PROYECTO FINAL

DISEÑO DE BASES DE DATOS

: Arias Rocha Omar, Buenfil Aguilar Jean Carlos, Castillo Sánchez Miguel Ángel, Chí Quintal José de Jesús y Cutz Anguas Carlos Genaro.

PROFESOR:RAÚL VERA | FACULTAD DE MATEMATICAS - UADY

Reporte Proyecto final - Diseño de base de datos

Autores: Arias Rocha Omar, Buenfil Aguilar Jean Carlos, Castillo Sánchez Miguel Ángel, Chí Quintal José de Jesús y Cutz Anguas Carlos Genaro.

Profesor: Raul Antonio Aguilar Vera.

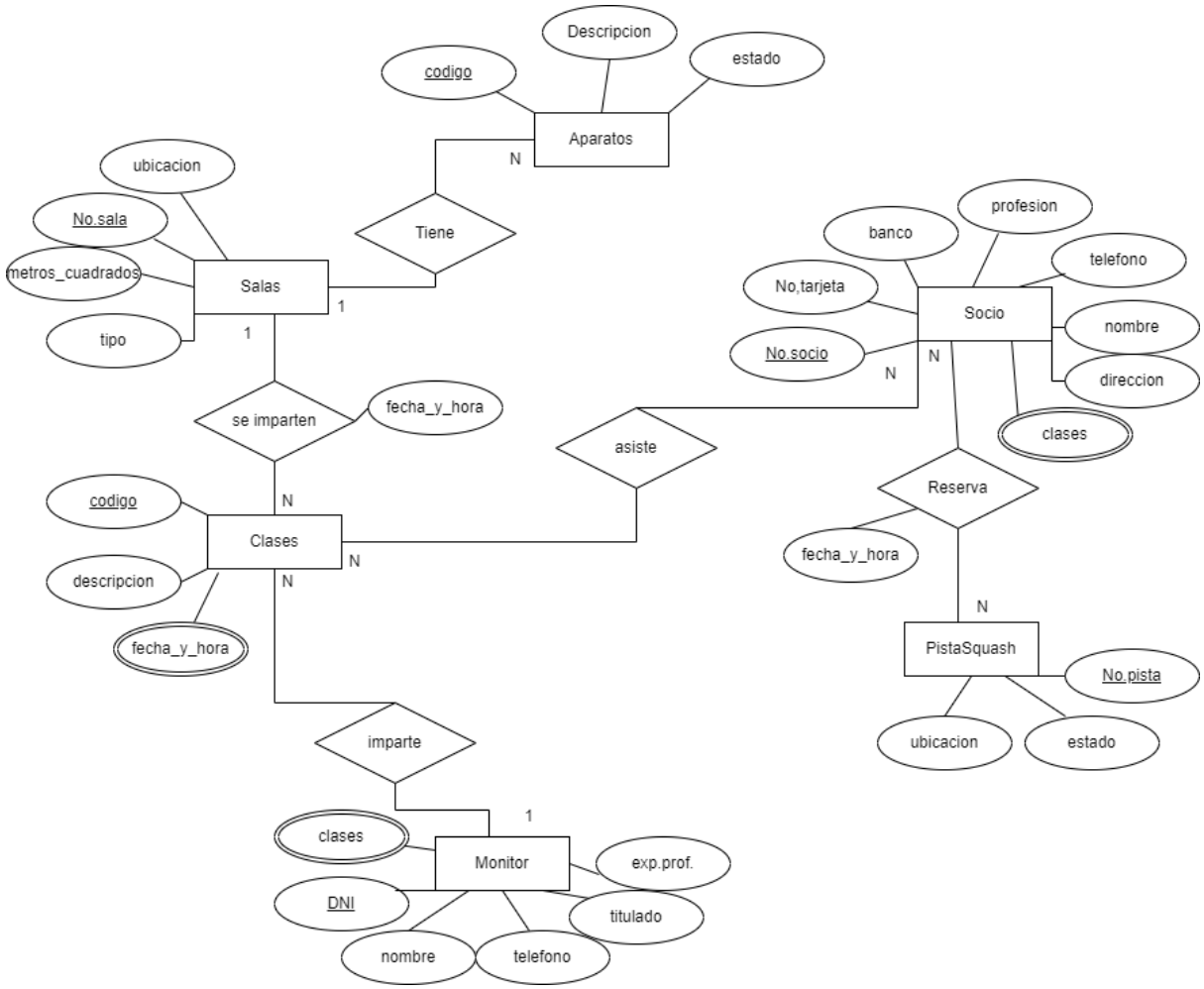
En el gimnasio Zagros Sports se quiere implementar un sistema de información para llevar el control de los socios y recursos, etc. Las especificaciones que nos han dado para el diseño de la base de datos son los siguientes:

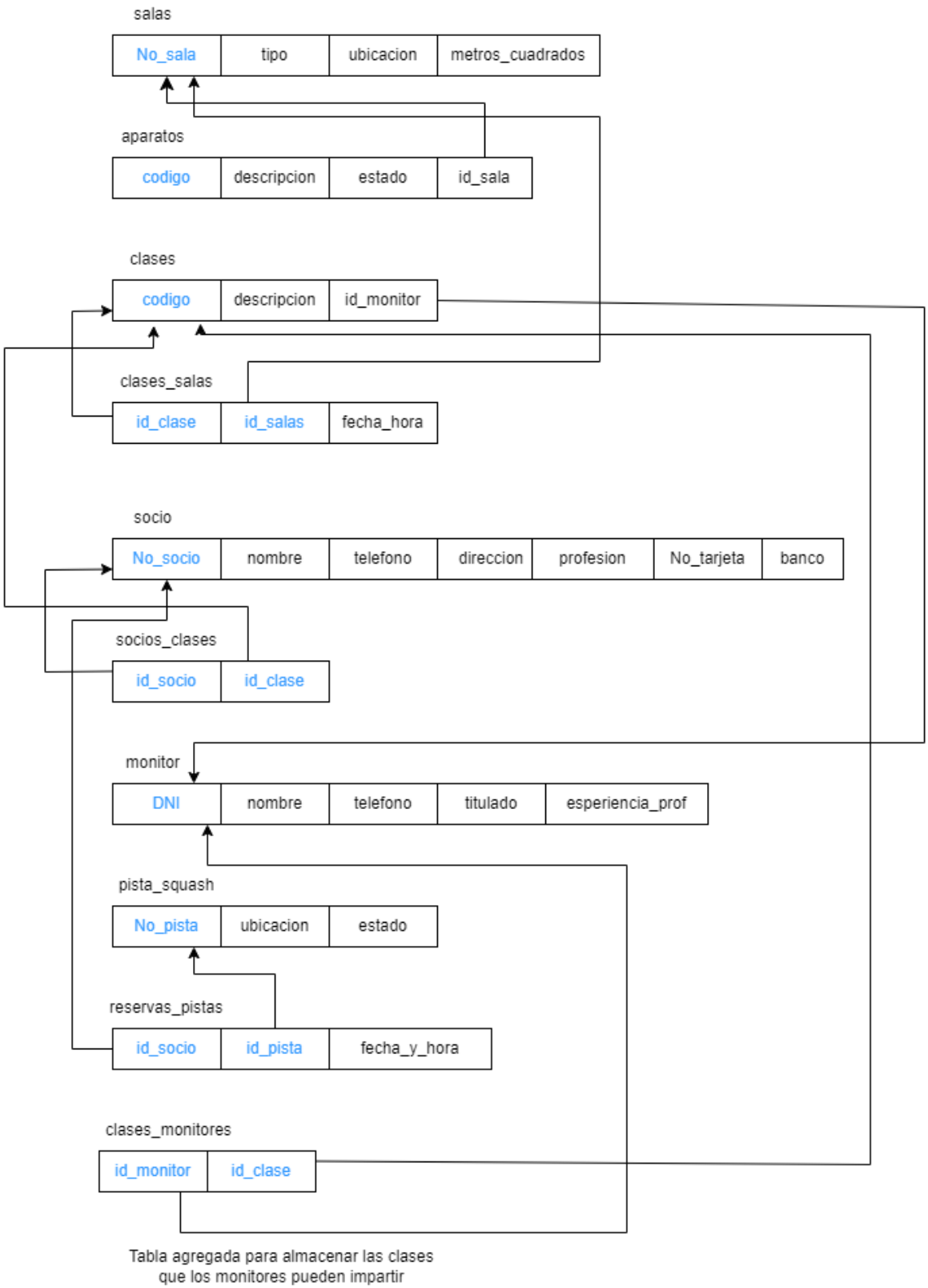
- Existen varias salas, de las cuales se quiere guardar información, como los metros cuadrados que ocupa, ubicación y el tipo de sala(cardio, general, muscular, cada sala se identifica con un número.
- Hay salas que tienen aparatos y salas que no, En las salas se pueden o no impartir clases
- Cada aparato está asignado a una única sala y de cada uno de ellos se quiere tener almacenado su codigo, descripcion y estado de conservación,
- También se quiere mantener información relacionada con las clases que se imparten (descripción, día y hora en la que se imparte) cada clase se identifica por un código de clase. cada clase tiene asignada una sala en la que se imparte y un monitor
- De cada monitor se quiere conocer el DNI , nombre, teléfono, si tiene o no titulación, y experiencia profesional, así como las clases que que pueden impartir(preparación como monitor de aerobic, step, streching, etc.
- De cada socio se quiere conocer el número de socio, nombre, dirección, teléfono, profesión y datos bancarios, así como las clases a las que asiste.
- El gimnasio dispone también de pistas de squash, de las que se quiere conocer el número de pista, ubicación y estado. Las pistas de squash pueden ser utilizadas por socios y existe un servicio de reserva de pista(en una fecha y una hora).

I. Proceso para llevar a cabo el diagrama modelo entidad-relación:

En el gimnasio Zagros Sports se quiere implementar un sistema de información para llevar el control de los socios y recursos, etc. Las especificaciones que nos han dado para el diseño de la base de datos son los siguientes:

- Existen varias salas, de las cuales se quiere guardar información, como los metros cuadrados que ocupa, ubicación y el tipo de sala(cardio, general, muscular, cada sala se identifica con un número.
- Hay salas que tienen aparatos y salas que no, En las salas se pueden o no impartir clases.
- cada aparato está asignado a una única sala y de cada uno de ellos se quiere tener almacenado su codigo, descripcion y estado de conservación,
- También se quiere mantener información relacionada con las clases que se imparten (descripción, día y hora en la que se imparte) cada clase se identifica por un código de clase. cada clase tiene asignada una sala en la que se imparte y un monitor
- De cada monitor se quiere conocer el DNI , nombre, teléfono, si tiene o no titulación, y experiencia profesional, así como las clases que que pueden impartir(preparación como monitor de aerobic, step, streching, etc.
- De cada socio se quiere conocer el número de socio, nombre, dirección, teléfono, profesión y datos bancarios, así como las clases a las que asiste.
- El gimnasio dispone también de pistas de squash, de las que se quiere conocer el número de pista, ubicación y estado. Las pistas de squash pueden ser utilizadas por socios y existe un servicio de reserva de pista(en una fecha y una hora).


Descripcion
codigo
estado
Aparatos
N
Tiene
ubicacion
No.sala
metros_cuadrados
Salas
1
tipo
1
se imparten
fecha_y_hora
profesion
banco
telefono
No,tarjeta
Socio
nombre
No.socio
N
N
direccion
clases
asiste
Reserva
codigo
N
Clases
N
descripcion
N
fecha_y_hora
fecha_y_hora
N
PistaSquash
No.pista
ubicacion
estado
imparte
1
clases
Monitor
exp.prof.
DNI
titulado
nombre
telefono



Se consideró una relación extra entre salas y clases para almacenar las fechas como atributo multivalorado y abriendo la posibilidad a tener una clase en múltiples salas.

Reporte Proyecto final - Diseño de base de datos Punto II

proceso de normalización.

En el gimnasio Zagros Sports se quiere implementar un sistema de información para llevar el control de los socios y recursos, etc. Las especificaciones que nos han dado para el diseño de la base de datos son los siguientes:

- Existen varias salas, de las cuales se quiere guardar información, como los metros cuadrados que ocupa, ubicación y el tipo de sala(cardio, general, muscular, cada sala se identifica con un número.
- Hay salas que tienen aparatos y salas que no, En las salas se pueden o no impartir clases.
- cada aparato está asignado a una única sala y de cada uno de ellos se quiere tener almacenado su codigo, descripcion y estado de conservación,
- También se quiere mantener información relacionada con las clases que se imparten (descripción, día y hora en la que se imparte) cada clase se identifica por un código de clase. cada clase tiene asignada una sala en la que se imparte y un monitor
- De cada monitor se quiere conocer el DNI , nombre, teléfono, si tiene o no titulación, y experiencia profesional, así como las clases que que pueden impartir(preparación como monitor de aerobic, step, streching, etc.
- De cada socio se quiere conocer el número de socio, nombre, dirección, teléfono, profesión y datos bancarios, así como las clases a las que asiste.
- El gimnasio dispone también de pistas de squash, de las que se quiere conocer el número de pista, ubicación y estado. Las pistas de squash pueden ser utilizadas por socios y existe un servicio de reserva de pista(en una fecha y una hora).

No_sala	metros_c2_sala	ubicacion_sala	tipo_sala	**Cod_aparato**	descripcion_aparto

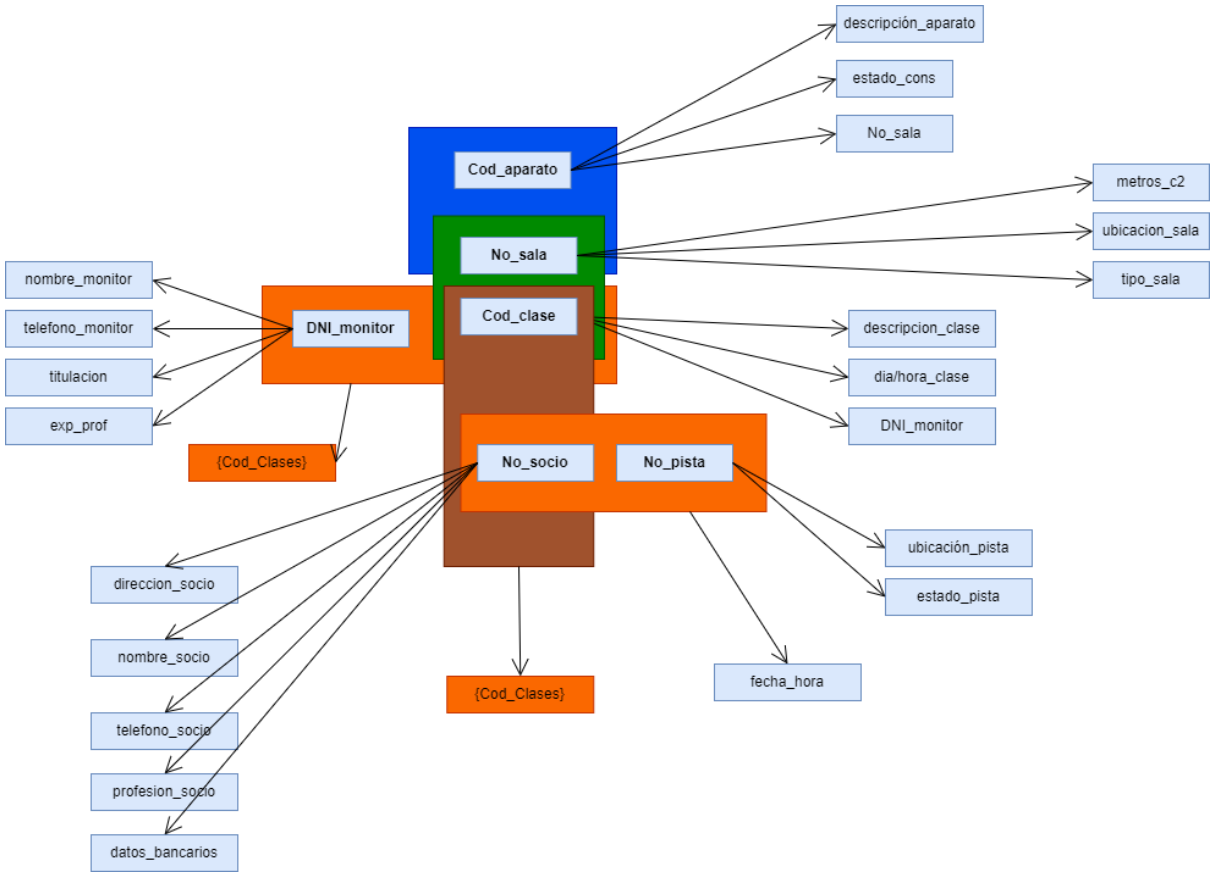
estado_cons	**No_sala**	**DNI_monitor**	nombre_monitor	teléfono_monitor	titulacion

exp_prof	**{Cod_clase}**	**Cod_clase**	descripcion_clase	dia/hora_clase	**DNI_monitor**

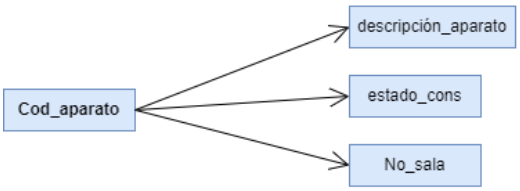
No_socio	nombre_socio	direccion_socio	profesion_socio	**{No_clases}**	datos_bancarios	**No_pista**

ubicación_pista	estado_pista	**No_socio**	**No_pista**	**fecha y hora**

Diagrama de dependencias:



proceso de normalización:



Aparatos

Cod_aparato	descripcion_aparato	estado_cons	No_sala

Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

Una relación R está en Tercera Forma Normal (3FN) si y sólo sí esta en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

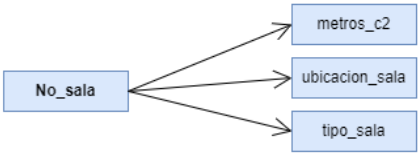
La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

La relación se encuentra en 4FN ya que se encuentra en 3FN y no tiene dependencias de valores múltiples.



Salas

No_sala	tipo_sala	ubicacion_sale	tipo_sala

Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

Una relación R está en Tercera Forma Normal (3FN) si y sólo sí esta en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

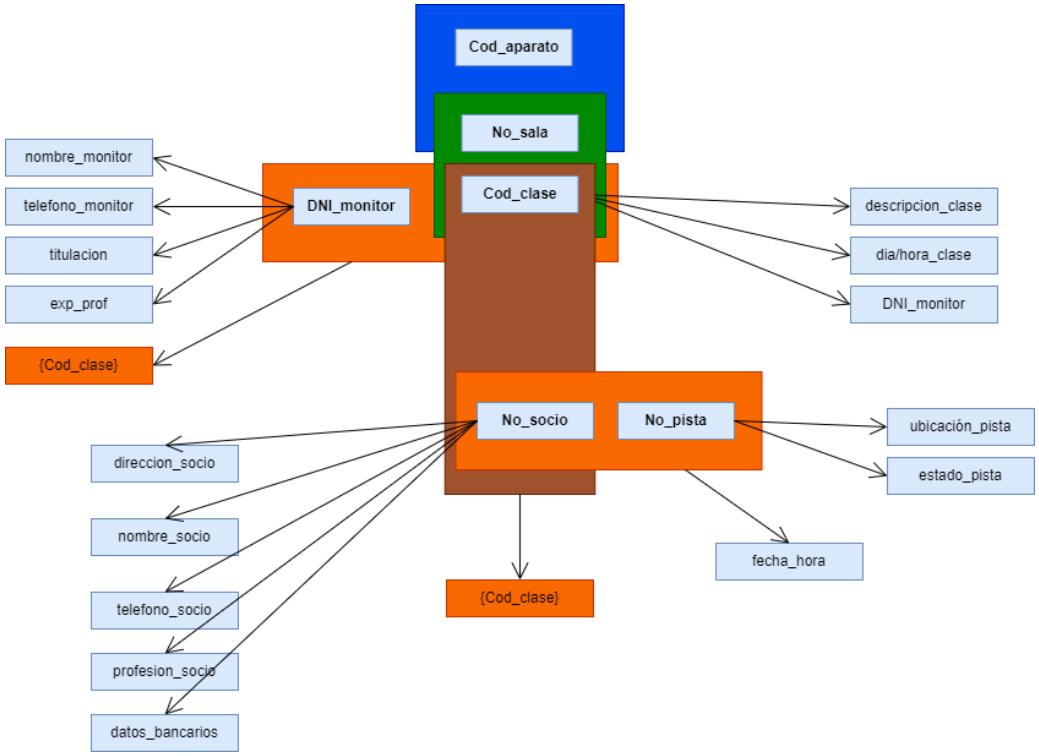
La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

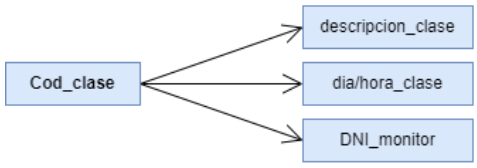
Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

La relación se encuentra en 4FN ya que se encuentra en 3FN y no tiene dependencias de valores múltiples.



Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera



Una relación R está en Tercera Forma Normal (3FN) si y sólo sí esta en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

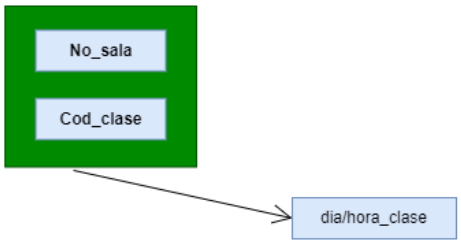
La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

clases

codigo	descripcion	id_monitor

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

La relación se encuentra en 4FN ya que se encuentra en 3FN y no tiene dependencias de valores múltiples.



Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

la relación no se encuentra en 3 forma normal por lo que habría que hacer una extracción de entidad para eliminar los atributos multi valorados.

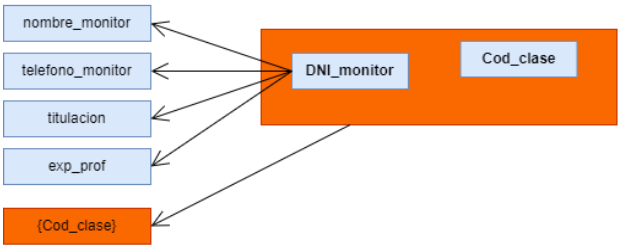
clases_salas

cod_clase	cod_sala	fecha_y_hora

Se decidió agregar esta tabla para que el atributo multivalorado de fecha y hora se almacene en otra tabla agregando la posibilidad de tener clases en salas distintas.

Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera



La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

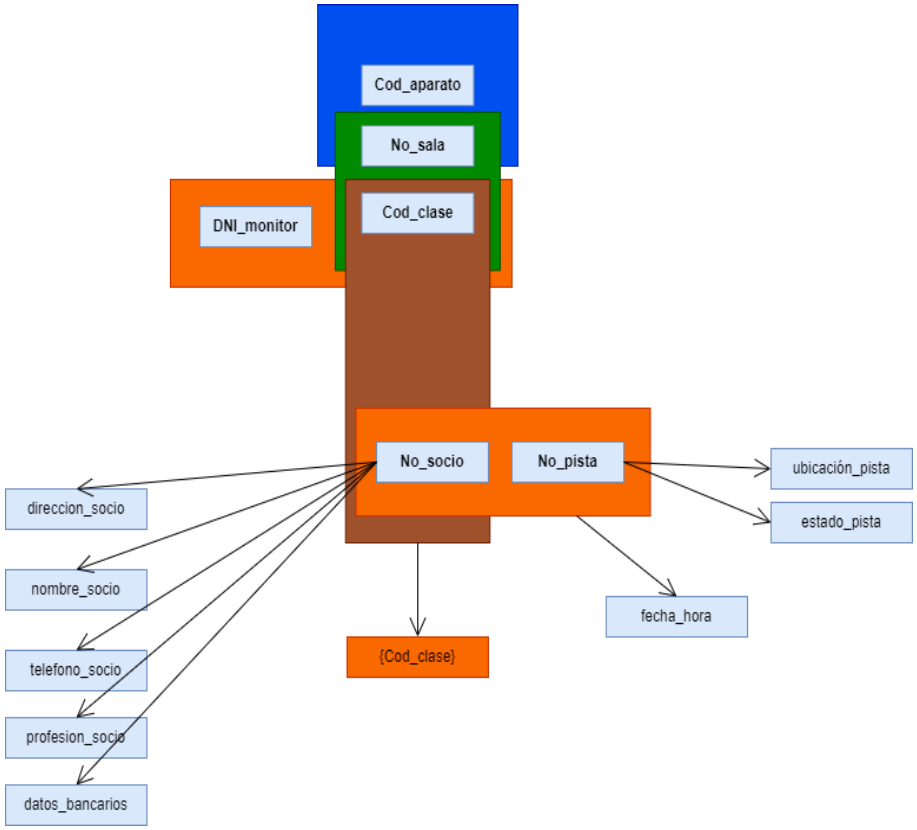
Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

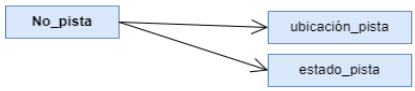
la relación no se encuentra en 3 forma normal por lo que habría que hacer una extracción de entidad para eliminar los atributos multi valorados.

monitor

DNI	nombre	telefono	titulado	esperiencia_prof

id_monitor	id_clase





pista_squash

No_pista	ubicacion	estado

Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

Una relación R está en Tercera Forma Normal (3FN) si y sólo sí esta en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

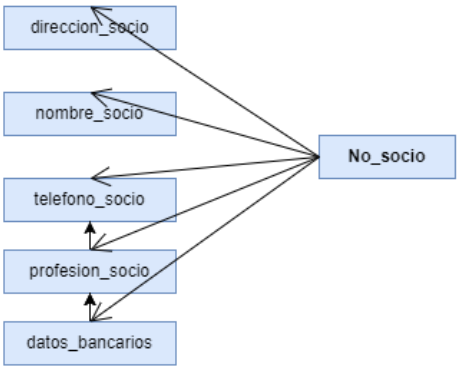
La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

La relación se encuentra en 4FN ya que se encuentra en 3FN y no tiene dependencias de valores múltiples.



Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

Una relación R está en Tercera Forma Normal (3FN) si y sólo sí esta en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

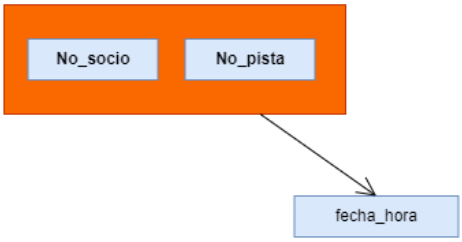
La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

La relación se encuentra en 4FN ya que se encuentra en 3FN y no tiene dependencias de valores múltiples.

socio

No_socio	nombre	telefono	direccion	profesion	No_tarjeta	banco



Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

La relación se encuentra en Segunda Forma Normal ya que todos los atributos no llave dependen por completo de la primera

La relación se encuentra en Tercera Forma Normal ya que se encuentra en 2FN y todos los atributos no clave dependen de manera no transitiva de la primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

La relación se encuentra en BCFN ya que todo determinante de ella puede ser una llave candidata.

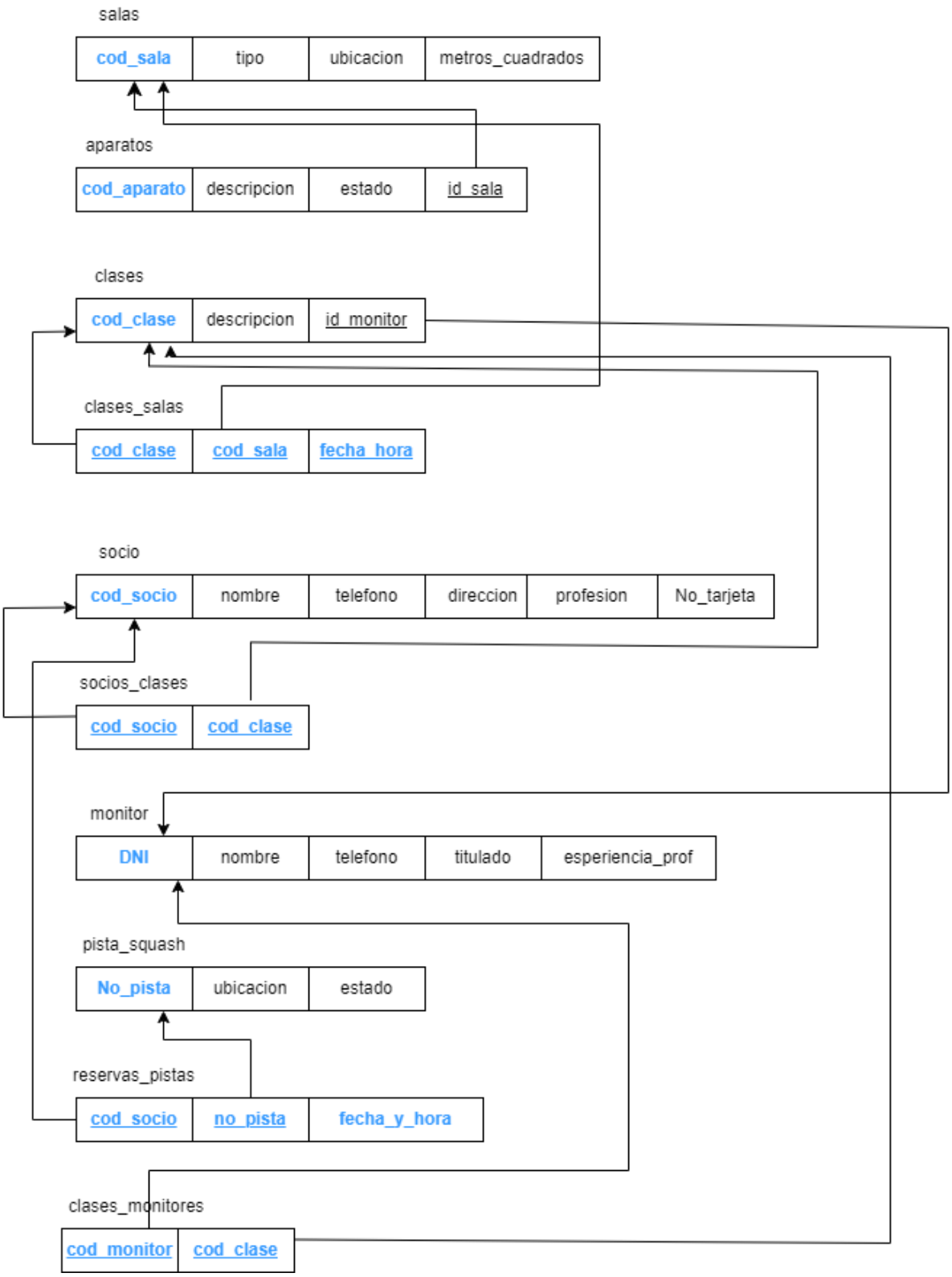
Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

la relación no se encuentra en 3 forma normal por lo que habría que hacer una extracción de entidad para eliminar los atributos multi valorados.

reservas_pistas

id_socio	id_pista	fecha_y_hora

Quedándonos el diagrama de la siguiente forma:



Reporte reflexivo – punto III
Proyecto: diseño de base de datos
Profesor: Raúl Antonio Aguilar Vera.

Comparativa entre los procesos de diseño de una base de datos relacional.

En este reporte se abordaran los dos tipos de procesos con los que se puede construir un diseño de una base de datos, hablando de forma específica se tratara como nosotros percibimos estos dos procesos y hablaremos acerca de nuestras consideraciones al momento de realizar un modelo de base de datos relacional.

Proceso1 – diagrama entidad relación

Un diagrama entidad-relación (ER) es una representación gráfica de los componentes fundamentales de una base de datos. Este diagrama ayuda a visualizar y comprender las relaciones entre las entidades (tablas) y sus atributos (columnas) en un sistema de información. Un diagrama ER contiene entidades, atributos y relaciones. Las entidades representan las tablas, los atributos representan las columnas y las relaciones representan las conexiones entre las tablas. Los diagramas ER son útiles para visualizar la estructura lógica de una base de datos, posteriormente se realiza el diseño lógico de está siguiendo ciertas reglas para la transformación, si tenemos una relación de uno a muchos se agrega una llave foránea al valor que representa un único valor y en el caso de tener atributos multivalorados (elipses dobles), se desglosa como una entidad que represente la relación entre dos entidades y finalmente se agregan las flechas que representan las relaciones apuntando desde las llaves foráneas hasta la llave primaria a la que pertenecen.

Proceso 2 – proceso de normalización y diagrama de dependencias.

Un diagrama de dependencias en base de datos (DDD) es una representación gráfica de las tablas, columnas y relaciones entre las tablas en una base de datos. Este diagrama muestra las relaciones entre las entidades (tablas) y sus atributos (columnas) en un sistema de información.

En un DDD, las entidades se representan como rectángulos que se encuentran dentro de rectángulos más grandes representando las relaciones y los atributos como rectángulos que se encuentran fuera de estos, en cambio las relaciones son representadas como líneas que conectan las entidades y los atributos. Posteriormente se realiza el proceso de normalización el cual consiste en seguir las siguientes normativas hasta tener un diseño funcional de la base de datos:

Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no llave dependen por completo de la llave primaria.

Una relación R está en Tercera Forma Normal (3FN) si y sólo sí está en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

Una relación R está en BCFN cuando todo determinante de ella sea una llave candidata.

Una relación R está en Cuarta Forma Normal (4FN) si y sólo sí, está en 3FN y no tiene dependencias de valores múltiples.

El proceso más comprensible

Para nosotros ambas técnicas están bien definida debido a sus reglas y pasos necesarios que se siguen hasta tener el modelo lógico, sin embargo debido a la practicidad, y el uso del tiempo y ya que nosotros ya teníamos un tiempo trabajando con la primera forma de realizar diseños lógicos se nos hizo más adecuado este proceso por los siguientes motivos:

Experiencia en el uso de este proceso: ya teníamos unos cuantos proyectos diseñados con este formato por lo que todos en el equipo entendían como es que se llevaba a cabo

Eficacia: sin duda alguna este proceso es el más rápido ya que únicamente basándote del diagrama puedes obtener el diseño lógico de este haciendo más rápido el proceso de diseño y aumentando la productividad del equipo

Comprensibilidad: debido a que este proceso tiene un alto grado de entendimiento gráfico, podrías entender fácilmente lo que estos procesos significan y como se representan mediante las formas que se le atribuyen.

El proceso más formal:

A pesar de que a nuestro parecer el primer proceso es el más simple, consideramos que no es lo suficientemente formal, por lo que para nosotros el proceso con más formalidad es el de normalización por los siguientes motivos:

- Análisis detallado de cada componente: al aplicarle las reglas de transformación a cada elemento que compone el modelo podemos fácilmente identificar errores y corregirlos en el momento, cosa la cual en el otro proceso era muy difícil, al derivar cada una de las tablas el proceso se vuelve más analítico y correcto.
- Pasos más específicos: debido a que los pasos ya están bien definidos se nos hizo más sencillo de manera lógica, ya que únicamente debíamos seguir los pasos para llegar al resultado correcto, a pesar de que nos llevase más tiempo.
- Estructura gráfica: la estructura grafica se comprende fácilmente y mediante el proceso podemos identificar hábilmente los elementos internos que lo componen y de donde se derivan mientras se va haciendo la segmentación del diagrama de dependencias.

V. Realice una investigación documental sobre 3 SMBD relacionales, y elabore un reporte indicando: datos de autoría, características y tabla comparativa. Seleccione (con argumentos) uno de dichos SMBD.

Introducción:

Los sistemas de gestión de bases de datos (SMBD) son herramientas esenciales para la administración eficiente de bases de datos. Existen diversos tipos de SMBD, pero en este reporte se analizarán tres sistemas de gestión de bases de datos relacionales: Oracle, MySQL Workbench y phpMyAdmin.

A continuación, se presentan los tres SMBDR seleccionados:

a. Oracle Database (Oracle)

1 - Desarrollado por Oracle Corporation.

2 - Ofrece soporte para bases de datos de diferentes tipos y tamaños.

3 - Se basa en un modelo de dos capas (llaves primarias y llaves foráneas).

Es un sistema de gestión de bases de datos relacional de alto rendimiento y escalabilidad. Ofrece una amplia gama de características avanzadas, como particionamiento, replicación, indexación avanzada y soporte para procedimientos almacenados. Oracle Database es conocido por su capacidad para manejar grandes cantidades de datos y proporcionar un rendimiento eficiente en entornos empresariales. Además, ofrece herramientas de administración y desarrollo, como Oracle SQL Developer, que facilitan la creación y gestión de bases de datos Oracle.

b. MySQL Workbench

1 - Desarrollado por Oracle Corporation.

2 - Orientado a sistemas de información empresarial (ERP).

3 - Ofrece un modelo de tres capas (transacciones, almacenamiento y gestión de la memoria).

Es una herramienta de diseño y administración de bases de datos que se utiliza con el sistema de gestión de bases de datos relacional MySQL. Proporciona una interfaz gráfica intuitiva que facilita la creación, diseño y mantenimiento de bases de datos. MySQL Workbench incluye características como modelado visual de datos, consultas SQL, administración de usuarios, respaldo y restauración de bases de datos, entre otras funcionalidades, lo que lo convierte en una herramienta integral para desarrolladores y administradores de bases de datos MySQL.

c. phpMyAdmin

1 - Desarrollado por PHPMyAdmin.

2 - Utiliza el lenguaje de consulta estructurado (SQL).

3 - Ofrece un modelo de cinco capas (acceso a datos, procesamiento, gestión de transacciones, administración de bases de datos y seguridad).

Es una herramienta de administración de bases de datos MySQL basada en web. Permite a los usuarios gestionar sus bases de datos MySQL a través de una interfaz gráfica desde cualquier navegador web. Con phpMyAdmin, los usuarios pueden realizar tareas como crear, modificar y eliminar bases de datos, tablas y campos, así como ejecutar consultas SQL. Es especialmente útil para desarrolladores web y administradores que desean gestionar bases de datos MySQL de manera eficiente sin depender de una interfaz de línea de comandos. Además, phpMyAdmin es de código abierto y gratuito, lo que lo convierte en una opción popular en entornos de desarrollo web.

La siguiente tabla comparativa presenta una visión general de las características de los tres SMBDR:

	phpMyAdmin	**MySQL Workbench**	**Oracle SQL Developer**
Plataforma	PHP basado en web, escrito en C, Apache License	Java basado en Windows, Linux, Mac, desarrollado por Oracle Corporation	Java basado en Windows, Linux, Mac, desarrollado por Oracle Corporation
Licencia	GNU General Public License versión 2 (GPL)	Dual license consisting of GNU General Public License, version 2 and Oracle Technology Network License, version 1	Dual license consisting of GNU General Public License, version 2 and Oracle Technology Network License, version 1
Navegador soportado	Todos los navegadores	Microsoft Edge, Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari	Microsoft Edge, Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari
Desarrollador	Oracle Corporation	Oracle Corporation	Oracle Corporation

Interfaz de usuario	Gráfica, similar a Windows, permite administrar servidores y bases de datos de manera fácil y rápida	Similar a phpMyAdmin, ofrece una amplia variedad de herramientas para trabajar con bases de datos MySQL	Gráfica, ofrece herramientas avanzadas para administrar y trabajar con bases de datos Oracle
Funciones soportadas	Gestión de servidores y bases de datos, exportación e importación de datos, respaldo y restauración de bases de datos, gestión de usuarios y privilegios, monitoreo de recursos del servidor	Similar a phpMyAdmin, incluye funciones de administración de bases de datos, respaldo y restauración, importación y exportación de datos, monitoreo de recursos del servidor, optimización de consultas SQL, y análisis de desempeño	Ofrece un amplio conjunto de herramientas para desarrollar, administrar y optimizar bases de datos Oracle, como diseñador de bases de datos, asistente para la optimización de consultas, análisis de desempeño, administración de usuarios, entre otros
Programación de funciones	Básico	Básico	Básico
Costo	Gratuito	Gratuito	Gratuito para fines educativos, no gratuitos para fines comerciales

¿Por qué elegimos PHPMyAdmin?

Consideramos que para nuestras necesidades phpMyAdmin es una excelente opción por varias razones:

1. Facilidad de uso: phpMyAdmin tiene una interfaz gráfica fácil de usar, lo que facilita su aprendizaje y uso incluso por parte de personas sin experiencia previa en bases de datos.
2. Herramientas integradas: phpMyAdmin ofrece herramientas integradas para la gestión de bases de datos, lo que evita la necesidad de instalar y configurar múltiples herramientas separadas.
3. Acceso remoto: phpMyAdmin permite el acceso remoto a las bases de datos, lo que es útil para aquellos que trabajan en equipo o para aquellos que necesitan acceder a sus bases de datos desde lugares distintos a su casa.
4. Integración con Apache y PHP: phpMyAdmin es una herramienta que funciona directamente con el servidor web Apache y el lenguaje de programación PHP, lo que facilita su implementación y configuración en entornos de desarrollo basados en PHP y MySQL en el caso de que se le quiera dar un seguimiento a este proyecto.

Conclusión.

Cada uno de estos sistemas de gestión de bases de datos tiene sus propias ventajas y desventajas. Oracle, por ejemplo, es ampliamente utilizado en el ámbito empresarial, pero su curva de aprendizaje puede ser más pronunciada. MySQL Workbench, por otro lado, es gratuito y ofrece un buen equilibrio entre características y precio. Finalmente, phpMyAdmin es una herramienta open source que se adapta perfectamente a los proyectos de desarrollo web y ofrece un buen rendimiento.

Referencias:

PhpMyAdmin. (s.f.). Welcome to PhpMyAdmin. Retrieved from https://www.phpmyadmin.net/

Oracle SQL Developer. (s.f.). Welcome to Oracle SQL Developer. Retrieved from https://www.oracle.com/tools/downloads/sqldev-downloads.html

MySQL Workbench. (s.f.). MySQL Workbench - The integrated tools for MySQL Database Management. Retrieved from https://www.mysql.com/products/workbench/

Fernández, P. (2017). Diseño de bases de datos en tiempo real y big data: Conceptos y aplicaciones en la gestión de la información y la toma de decisiones en la era digital. Editorial Anaya Multimedia.

Reporte proyecto final- Punto V,VI y VII

Autores: Arias Rocha Omar, Buenfil Aguilar Jean Carlos, Castillo Sanchez Miguel Angel, Chí Quintal José de Jesus y Cutz Anguas Carlos Genaro.

V. Utilizando el Sistema Manejador de Bases de Datos seleccionado, Implemente el Diseño Físico de la Base de Datos del problema descrito (incluya las sentencias en el reporte).

CREACIÓN DE TABLAS:

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0074 segundos.)

```
CREATE TABLE salas ( no_salas INT PRIMARY KEY AUTO_INCREMENT, tipo VARCHAR(255), ubicacion VARCHAR(255), metros_cuadrados
DECIMAL(10, 2) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0233 segundos.)

```
CREATE TABLE aparatos ( codigo INT PRIMARY KEY AUTO_INCREMENT, descripcion VARCHAR(255), estado VARCHAR(50), id_sala INT, FOREIGN
KEY (id_sala) REFERENCES salas(no_salas) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0078 segundos.)

```
CREATE TABLE socio ( No_socio INT PRIMARY KEY AUTO_INCREMENT, nombre VARCHAR(255), telefono VARCHAR(20), direccion VARCHAR(255),
profesion VARCHAR(100), banco VARCHAR(50), numero_tarjeta VARCHAR(16) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0077 segundos.)

```
CREATE TABLE monitor ( DNI VARCHAR(20) PRIMARY KEY, nombre VARCHAR(255), telefono VARCHAR(20), titulado BOOLEAN,
experiencia_profesional VARCHAR(255), titulacion VARCHAR(255) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0211 segundos.)

```
CREATE TABLE clases ( codigo INT PRIMARY KEY AUTO_INCREMENT, descripcion VARCHAR(255), id_monitor VARCHAR(20), FOREIGN KEY (id_monitor) REFERENCES monitor(DNI) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0227 segundos.)

```
CREATE TABLE clases_salas ( id_clase INT, id_sala INT, fecha_hora DATETIME, PRIMARY KEY (id_clase, id_sala, fecha_hora), FOREIGN KEY (id_clase) REFERENCES clases(codigo), FOREIGN KEY (id_sala) REFERENCES salas(no_salas) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0220 segundos.)

```
CREATE TABLE socio_clases ( id_socio INT, id_clase INT, PRIMARY KEY (id_socio, id_clase), FOREIGN KEY (id_socio) REFERENCES socio(No_socio), FOREIGN KEY (id_clase) REFERENCES clases(codigo) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0071 segundos.)

```
CREATE TABLE pistas_squash ( No_pista INT PRIMARY KEY AUTO_INCREMENT, ubicacion VARCHAR(255), estado VARCHAR(50) );
```

[Editar en línea] [Editar] [Crear código PHP]

Mostrar ventana de consultas SQL

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0207 segundos.)

```
CREATE TABLE reservas_pistas ( id_socio INT, id_pista INT, fecha_hora DATETIME, PRIMARY KEY (id_socio, id_pista, fecha_hora), FOREIGN KEY (id_socio) REFERENCES socio(No_socio), FOREIGN KEY (id_pista) REFERENCES pistas_squash(No_pista) );
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0194 segundos.)

```sql
CREATE TABLE clases_monitores ( id_clase INT, id_monitor VARCHAR(20), PRIMARY KEY (id_clase, id_monitor), FOREIGN KEY (id_clase)
REFERENCES clases(codigo), FOREIGN KEY (id_monitor) REFERENCES monitor(DNI) );
```

[Editar en línea] [Editar] [Crear código PHP]

VI. Utilizando el Sistema Manejador de Bases de Datos seleccionado, capture un conjunto de tuplas relacionadas con la información que dicha BD mantiene (incluya un par de ejemplos de las sentencias utilizadas para la captura en el reporte).

Tuplas insertadas:

Ejecutar la(s) consulta(s) SQL en la base de datos zagros_sport:

```sql
INSERT INTO aparatos (descripcion, estado, id_sala) VALUES
    ('Cinta de Correr', 'Disponible', 2),
    ('Maquina de Pesas', 'En Mantenimiento', 2),
    ('Bicicleta Estacionaria', 'Disponible', 3);


INSERT INTO socio (nombre, telefono, direccion, profesion, banco, numero_tarjeta) VALUES
    ('Juan Pérez', '555-1234', 'Calle Principal 123', 'Ingeniero', 'Banco A', '1234-5678-9012-3456'),
    ('María Rodríguez', '555-5678', 'Avenida Secundaria 456', 'Médico', 'Banco B', '2345-6789-0123-4567'),
    ('Carlos González', '555-9876', 'Calle Secundaria 789', 'Abogado', 'Banco C', '3456-7890-1234-5678');
```

✔ 2 filas insertadas.
La Id de la fila insertada es: 5

```
INSERT INTO `clases` (`codigo`, `descripcion`, `id_monitor`) VALUES (NULL, 'Clase de pilates utilizando objetos de casa', '45678901C'), (NULL, 'clase de halterofilia avanzada', '12345678A');
```

[Editar en línea] [Editar] [Crear código PHP]

✔ 1 fila insertada.

```
INSERT INTO `monitor` (`DNI`, `nombre`, `telefono`, `experiencia_profesional`, `titulacion`) VALUES ('6734532389434589', 'Raul Jose Castillo Rocha', '999 8775 344', 'instructor profesional de 5 años', 'Escuela superior de artes marciales');
```

[Editar en línea] [Editar] [Crear código PHP]

✔ 2 filas insertadas.
La Id de la fila insertada es: 5

```
INSERT INTO `salas` (`no_salas`, `tipo`, `ubicacion`, `metros_cuadrados`) VALUES (NULL, 'Sala de cardio', 'primer piso segunda puerta', '80'), (NULL, 'Sala de mancuernas y poleas', 'primer piso, primera puerta ', '35.80');
```

[Editar en línea] [Editar] [Crear código PHP]

✔ 2 filas insertadas.

```
INSERT INTO `clases_salas` (`id_clase`, `id_sala`, `fecha_hora`) VALUES ('5', '5', '2023-12-08 12:14:37'), ('1', '3', '2023-12-06 18:14:37');
```

[Editar en línea] [Editar] [Crear código PHP]

VII. Utilizando el Sistema Manejador de Bases de Datos seleccionado, implemente el siguiente conjunto de consultas (incluya en el reporte, las sentencias utilizadas para las 7 solicitudes, así como capturas de pantalla de lo generado una vez ejecutados dichas consultas)

Querys:

• Obtener la relación de salas de que consta el GYM, con el detalle: Identificador, Ubicación, Tipo, metros cuadrados.

Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0003 segundos.)

```
SELECT * FROM salas;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	no_salas	tipo	ubicacion	metros_cuadrados
Editar Copiar Borrar	1	Sala de Reuniones	Planta Baja	30.50
Editar Copiar Borrar	2	Gimnasio	Piso 1	200.75
Editar Copiar Borrar	3	Sala de Yoga	Piso 2	50.00
Editar Copiar Borrar	4	Sala de cardio	primer piso segunda puerta	80.00
Editar Copiar Borrar	5	Sala de mancuernas y poleas	primer piso, primera puerta	35.80

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

• Obtener la relación de aparatos adquiridos por el GYM con el detalle: Código, descripción, edo. Conservación.

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0003 segundos.)

```
SELECT * FROM aparatos;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	codigo	descripcion	estado	id_sala
Editar Copiar Borrar	1	Cinta de Correr	Nuevo	2
Editar Copiar Borrar	2	Máquina de Pesas	En Mantenimiento	2
Editar Copiar Borrar	3	Bicicleta Estacionaria	Disponible	3

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

• Obtener la relación de instructores que trabajan para el GYM, con el detalle: DNI, Nombre, Preparación.

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0003 segundos.)

```
SELECT DNI, nombre,titulacion,experiencia_profesional FROM monitor;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	DNI	nombre	titulacion	experiencia_profesional
Editar Copiar Borrar	12345678A	Laura Martínez	Licenciatura en Educación Física	Entrenadora Certificada
Editar Copiar Borrar	45678901C	Pedro Rodríguez	Licenciatura en Ciencias del Deporte	Entrenador de Pesas
Editar Copiar Borrar	6734532389434589	Raul Jose Castillo Rocha	Escuela superior de artes marciales	instructor profesional de 5 años
Editar Copiar Borrar	87654321B	Alejandro Gómez	Certificación en Yoga	Especialista en Yoga

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

• Obtener la relación de socios registrados al GYM, con el detalle: No. Socio, Nombre, Teléfono.

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0003 segundos.)

```
SELECT No_socio, nombre,telefono FROM socio;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	No_socio	nombre	telefono
Editar Copiar Borrar	1	Juan Pérez	555-1234
Editar Copiar Borrar	2	María Rodríguez	555-5678
Editar Copiar Borrar	3	Carlos González	555-9876

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

• Obtener la relación de clases programadas para la semana, con el detalle: No. Sala, Instructor, Descripción, Horario, Día.

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0004 segundos.)

```
SELECT s.no_salas AS "No. Sala", m.nombre AS Instructor, c.descripcion AS Descripción, DATE_FORMAT(cs.fecha_hora, '%Y-%m-%d
%H:%i') AS "Fecha y Hora", DAYNAME(cs.fecha_hora) AS Día FROM clases_salas cs JOIN clases c ON cs.id_clase = c.codigo JOIN salas
s ON cs.id_sala = s.no_salas JOIN monitor m ON c.id_monitor = m.DNI WHERE WEEK(cs.fecha_hora, 1) = WEEK(CURRENT_DATE(), 1) AND
YEAR(cs.fecha_hora) = YEAR(CURRENT_DATE());
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

No. Sala	Instructor	Descripción	Fecha y Hora	Día
5	Laura Martínez	clase de halterofilia avanzada	2023-12-08 12:14	Friday
3	Alejandro Gómez	Clase de Yoga	2023-12-06 18:14	Wednesday

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

• Obtener la relación de pistas reservadas para el fin de semana, con el detalle: Fecha, Hora, No. pista, No. Socio que realizó la reserva.

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0005 segundos.)

```
SELECT DATE_FORMAT(rp.fecha_hora, '%Y-%m-%d') AS Fecha, DATE_FORMAT(rp.fecha_hora, '%H:%i') AS Hora, ps.No_pista AS "No. Pista",
s.No_socio AS "No. Socio que Reservó" FROM reservas_pistas rp JOIN pistas_squash ps ON rp.id_pista = ps.No_pista JOIN socio s ON
rp.id_socio = s.No_socio WHERE DAYNAME(rp.fecha_hora) IN ('Saturday', 'Sunday') AND WEEK(rp.fecha_hora) = WEEK(CURRENT_DATE());
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

Fecha	Hora	No. Pista	No. Socio que Reservó
2023-12-09	12:00	2	2
2023-12-03	10:00	2	3

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

• Agregue una consulta que considere requerida para el administrador del GYM (justifíquela).

aqui se muestra una consulta que nosotros consideramos relevantes para el gimnasio, en ella se muestran las clases y reservas del dia de hoy y la cantidad de socios totales que existen

Su consulta se ejecutó con éxito.

```
SELECT (SELECT COUNT(*) FROM socio) AS "Total de Socios", (SELECT COUNT(*) FROM clases_salas WHERE fecha_hora >= CURRENT_DATE()) AS
"Clases Programadas Hoy", (SELECT COUNT(*) FROM reservas_pistas WHERE DATE_FORMAT(fecha_hora, '%Y-%m-%d') = CURRENT_DATE()) AS
"Reservas de Pistas Hoy";
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Opciones extra

Total de Socios	Clases Programadas Hoy	Reservas de Pistas Hoy
3	2	0

Operaciones sobre los resultados de la consulta

Imprimir | Copiar al portapapeles | Exportar | Mostrar gráfico | Crear vista