

3. Servidor de Base de Datos:

- Verifica que el servidor donde instalarás MySQL cumpla con los siguientes requisitos mínimos:
 - Sistema operativo compatible: Windows, Linux o macOS.
 - Espacio en disco suficiente para instalar el servidor MySQL y almacenar los datos.
 - Memoria RAM adecuada para el tamaño de la base de datos y el número de conexiones concurrentes esperadas.
 - Procesador con capacidad suficiente para manejar la carga de trabajo.

4. Acceso a la Documentación:

- Consulta la documentación oficial de MySQL para obtener información detallada sobre la instalación, configuración y administración del servidor MySQL: [MySQL Documentation](#).

8. Scripts

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

--
-- Schema mydb
--

--
-- Schema mydb
--

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8
USE `mydb` ;

--
-- Table `mydb`.`Usuario`
--
```

```

-----
CREATE TABLE IF NOT EXISTS `mydb`.`Usuario` (
  `idUsuario` INT NOT NULL AUTO_INCREMENT,
  `nombres` VARCHAR(45) NULL,
  `apellidos` VARCHAR(45) NULL,
  `email` VARCHAR(100) NULL,
  `tipo_u` VARCHAR(45) NULL,
  PRIMARY KEY (`idUsuario`))
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Perfil_cliente`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Perfil_cliente` (
  `id_cliente` INT NOT NULL AUTO_INCREMENT,
  `telefono` VARCHAR(10) NULL,
  `ImgPerfil` VARCHAR(45) NULL,
  `Usuario_idUsuario` INT NOT NULL,
  PRIMARY KEY (`id_cliente`),
  INDEX `fk_Perfil_cliente_Usuario_idx` (`Usuario_idUsuario`),
  CONSTRAINT `fk_Perfil_cliente_Usuario`
    FOREIGN KEY (`Usuario_idUsuario`)
      REFERENCES `mydb`.`Usuario` (`idUsuario`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Perfil_empresa`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Perfil_empresa` (
  `id_empresa` INT NOT NULL AUTO_INCREMENT,
  `nombre_empresa` VARCHAR(45) NULL,
  `CC_propietario` VARCHAR(30) NULL,
  `telefono` VARCHAR(45) NULL,
  `email_empresa` VARCHAR(100) NULL,

```

```

    `Usuario_idUsuario` INT NOT NULL,
PRIMARY KEY (`id_empresa`),
INDEX `fk_Perfil_empresa_Usuario1_idx` (`Usuario_idUsuario`
CONSTRAINT `fk_Perfil_empresa_Usuario1`
    FOREIGN KEY (`Usuario_idUsuario`)
    REFERENCES `mydb`.`Usuario` (`idUsuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Metodo_pago`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Metodo_pago` (
    `idMetodo_pago_id_reserva` INT NOT NULL AUTO_INCREMENT,
    `efectivo` VARCHAR(45) NULL,
    `nequi` VARCHAR(45) NULL,
    `pse` VARCHAR(45) NULL,
    `tarjeta_debito` VARCHAR(45) NULL,
    `tarjeta_credito` VARCHAR(45) NULL,
    `Datos_tarjeta_idDatos_tarjeta` INT NOT NULL,
PRIMARY KEY (`idMetodo_pago_id_reserva`))
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Cancha_sintetica`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Cancha_sintetica` (
    `idCancha_sintetica` INT NOT NULL AUTO_INCREMENT,
    `nombre` VARCHAR(50) NULL,
    `img` VARCHAR(300) NULL,
    `servicios` VARCHAR(1000) NULL,
    `estado` ENUM('Reservada', 'Disponible', 'Fuera de servicio')
    `precio` DECIMAL(10,2) NULL,
    `Perfil_empresa_id_empresa` INT NOT NULL,
PRIMARY KEY (`idCancha_sintetica`),

```

```

INDEX `fk_Cancha_sintetica_Perfil_empresa1_idx` (`Perfil_em
CONSTRAINT `fk_Cancha_sintetica_Perfil_empresa1`
    FOREIGN KEY (`Perfil_empresa_id_empresa`)
    REFERENCES `mydb`.`Perfil_empresa` (`id_empresa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Reserva`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Reserva` (
    `idReserva` INT NOT NULL AUTO_INCREMENT,
    `fecha_hora_inicio` DATETIME NULL,
    `fecha_hora_final` DATETIME NULL,
    `fecha_pago` DATETIME NULL,
    `estado` ENUM('Reservado', 'Pagado', 'Cancelado') NULL,
    `costo_horas` DECIMAL(10,2) NULL,
    `costo_total` DECIMAL(10,2) NULL,
    `Perfil_cliente_id_cliente` INT NOT NULL,
    `Metodo_pago_idMetodo_pago` INT NOT NULL,
    `Cancha_sintetica_idCancha_sintetica` INT NOT NULL,
    PRIMARY KEY (`idReserva`),
    INDEX `fk_Reserva_Perfil_cliente1_idx` (`Perfil_cliente_id_
    INDEX `fk_Reserva_Metodo_pago1_idx` (`Metodo_pago_idMetodo_
    INDEX `fk_Reserva_Cancha_sintetica1_idx` (`Cancha_sintetica
    CONSTRAINT `fk_Reserva_Perfil_cliente1`
        FOREIGN KEY (`Perfil_cliente_id_cliente`)
        REFERENCES `mydb`.`Perfil_cliente` (`id_cliente`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Reserva_Metodo_pago1`
        FOREIGN KEY (`Metodo_pago_idMetodo_pago`)
        REFERENCES `mydb`.`Metodo_pago` (`idMetodo_pago_id_reserv
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Reserva_Cancha_sintetica1`

```

```

        FOREIGN KEY (`Cancha_sintetica_idCancha_sintetica`)
        REFERENCES `mydb`.`Cancha_sintetica` (`idCancha_sintetica`
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Perfil_gestor`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Perfil_gestor` (
  `idPerfil_gestor` INT NOT NULL AUTO_INCREMENT,
  `CcGestor` VARCHAR(45) NOT NULL,
  `Nombres` VARCHAR(45) NOT NULL,
  `Apellidos` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `Telefono` VARCHAR(45) NOT NULL,
  `Perfil_empresa_id_empresa` INT NOT NULL,
  PRIMARY KEY (`idPerfil_gestor`),
  INDEX `fk_Perfil_gestor_Perfil_empresa1_idx` (`Perfil_empre
  CONSTRAINT `fk_Perfil_gestor_Perfil_empresa1`
    FOREIGN KEY (`Perfil_empresa_id_empresa`)
    REFERENCES `mydb`.`Perfil_empresa` (`id_empresa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Reserva_has_Cancha_sintetica`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Reserva_has_Cancha_sinteti
  `Reserva_has_Cancha` VARCHAR(45) NOT NULL,
  `Reserva_idReserva` INT NOT NULL,
  `Cancha_sintetica_idCancha_sintetica` INT NOT NULL,
  INDEX `fk_Reserva_has_Cancha_sintetica_Cancha_sintetica1_id
  INDEX `fk_Reserva_has_Cancha_sintetica_Reserva1_idx` (`Rese
  PRIMARY KEY (`Reserva_has_Cancha`),

```

```

CONSTRAINT `fk_Reserva_has_Cancha_sintetica_Reserva1`
  FOREIGN KEY (`Reserva_idReserva`)
  REFERENCES `mydb`.`Reserva` (`idReserva`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Reserva_has_Cancha_sintetica_Cancha_sintetica`
  FOREIGN KEY (`Cancha_sintetica_idCancha_sintetica`)
  REFERENCES `mydb`.`Cancha_sintetica` (`idCancha_sintetica`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

9. Configuración y Ejecución de la Base de Datos

Manual de Configuración de la Base de Datos

Paso 1: Instalación del Sistema de Gestión de Bases de Datos (SGBD)

1. Descarga el instalador del SGBD desde el sitio web oficial.
2. Ejecuta el instalador y sigue las instrucciones en pantalla para completar la instalación.
3. Durante la instalación, asegúrate de seleccionar las opciones adecuadas según tus necesidades, como la ubicación de instalación y las características a incluir.

Paso 2: Configuración del Cliente de Base de Datos

1. Descarga e instala el cliente de base de datos compatible con tu SGBD desde el sitio web oficial del proveedor.
2. Ejecuta el instalador del cliente y sigue las instrucciones para completar la instalación.

3. Una vez instalado, abre el cliente de base de datos y configura la conexión a la base de datos utilizando la dirección del servidor, el puerto, el nombre de usuario y la contraseña proporcionados por el administrador del sistema.

Paso 3: Ejecución del Script de Creación de la Base de Datos

1. Abre el cliente de base de datos y conéctate al servidor utilizando las credenciales adecuadas.
2. Una vez conectado, abre un nuevo editor de consultas o ventana de comandos SQL.
3. Copia y pega el script de creación de la base de datos proporcionado en el archivo adjunto o en el manual.
4. Ejecuta el script seleccionando todo el contenido y presionando el botón de ejecución o utilizando el comando adecuado según el cliente de base de datos.
5. Verifica que la ejecución del script se haya completado correctamente sin errores.
6. Confirma la creación de la base de datos revisando la estructura y los objetos creados, como tablas, índices y procedimientos almacenados.

10. Otras Consideraciones

Al configurar una base de datos MySQL, es importante tener en cuenta algunos aspectos adicionales para garantizar un funcionamiento correcto y seguro. A continuación, se detallan algunas consideraciones importantes:

1. Puertos de Comunicación:

- Por defecto, MySQL utiliza el puerto 3306 para la comunicación cliente-servidor. Asegúrate de que este puerto esté abierto en el firewall de tu sistema y de tu red para permitir el acceso al servidor MySQL desde clientes remotos si es necesario.
- Si decides utilizar un puerto diferente al predeterminado, asegúrate de configurar adecuadamente tanto el servidor MySQL como el cliente para utilizar el nuevo puerto.

2. Archivo de Configuración (my.cnf o my.ini):

- El archivo de configuración principal de MySQL, llamado `my.cnf` en sistemas Unix/Linux y `my.ini` en sistemas Windows, contiene ajustes importantes que afectan el rendimiento y la seguridad del servidor MySQL.
- Algunos parámetros comunes que puedes necesitar configurar en este archivo incluyen la cantidad de memoria asignada para el búfer de consultas, el tamaño máximo de los archivos de registro, y la configuración de la seguridad como la autenticación y el cifrado de conexiones.

3. Dirección IP del Servidor:

- Si estás configurando MySQL para aceptar conexiones desde clientes remotos, asegúrate de configurar el servidor MySQL para escuchar en la dirección IP adecuada.
- Puedes configurar MySQL para que escuche en todas las interfaces de red (`bind-address = 0.0.0.0`), en una dirección IP específica, o en localhost (`bind-address = 127.0.0.1`).

4. Seguridad y Autenticación:

- Configura adecuadamente la autenticación de usuarios y los permisos de acceso para garantizar que solo los usuarios autorizados puedan acceder y manipular la base de datos.
- Utiliza contraseñas seguras para las cuentas de usuario y evita el acceso como usuario root desde clientes remotos cuando sea posible.

5. Respaldo y Recuperación:

- Implementa un plan de respaldo regular para proteger tus datos contra pérdidas accidentales o corrupción.
- Utiliza herramientas como `mysqldump` o soluciones de respaldo automatizadas para realizar copias de seguridad de tus bases de datos MySQL de forma periódica.

DESCARGAR TRATAMIENTO DE DATOS