



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Diseño e implementación de una base de datos para el Sistema de Servicios y Control de Salas de Cómputo de la Facultad de Ingeniería, utilizando las nuevas tecnologías mediante software libre**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico Electrónico**

**P R E S E N T A**

Alan Limón Badillo

**DIRECTOR DE TESIS**

M.I. Cruz Sergio Aguilar Díaz



**Ciudad Universitaria, Cd. Mx., 2024**

# ÍNDICE GENERAL

INDICE DE FIGURAS .....	iv
INDICE DE TABLAS .....	vi
INTRODUCCIÓN.....	vii
Capítulo 1 ANTECEDENTES .....	9
1.1 PLANTEAMIENTO DEL PROBLEMA.....	12
1.2 OBJETIVO.....	12
1.3 JUSTIFICACIÓN.....	13
1.4 RESULTADOS .....	14
Capítulo 2 FUNDAMENTOS TEÓRICOS.....	16
2.1 SOFTWARE.....	16
2.1.1 Definición de software.....	16
2.1.2 Clasificación de software .....	16
2.2 SISTEMAS OPERATIVOS.....	21
2.2.1 Definición de sistema operativo.....	21
2.2.2 Clasificación de sistemas operativos .....	21
2.2.3 Distribuciones GNU/Linux .....	23
2.2.4 Comparativa entre distribuciones Linux .....	25
2.3 CONCEPTOS DE BASES DE DATOS .....	27
2.3.1 Definición de sistema de base de datos .....	27
2.3.2 Elementos de un sistema de base de datos.....	28
2.3.3 Cualidades de un sistema de base de datos .....	36
2.3.4 Clasificación de un sistema de base de datos .....	39
2.3.5 Arquitectura de un sistema de base de datos .....	41
2.3.6 Modelos de datos.....	42
2.3.7 Normalización.....	50
2.4 MANEJADORES DE BASES DE DATOS .....	54
2.4.1 Manejadores de bases de datos (SQL y NoSQL) .....	54
2.4.2 Principales manejadores de bases de datos.....	57
2.4.3 Comparativa entre manejadores de bases de datos.....	59
2.5 LENGUAJE DE CONSULTAS SQL.....	62
2.5.1 Definición de SQL.....	62
2.5.2 Sentencias SQL .....	63
2.6 APLICACIONES.....	83

<b>Capítulo 3 IMPLEMENTACIÓN DE LA BASE DE DATOS .....</b>	<b>85</b>
<b>3.1 JUSTIFICACIÓN DE LAS HERRAMIENTAS.....</b>	<b>85</b>
3.1.1 ¿Por qué GNU/Linux? .....	85
3.1.2 ¿Por qué PostgreSQL? .....	86
3.1.3 ¿Por qué CentOS? .....	87
<b>3.2 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA BASE DE DATOS .....</b>	<b>88</b>
<b>3.3 DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS .....</b>	<b>89</b>
3.3.1 Antecedentes para el análisis y diseño .....	89
3.3.2 Diseño .....	94
3.3.3 Instalación y configuración.....	108
3.3.4 Implementación .....	109
3.3.5 Población de la base de datos .....	113
3.3.6 Diseño e implementación de la interfaz de usuario .....	115
<b>3.4 PRUEBAS Y MANTENIMIENTO (AJUSTES).....</b>	<b>117</b>
<b>RESULTADOS.....</b>	<b>124</b>
<b>CONCLUSIONES.....</b>	<b>125</b>
<b>Apéndice A. INSTALACIÓN DEL SOFTWARE DE LA BASE DE DATOS .....</b>	<b>127</b>
<b>Apéndice B. LÓGICA DE DESARROLLO DE LA BASE DE DATOS.....</b>	<b>139</b>
<b>Apéndice C. REPRESENTACIÓN GRÁFICA DE LA INTERFAZ DE USUARIO.....</b>	<b>146</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>151</b>
<b>REFERENCIAS ELECTRÓNICAS .....</b>	<b>151</b>
<b>INTRODUCCIÓN .....</b>	<b>151</b>
<b>CAPÍTULO 1 ANTECEDENTES.....</b>	<b>151</b>
<b>CAPÍTULO 2 FUNDAMENTOS TEÓRICOS.....</b>	<b>152</b>
<b>CAPÍTULO 3 IMPLEMENTACIÓN DE LA BASE DE DATOS .....</b>	<b>158</b>
<b>Apéndice A. INSTALACIÓN DEL SOFTWARE DE LA BASE DE DATOS.....</b>	<b>160</b>
<b>Referencias.....</b>	<b>162</b>

## INDICE DE FIGURAS

<b>Figura 1.1</b> Organigrama de las áreas de la Facultad de Ingeniería.....	11
<b>Figura 1.2</b> Ubicación de las Salas de Cómputo UNICA. ....	12
<b>Figura 2.1</b> Software de sistema. ....	17
<b>Figura 2.2</b> Software de desarrollo. ....	18
<b>Figura 2.3</b> Software de aplicación.....	20
<b>Figura 2.4</b> Software operativo. ....	21
<b>Figura 2.5</b> Sistemas operativos móviles. ....	22
<b>Figura 2.6</b> Distribuciones GNU/Linux. ....	23
<b>Figura 2.7</b> Relación usuario, aplicación de bases de datos, SGBD y base de datos.....	27
<b>Figura 2.8</b> Flujo de información en un sistema de base de datos. ....	28
<b>Figura 2.9</b> Un sistema informático centralizado.....	29
<b>Figura 2.10</b> Organización de un sistema de base de datos. ....	40
<b>Figura 2.11</b> Arquitectura de tres niveles de abstracción de datos. ....	41
<b>Figura 2.12</b> Elementos básicos de un modelo entidad relación.....	42
<b>Figura 2.13</b> Clasificación de las entidades. ....	43
<b>Figura 2.14</b> Clasificación de los atributos. ....	44
<b>Figura 2.15</b> Cardinalidad de las relaciones.....	46
<b>Figura 2.16</b> Ejemplo de diagrama E-R. ....	47
<b>Figura 2.17</b> Ejemplo de base de datos relacional. ....	48
<b>Figura 2.18</b> Ejemplo de una base de datos jerárquica y de red. ....	50
<b>Figura 2.19</b> (a) Relación no normalizada. (b) Relación en primera forma normal. ....	51
<b>Figura 2.20</b> Instancia de la tabla ARTÍCULO.....	52
<b>Figura 2.21</b> Instancia de la tabla ARTÍCULO1.....	53
<b>Figura 2.22</b> Instancia de la tabla FABRICANTE1.....	53
<b>Figura 2.23</b> Tablas en tercera forma normal. ....	54
<b>Figura 2.24</b> Manejadores de bases de datos. ....	57
<b>Figura 3.1</b> Equipos de cómputo contemplados para implementar la base de datos. ....	88
<b>Figura 3.2</b> Sala 1 UNICA. ....	89
<b>Figura 3.3</b> Sala 2 UNICA. ....	90
<b>Figura 3.4</b> Sala 3 UNICA. ....	91
<b>Figura 3.5</b> Sala 4 UNICA. ....	92
<b>Figura 3.6</b> Modelo ER de las salas de cómputo de UNICA. ....	99
<b>Figura 3.7</b> Diagrama de esquema de la base de datos UNICA.....	100
<b>Figura 3.8</b> Comandos iniciales para iniciar sesión en el sistema operativo. ....	109
<b>Figura 3.9</b> Relación padre hijo entre las tablas de la base de datos.....	110
<b>Figura 3.10</b> Consulta de datos de las aulas de sala 3 UNICA. ....	117
<b>Figura 3.11</b> Consulta de datos de los cursos presenciales de UNICA.....	118
<b>Figura 3.12</b> Consulta del software disponible en sala 1. ....	118
<b>Figura 3.13</b> Formulario de inicio de sesión. ....	119

<b>Figura 3.14</b> Consulta realizada en el apartado reporte. ....	119
<b>Figura 3.15</b> Resultados iniciales de la prueba de estrés. ....	121
<b>Figura 3.16</b> Resultados finales de la prueba de estrés. ....	122
<b>Figura A.1</b> Pantalla de inicio de VirtualBox. ....	127
<b>Figura A.2</b> Datos básicos de la máquina virtual. ....	127
<b>Figura A.3</b> Cantidad de memoria RAM reservada. ....	128
<b>Figura A.4</b> Forma de implementación del disco duro. ....	128
<b>Figura A.5</b> Tipo de archivo del disco duro. ....	129
<b>Figura A.6</b> Configuración del comportamiento del disco duro. ....	129
<b>Figura A.7</b> Directorio y capacidad del disco duro. ....	130
<b>Figura A.8</b> Selección del archivo ISO para instalar el sistema operativo. ....	130
<b>Figura A.9</b> Pantalla de inicio para la instalación de CentOS 7. ....	131
<b>Figura A.10</b> Selección del idioma de CentOS 7. ....	131
<b>Figura A.11</b> Menú principal de configuración del sistema. ....	132
<b>Figura A.12</b> Selección de software. ....	132
<b>Figura A.13</b> Asignación de usuario y contraseña. ....	133
<b>Figura A.14</b> Configuración de red. ....	135
<b>Figura A.15</b> Edición de conexión. ....	135
<b>Figura A.16</b> Prueba de conexión PING. ....	136
<b>Figura B.1</b> Definición de la tabla aula. ....	139
<b>Figura B.2</b> Definición de la tabla aviso. ....	139
<b>Figura B.3</b> Definición de la tabla curso. ....	140
<b>Figura B.4</b> Definición de la tabla equipo. ....	140
<b>Figura B.5</b> Definición de la tabla reporte. ....	141
<b>Figura B.6</b> Definición de la tabla sala. ....	141
<b>Figura B.7</b> Definición de la tabla software. ....	142
<b>Figura B.8</b> Definición de la tabla software_curso. ....	142
<b>Figura B.9</b> Definición de la tabla software_equipo. ....	142
<b>Figura B.10</b> Definición de la tabla software_sala. ....	143
<b>Figura B.11</b> Definición de la tabla usuario. ....	143
<b>Figura B.12</b> Definición de la tabla usuario_curso. ....	144
<b>Figura B.13</b> Definición de la tabla usuario_reporte. ....	144
<b>Figura B.14</b> Creación de los roles de la base de datos. ....	144
<b>Figura B.15</b> Asignación de privilegios a los roles de la base de datos. ....	145
<b>Figura B.16</b> Creación de cuentas de usuario de la base de datos. ....	145
<b>Figura B.17</b> Asignación de roles a los usuarios de la base de datos. ....	145
<b>Figura C.1</b> Apartados de la interfaz pública de usuario. ....	147
<b>Figura C.2</b> Apartados de la interfaz privada de usuario. ....	148
<b>Figura C.3</b> Vistas resultantes del apartado equipos. ....	149
<b>Figura C.4</b> Vistas resultantes del apartado reportes. ....	150

## INDICE DE TABLAS

<b>Tabla 2.1</b> Comparativa entre distribuciones Linux.....	25
<b>Tabla 2.2</b> Comparativa entre manejadores de base de datos. ....	59
<b>Tabla 2.3</b> Datos tipo caracter.....	64
<b>Tabla 2.4</b> Datos tipo numérico. ....	65
<b>Tabla 2.5</b> Datos tipo fecha y hora.....	66
<b>Tabla 2.6</b> Restricciones de integridad.....	66
<b>Tabla 2.7</b> Funciones de tipo fecha y hora.....	67
<b>Tabla 2.8</b> Acciones de activación referencial.....	68
<b>Tabla 2.9</b> Privilegios para interactuar con la base de datos.....	80
<b>Tabla 3.1</b> Características de las computadoras contempladas para el proyecto.....	88
<b>Tabla 3.2</b> Recursos Sala 1 UNICA.....	89
<b>Tabla 3.3</b> Recursos Sala 2 UNICA.....	91
<b>Tabla 3.4</b> Recursos Sala 3 UNICA.....	92
<b>Tabla 3.5</b> Recursos Sala 4 UNICA.....	93
<b>Tabla 3.6</b> Subsistemas relevantes del sistema principal.....	94
<b>Tabla 3.7</b> Planteamiento de las entidades de las salas de cómputo.....	95
<b>Tabla 3.8</b> Atributos asociados a las entidades de las salas de cómputo.....	96
<b>Tabla 3.9</b> Diccionario de datos de la tabla aula.....	101
<b>Tabla 3.10</b> Diccionario de datos de la tabla aviso.....	102
<b>Tabla 3.11</b> Diccionario de datos de la tabla curso.....	103
<b>Tabla 3.12</b> Diccionario de datos de la tabla equipo.....	103
<b>Tabla 3.13</b> Diccionario de datos de la tabla reporte.....	104
<b>Tabla 3.14</b> Diccionario de datos de la tabla sala.....	105
<b>Tabla 3.15</b> Diccionario de datos de la tabla software.....	105
<b>Tabla 3.16</b> Diccionario de datos de la tabla software_curso.....	105
<b>Tabla 3.17</b> Diccionario de datos de la tabla software_equipo.....	106
<b>Tabla 3.18</b> Diccionario de datos de la tabla software_sala.....	106
<b>Tabla 3.19</b> Diccionario de datos de la tabla usuario.....	107
<b>Tabla 3.20</b> Diccionario de datos de la tabla usuario_curso.....	107
<b>Tabla 3.21</b> Diccionario de datos de la tabla usuario_reporte.....	107
<b>Tabla 3.22</b> Tablas generadas por la sentencia pgbench.....	120

# INTRODUCCIÓN

La Universidad Nacional Autónoma de México (UNAM) es considerada como una de las universidades de mayor prestigio en el país y en América Latina. A nivel licenciatura esta ofrece 129 carreras en las áreas de las Ciencias Físico-Matemáticas y de las Ingenierías; Ciencias Biológicas, Químicas y de la Salud; Ciencias Sociales y Humanidades y Artes.

La universidad hoy en día se encuentra al sur de la ciudad de México (anteriormente localizada en el centro de la ciudad de México). El lugar en el que ahora se encuentra se le llama Ciudad Universitaria (CU), siendo este el principal campus de la universidad.

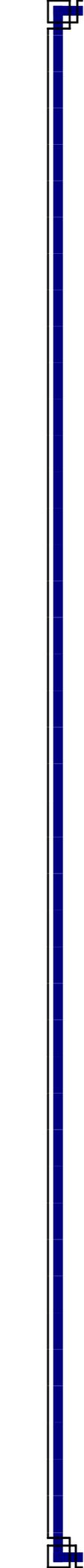
Desde que se dio inicio a su construcción en el año de 1950, la UNAM se ha ido extendiendo en todo el país. Al día de hoy la UNAM tiene presencia en todos los estados de la República Mexicana, por medio de sus diversas instalaciones destinadas a la educación, la investigación científica y la difusión cultural. Inclusive esta cuenta con sedes en otros países, entre los cuales tenemos a Costa Rica, Estados Unidos, Canadá, España, Francia, Sudáfrica, Reino Unido, China y Alemania.

Entre las diversas comodidades que ofrece la UNAM a sus estudiantes se encuentran las salas de cómputo, cuyo objetivo principal es que los alumnos en el cumplimiento de sus deberes puedan acudir a las salas de cómputo sin que tengan la necesidad de salir de su Facultad, de no tener que pagar por hacer uso del equipo, de disponer de servicio de internet, servicio de impresión, y de poder contar con software que les sea de utilidad.

La Unidad de Servicios de Cómputo Académico (UNICA) es la encargada de proporcionar en la Facultad de ingeniería los servicios que requieren del uso y del conocimiento de las Tecnologías de la Información y la Comunicación (TIC), y de las Tecnologías de Aprendizaje y el Conocimiento (TAC).

Entre los servicios que proporciona UNICA a los alumnos de la Facultad están: servicio de impresión, préstamo de equipo, asesorías en temas de cómputo, además de disponer de un área con wifi.

El proyecto de tesis a realizar es la implementación de un servidor de base de datos que recopile información relacionada al control y funcionamiento de las salas de cómputo. A su vez, se implementará una aplicación para dispositivos móviles desde la cual se podrá acceder al servidor de base de datos con la intención de disponer de un medio de información que mantenga al tanto a los usuarios sobre los servicios y actividades de las salas de UNICA, y que facilite al personal la toma de decisiones relacionadas a la organización y funcionamiento de las salas de cómputo.



# **CAPÍTULO 1**

## **ANTECEDENTES**

# Capítulo 1 ANTECEDENTES

## INTRODUCCIÓN

Los progresos que ha tenido la sociedad hoy en día se deben en gran parte a que hemos aprendido a trabajar de forma conjunta. Esto ha dado lugar al surgimiento de sistemas en los que un grupo de personas actúan de forma coordinada para lograr una misión. A este tipo de sistemas se les conoce con el nombre de organización.

Para que una organización pueda ser funcional es necesario contar con un grupo de recursos, entre los cuales podemos considerar como recursos humanos, tecnológicos y económicos.

Dentro de una organización, todos aquellos datos que proporcionan información que sea de utilidad para su funcionamiento deberán ser gestionados mediante algún recurso que facilite su manejo. Una organización puede recurrir al uso de las bases de datos con el objetivo de tener un mejor control de la información.

La importancia de un sistema de base de datos dentro de la organización dependerá de la misma organización (esto en base al tipo de organización, de sus alcances y necesidades).

## CONSTRUCCIÓN DEL OBJETO DE ESTUDIO

La Universidad Nacional Autónoma de México es una institución educativa que ha desempeñado un papel protagónico en la historia y formación de nuestro país. Su importancia radica en la formación de profesionistas, capacitados para satisfacer las diversas necesidades que día a día se presentan en nuestro país. Para lo cual, las personas que forman parte de esta institución desarrollan toda clase de actividades, las cuales consisten principalmente en la docencia, la investigación y la difusión de la cultura.

La Facultad de Ingeniería busca formar profesionistas, de forma que sean capaces de resolver problemas de forma creativa e innovadora en el ámbito de su competencia.

Se busca que los estudiantes adquieran un deber con su formación desde un punto de vista ecológico, ético y humanista, adquiriendo un sentido de la responsabilidad y de compromiso que los motive a mantenerse actualizados constantemente. Así como fomentar en ellos el realizar investigación científica y aplicada de acuerdo a las necesidades de la sociedad.

Por su parte, la Secretaría General está en concordancia con la misión definida para la Facultad de Ingeniería, cuya esencia es la formación integral de profesionales calificados y comprometidos. Bajo este marco, y con la declaración expresa que la Secretaría General es una instancia de servicio y apoyo a todos los órganos que conforman la dependencia.

La Unidad de Servicios de Cómputo Académico (**UNICA**) es parte de la Secretaría General de la Facultad de Ingeniería, siendo la responsable de proporcionar servicios de cómputo académico para los alumnos de dicha institución, en apoyo a las actividades académicas que fortalezcan el proceso de formación de los alumnos de licenciatura, intercambio y posgrado.

Los servicios de cómputo de la Facultad de Ingeniería son el resultado de un proceso de innovación. Para esto, la Facultad en un inicio creó el Centro de Cálculo de la Facultad de Ingeniería (CECAFI) en el año de 1972, el cual se encargaba de dirigir el desarrollo en materia de cómputo de la Facultad.

En 1980 se crea un plan de becarios, el cual hoy se le conoce como Programa de Formación de Becarios de UNICA.

En 1987 se establece otra sala para proporcionar a los alumnos el préstamo de equipo de cómputo, ubicada en el Conjunto Sur, hoy conocida como Sala UNICA 2.

En 1994 aparece UNICA y USECAD, cuyas funciones son proporcionar servicios académicos y proporcionar servicios administrativos respectivamente. Esto porque CECAFI se dividió en estos dos órganos de la Facultad.

En 1996 se proporciona una tercera sala, hoy conocida como Sala UNICA 3.

Los fundamentos bajo los cuales se sustenta UNICA para llevar a cabo la administración de los servicios de cómputo académicos de la Facultad de Ingeniería son:

El **Objetivo** consiste en proporcionar los servicios de apoyo en cómputo enfocado a los alumnos de la Facultad de Ingeniería que requieren en su formación para la realización y cumplimiento eficaz de sus tareas académicas de manera sustantiva.

La **Misión** consiste en proporcionar eficaz y eficientemente en el ámbito institucional, los servicios de tecnologías de la información y comunicaciones, en apoyo a las actividades relacionadas, que coadyuven al proceso integral de formación académica en la Facultad de Ingeniería, UNAM.

La **Visión** consiste en ser un referente dentro de la Universidad en el sector de las TIC, y para ello abarcaremos todos los servicios que ofrecemos actualmente agregando aquellos que vayan surgiendo debido a la necesidad de cambio provocado por los avances tecnológicos.

Los **Valores** consisten en establecer un ambiente de cordialidad, respeto, honestidad, responsabilidad, ética y compromiso entre los integrantes de UNICA.

La **Política de calidad** consiste en cumplir con los requisitos de los usuarios en materia de las TIC. La meta es actualizar y elevar la calidad de los productos y servicios, para lo cual el personal se compromete a trabajar en un proceso de mejora continua.

## ORGANIGRAMA

Para gestionar los diferentes aspectos bajo los cuales opera la Facultad de Ingeniería, esta se divide en las áreas que se muestra en la Figura 1.1.



**Figura 1.1** Organigrama de las áreas de la Facultad de Ingeniería.

Fuente: Elaboración propia <sup>1</sup>

Como se mencionó anteriormente, para poder proporcionar los servicios de cómputo se hace uso de salas de cómputo, las cuales se encuentran distribuidas dentro de la Facultad.

Cabe recalcar que, el complejo de la Facultad de Ingeniería está organizado en una zona norte y una zona sur (siendo la primera la zona principal de la Facultad de Ingeniería, y la otra un anexo).

Por lo cual, la distribución de las salas de cómputo en la Facultad de Ingeniería es la que se muestra en la Figura 1.2 (a) y en la Figura 1.2 (b), correspondientes a zona norte y zona sur respectivamente.

Es así que las salas que están a cargo de UNICA para proporcionar los servicios de cómputo en la Facultad de Ingeniería son:

Sala UNICA 1, ubicada en el edificio principal de ingeniería (zona norte).

Sala UNICA 2, ubicada debajo del auditorio Sotero Prieto (zona sur).

Sala UNICA 3, ubicada en la planta baja de la torre de ingeniería (zona sur).

Sala UNICA 4, ubicada en el posgrado de ingeniería (zona sur).

<sup>1</sup> Figura ejemplificativa de las áreas bajo las cuales opera la Facultad de Ingeniería.



**Figura 1.2** Ubicación de las Salas de Cómputo UNICA.  
 Fuente: <http://www.fi-a.unam.mx/unica/instalaciones/index.php> <sup>2</sup>

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Se requiere diseñar e implementar un servidor de base de datos que contenga el registro de toda la relación de los servicios, actividades y recursos de las salas de cómputo, proporcionando información que promueva el uso de los servicios y equipos, y que facilite la administración de las actividades de las salas de cómputo.

## 1.2 OBJETIVO

### OBJETIVO GENERAL

Diseñar e implementar un servidor de base de datos que permita gestionar y consultar información relacionada al control y funcionamiento de las salas de cómputo mediante el Sistema de Servicios y Control de Salas de Cómputo “SISEC” de la Facultad de Ingeniería, y que favorezca el funcionamiento de las salas de cómputo de la Facultad de Ingeniería.

### OBJETIVOS ESPECÍFICOS

- Disponer de un medio que permita acceder a toda aquella información que sea de interés de forma fácil, rápida y sencilla.
- Facilitar la toma de decisiones en relación a la administración de los recursos y servicios de las salas de cómputo.

<sup>2</sup> Ubicación de las Salas de Cómputo de la Facultad de Ingeniería.

- Mejorar la comunicación entre los integrantes que conforman el personal de UNICA.
- Proporcionar un servicio de mayor calidad a los estudiantes interesados en hacer uso del servicio de los equipos de las salas de cómputo de la Facultad de Ingeniería.
- Supervisar el buen funcionamiento de las salas de cómputo.
- Capturar información relevante para el buen funcionamiento de las salas.

### **1.3 JUSTIFICACIÓN**

La Unidad de Servicios de Cómputo Académico (UNICA) se caracteriza por ser la responsable de brindar y gestionar los servicios sobre el uso del equipo de cómputo por parte de los estudiantes de la Facultad. Por otro lado, uno de sus objetivos es mantenerse en un proceso de mejora continua que le permita mejorar sus funciones, y ser más óptima en cuanto al servicio a los usuarios.

El propósito de implementar un servidor de base de datos es brindar una herramienta que facilite la consulta de información y toma de decisiones en relación a las actividades, servicios y recursos de las salas de cómputo. La información recopilada será empleada para llevar un control del personal, mantenimiento de los equipos, control de inventario de los equipos, y brindar información sobre el software, servicios y cursos a los usuarios, y al personal de las salas de cómputo.

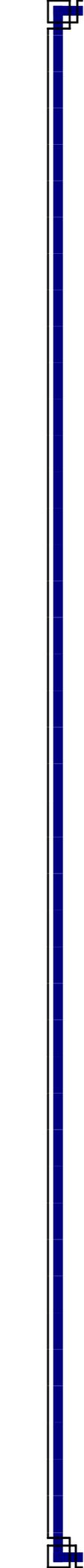
El proyecto a realizar será el desarrollo de un sistema de base de datos, donde el medio para acceder a la base de datos será una aplicación web. Dicho proyecto formará parte del sistema SISEC, para el cual habrá un segundo trabajo, en el cual se planteó el desarrollo de una aplicación para dispositivos móviles como interfaz de usuario, de forma que los usuarios puedan acceder a información de los servicios y actividades de las salas de UNICA desde sus propios dispositivos móviles.

Además, en la base de datos se podrá registrar información en forma de avisos sobre posibles eventualidades en las que se vea comprometido el desempeño del servicio en alguna de las salas de cómputo, algún cambio de las actividades de las salas, o sobre algún evento, esto para difundir información que sea de interés para los usuarios, o información que esté dirigida al personal. De igual forma, habrá otra sección donde se lleve un registro en forma de reportes sobre posibles eventualidades que comprometan el desempeño de los equipos de las salas de cómputo, esto para llevar un control de calidad del funcionamiento de los equipos. Estos reportes serán considerados por el personal para resolver dichos problemas, y una vez se le haya dado seguimiento al reporte, estos se mantendrán sobre la base de datos para poder ser consultados por otros usuarios del personal interesados en efectuar medidas preventivas/correctivas sobre otros equipos, o para identificar problemas que se presentan con regularidad e identificar las posibles soluciones para tales problemas.

#### **1.4 RESULTADOS**

Como resultado se tendrá un sistema de base de datos conformado de un servidor, el cual contenga la base de datos, además de una página web con la cual los usuarios podrán efectuar la consulta y actualización de información relacionada a las actividades, servicios y recursos de las salas de cómputo. Así el personal de las salas podrá hacer uso de este medio para gestionar la información mencionada, fortaleciendo así el desempeño y funcionamiento de las salas de cómputo de la Facultad de Ingeniería.

El servidor también estará disponible a los usuarios, para dar a conocer los servicios y recursos proporcionados en las salas de cómputo, para que los usuarios aprovechen la mayor parte de tales recursos que les sean de utilidad.



# **CAPÍTULO 2**

## **FUNDAMENTOS TEÓRICOS**

## Capítulo 2 FUNDAMENTOS TEÓRICOS

### 2.1 SOFTWARE

#### 2.1.1 Definición de software

El software es la parte lógica de aquellos equipos electrónicos que disponen de un procesador. Considerando por ejemplo dispositivos tales como las computadoras o los celulares, estos cuentan con una parte física o hardware que es todo aquello que es tangible. En cambio, el software es la parte lógica, y aunque es almacenado en una unidad física, se le considera un recurso intangible.

Los equipos electrónicos deben su funcionamiento al procesador aunado con el software, esto es porque el software contiene instrucciones predefinidas que a su vez son ejecutadas por el procesador. Se utiliza el software para lograr una tarea en concreto, ya sea para que el dispositivo ejecute programas, aplicaciones, algoritmos o para controlar los componentes que forman parte del hardware.

#### 2.1.2 Clasificación de software

En términos generales, el software de los dispositivos mencionados en la sección anterior es clasificado en tres grupos. Estos son el software operativo o de sistema, el software de desarrollo y el software de aplicación, los cuales se describen a continuación.

- **Software de sistema**

El software operativo es un conjunto de programas y rutinas que sirven para gestionar los recursos del sistema, proporciona un control en el caso de los recursos del hardware y da soporte a algunos programas. Los siguientes son ejemplos de software operativo.

##### *Controladores*

Un controlador o driver es el software que utiliza la computadora para establecer una comunicación de uno de los dispositivos hardware con el sistema operativo. Algunos ejemplos de estos controladores son los de las tarjetas de video, de sonido o de red.

##### *Sistemas operativos*

De manera general, un sistema operativo es el software encargado de coordinar las operaciones de la computadora para su funcionalidad, atiende los procesos realizados por los usuarios para que estos puedan interactuar con el equipo y ayuda a otros programas en su funcionamiento.

### **Antivirus**

Un antivirus es un programa que mantiene segura la computadora del software malicioso. Su función es la de acceder a los dispositivos de almacenamiento e inspecciona cada uno de los archivos con el fin de detectar y eliminar aquellos virus y amenazas que ponen en riesgo la integridad de los recursos.

### **Diagnóstico**

El software de diagnóstico es un programa que tiene la función de realizar un diagnóstico del equipo para detectar fallas, ya sea fallas a nivel tanto hardware como software, y mediante un aviso se informa al usuario sobre estas fallas y sus posibles soluciones.



**Figura 2.1** *Software de sistema.*

Fuente: Elaboración propia <sup>3</sup>

#### ▪ **Software de desarrollo**

Se le denomina software de desarrollo a las herramientas que utiliza un programador para generar, ya sea, un conjunto de instrucciones que indiquen a los equipos electrónicos las funciones que deben hacer para que estos puedan funcionar, o para desarrollar programas informáticos. Los siguientes son ejemplos de software de desarrollo.

### **Entornos IDE**

Un IDE o entorno de desarrollo integrado es un entorno de programación que agrupa herramientas de desarrollo para la creación de programas. Los elementos básicos que conforman al IDE son un editor de código, un compilador, un depurador y una interfaz gráfica de usuario (GUI).

---

<sup>3</sup> Imagen ejemplificativa de las diferentes plataformas de sistemas operativos en el mercado.

URL de las imágenes: <https://www.pinterest.com.mx/>

Un IDE puede estar orientado a un solo lenguaje de programación. Un ejemplo de esto es el IDE Dev C++, el cual se utiliza para desarrollar programas en lenguaje C++. Por otro lado, se tienen otros tipos de IDE, los cuales pueden manejar diferentes lenguajes de programación. Algunos ejemplos de estos tipos de IDE son Eclipse, NetBeans y Visual Studio, los cuales pueden ser utilizados para algunos lenguajes de programación como lo son C++, Python, Java, C#, entre otros.



**Figura 2.2** *Software de desarrollo.*

Fuente: Elaboración propia <sup>4</sup>

### ***Editores de texto***

Un editor de texto es un programa que permite crear y modificar archivos de texto plano. Los editores de texto son de utilidad en la programación, esto se debe a que el código fuente de los programas es desarrollado en texto plano. Los programadores hacen uso de los editores de texto para generar archivos de texto que contengan el código fuente de sus programas, estos archivos a su vez son usados por los compiladores para ejecutar el código del programa.

<sup>4</sup> URL de las imágenes: <https://www.wikipedia.org/>

Cuando se recurre a un editor de texto para programación, se puede recurrir al editor que se encuentra instalado en el sistema operativo, o se pueden instalar editores especiales de programación. En el caso de un entorno IDE, estos ya están incluidos en el programa.

### ***Compiladores***

Un compilador es una aplicación, la cual procesa el código de un programa para que este pueda ser ejecutado, ya sea que un programador haga uso de este para poder ejecutar algún código de programación, o para que un usuario pueda ejecutar cualquier programa de aplicación. El proceso que realiza el compilador consiste en traducir el código fuente a código objeto, es decir, traducir las instrucciones que están escritas en un lenguaje de alto nivel (esto es, un lenguaje que es fácil de interpretar por los humanos), a un lenguaje de bajo nivel (un lenguaje que solo puede ser interpretado por las máquinas), y esto es necesario para que puedan ser interpretadas dichas instrucciones por el ordenador y ejecutadas por los dispositivos del ordenador.

### ***Depuradores***

Los depuradores son una herramienta que forman parte de los entornos de programación, los cuales ayudan a los programadores durante el desarrollo de software a detectar errores en el código fuente.

- **Software de aplicación**

El Software de aplicación agrupa todos aquellos programas que utilizan los usuarios para resolver problemas o para realizar todas aquellas actividades para las cuales ha sido diseñado el software. Básicamente, este tipo de software es clasificado en tres grupos, los cuales se describen a continuación.

### ***Software de ofimática***

Tomando en cuenta que la palabra ofimática es un acrónimo entre las palabras oficina e informática, queda claro que la expresión software de ofimática (también denominado suite ofimática), hace referencia a un conjunto de programas que suelen ser de uso común para actividades de oficina, entre los cuales se tienen procesadores de texto, hojas de cálculo, gestores de base de datos, creador de presentaciones, entre otros.

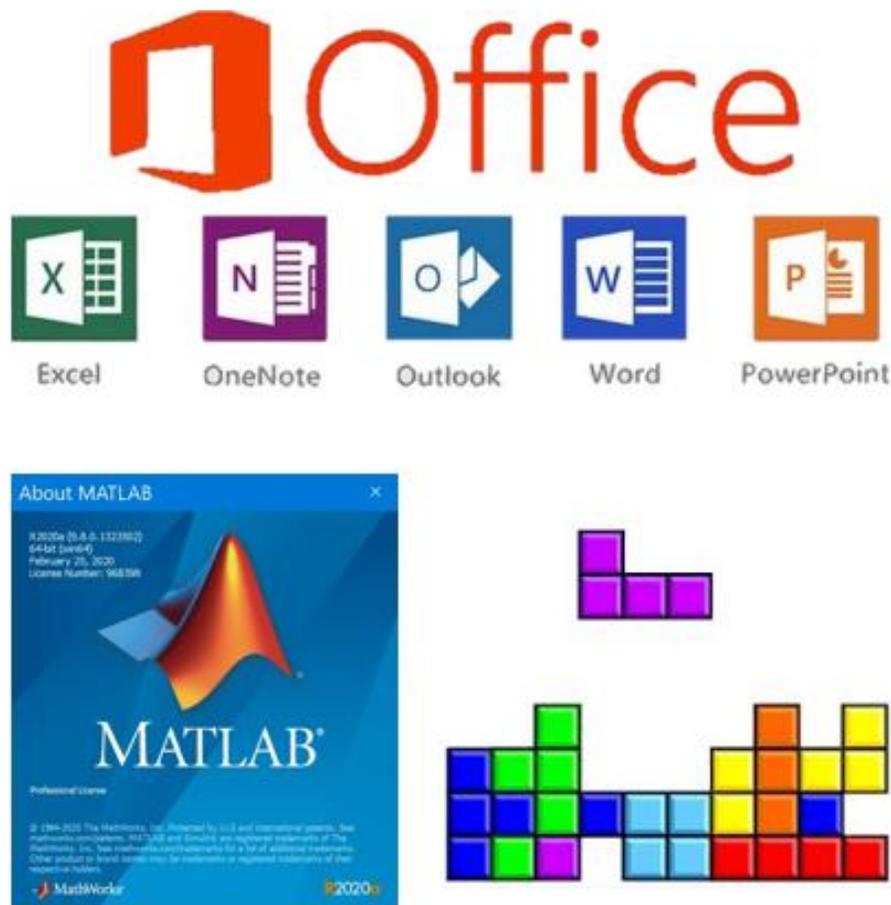
### ***Software especializado***

Dentro de la categoría de software especializado tenemos a los programas de diseño asistido y los simuladores, puesto que los programas están diseñados para ser ocupados en proyectos o situaciones muy particulares y complejas. Las aplicaciones de este tipo de software son las de diseño gráfico, simulación, análisis, procesamiento de datos, entre otras.

El software de diseño asistido por computadora (CAD) y los simuladores son ampliamente usados por profesionistas, estudiantes o por cualquier persona interesada en el diseño y la planeación gracias a sus múltiples cualidades.

### **Software de entretenimiento**

El software de entretenimiento son aquellas aplicaciones que brindan al usuario contenido que resulte de su agrado, de este tipo de contenido tenemos videojuegos, videos, música, entre otros.



**Figura 2.3** Software de aplicación.

Fuente: Elaboración propia <sup>5</sup>

<sup>5</sup> URL de las imágenes: <https://www.pinterest.com.mx/>

## 2.2 SISTEMAS OPERATIVOS

### 2.2.1 Definición de sistema operativo

Un sistema operativo, como ya se mencionó en la sección anterior, es el software encargado de coordinar las operaciones de la computadora para su funcionalidad. Este facilita el uso de la computadora a los usuarios. Le proporciona una interfaz gráfica al usuario siendo esto un medio para que el usuario pueda interactuar con el equipo y pueda observar los procesos que se están ejecutando sobre el equipo. Establece una comunicación entre los dispositivos hardware con las aplicaciones. Se ejecuta cada vez que es encendida la computadora para que esta quede lista para su uso. Estas y muchas otras son las funciones de las que está a cargo el sistema operativo siendo este el software más importante de la computadora.

### 2.2.2 Clasificación de sistemas operativos

- **Sistemas operativos para equipo de cómputo**

A continuación, se da una breve descripción de los sistemas operativos disponibles en el mercado de mayor popularidad.



**Figura 2.4** Software operativo.

Fuente: Elaboración propia <sup>6</sup>

#### **UNIX**

UNIX es una familia de sistemas operativos, la cual surgió en la década de 1960 en los laboratorios Bell. Considerando a UNIX como un solo sistema operativo, se caracteriza por ser un sistema operativo portable, multitareas y multiusuario. Algunas de sus versiones son comerciales con derechos de autor y algunas son de código abierto.

---

<sup>6</sup> Imagen ejemplificativa de las diferentes plataformas de sistemas operativos en el mercado.

URL de los logos: <https://www.sololinux.es/>

### ***GNU/Linux***

GNU/Linux es un sistema operativo derivado de UNIX. Sus características son muy similares a las de UNIX. Aunque este se destaca por poseer una licencia libre y gratuita en la mayoría de sus versiones (también llamadas distribuciones de GNU/Linux o Linux).

### ***Windows***

Microsoft es una compañía especializada en el desarrollo de software. Es mejor conocida por desarrollar el sistema operativo Windows (lanzado por primera vez en el año de 1985 y mejorado con nuevas versiones).

### ***MacOS***

MacOS es un sistema operativo desarrollado por la empresa Apple. Este sistema operativo fue creado exclusivamente para las computadoras Macintosh y otros dispositivos de Apple.

- **Sistemas operativos para dispositivos móviles**

Los principales sistemas operativos de los dispositivos móviles (dispositivos tales como celulares, tabletas, relojes) son iOS y Android, los cuales se describen a continuación.

### ***iOS***

iOS es un sistema operativo para dispositivos móviles de la empresa Apple. Dispositivos tales como celulares, tabletas y reproductores mp4, equivalente a los iPhone, los iPads y los iPod respectivamente.

### ***Android***

Android es un sistema operativo para dispositivos móviles basado en el sistema operativo GNU/Linux. Fue desarrollado por Android Inc., la cual fue adquirida en 2005 por la compañía Google.



**Figura 2.5** *Sistemas operativos móviles.*

Fuente: <https://www.sololinux.es/conectar-un-android-o-iphone-con-manjaro-y-arch/><sup>7</sup>

---

<sup>7</sup> Sistemas operativos para dispositivos móviles, 2021.

### 2.2.3 Distribuciones GNU/Linux

A pesar de que la mayoría de las distribuciones GNU/Linux pueden ser usadas de forma gratuita, hay otras que no, la diferencia está en que aquellas distribuciones que son de paga incluyen aplicaciones comerciales o soporte técnico por suscripción. A continuación, se muestra una agrupación de las diferentes distribuciones en base a si son sistemas operativos gratuitos, o si son de paga, y se proporciona una breve descripción de cada uno de estos.



**Figura 2.6** Distribuciones GNU/Linux.

Fuente: Elaboración propia <sup>8</sup>

- **Distribuciones de paga**

Las distribuciones GNU/Linux que son de paga están orientadas para uso empresarial. Su uso es para aquellas empresas que desean obtener soluciones a la medida y a bajo costo en comparación a otros sistemas operativos tales como Microsoft o Apple. Entre las distribuciones GNU/Linux que son de paga tenemos las siguientes:

---

<sup>8</sup> Algunas de las diferentes distribuciones del sistema operativo GNU / Linux. URL de los logos: <https://www.sololinux.es/>

### ***Red Hat***

Red Hat Enterprise Linux (RHEL) es una distribución comercial, esta incluye un conjunto de programas comerciales junto con un soporte técnico. Red Hat es utilizada en empresas y servidores.

### ***SuSE***

SuSE Linux Enterprise (SLE) es una distribución comercial, siendo la más sencilla de usar en entornos corporativos, mientras que es una distribución comunitaria. Dichas distribuciones disponen de recursos que facilitan la migración de una distribución a otra.

## ▪ **Distribuciones gratuitas**

Entre las distribuciones GNU/Linux que son de uso gratuito, por mencionar algunas, se tienen las que se muestran a continuación.

### ***CentOS***

Se puede considerar a CentOS como una distribución de gran calidad debido a que está continuamente siendo actualizada en base al código que Red Hat proporciona gratuitamente al público desde el 2004. Esta distribución está destinada al uso empresarial.

### ***Debian***

Debian es un sistema operativo muy confiable, por lo que es muy utilizado en el mundo académico, científico y en el sector público. Además, algunas distribuciones populares tales como Ubuntu están basadas en esta distribución.

### ***Fedora***

Fedora es una versión gratuita de Red Hat para usuarios domésticos. Dependiendo de la versión se puede usar en entornos de escritorio, en servidores o para un sistema en la nube. Además, Fedora es una distribución segura gracias a su sistema de seguridad SELinux.

### ***Slackware***

Slackware es una distribución diferente a las otras, está orientado para aquellos usuarios que estén interesados en aprender a usar el sistema operativo de una forma más técnica, ya que algunos procesos de configuración del sistema se deben realizar de forma manual.

### ***Ubuntu***

Ubuntu es una de las distribuciones más utilizadas debido a que es fácil de usar. Cuenta con dos tipos de distribuciones: la distribución Ubuntu Desktop para equipos de escritorio, y la distribución Ubuntu server para utilizarse en equipos servidores.

## 2.2.4 Comparativa entre distribuciones Linux

En la tabla 2.1 se muestra una evaluación de algunas de las distribuciones Linux en base a las características más relevantes de un sistema operativo. Para realizar la comparativa se seleccionaron las distribuciones de CentOS, Debian, openSUSE y Ubuntu Server.

La evaluación de cada uno de estos aspectos se realizó asignando una calificación, siendo que, una calificación sobre 10 hace referencia a que la distribución más se apega con dicho parámetro, y una calificación menor a 10 que la distribución cumple de manera parcial con dicho parámetro. A continuación, se muestra la evaluación de las distribuciones Linux en base a los parámetros seleccionados.

Nombre	Proceso de instalación amigable	Soporte técnico	Documentación	Consumo de recursos	Seguridad
CentOS	9/10	10/10	10/10	10/10	9/10
Debian	7/10	7/10	10/10	10/10	8/10
Open SUSE	8/10	7/10	9/10	8/10	6/10
Ubuntu Server	9/10	8/10	8/10	6/10	8/10

**Tabla 2.1** Comparativa entre distribuciones Linux.

Fuente: ESTUDIO COMPARATIVO DE LAS DISTRIBUCIONES LINUX ORIENTADO A LA SEGURIDAD DE REDES DE COMUNICACIÓN <sup>9</sup>

### ▪ Proceso de instalación amigable

Esta característica es tomada en consideración ya que hay distribuciones para las cuales se requiere de conocimientos previos para su instalación. Por lo que habrá distribuciones que cuente con una interfaz más versátil que otras, y que contenga información de ayuda para que sea más fácil su instalación. Entre las distribuciones mejores evaluadas en cuanto a un proceso de instalación amigable tenemos a CentOS y Ubuntu Server.

---

<sup>9</sup> ESTUDIO COMPARATIVO DE LAS DISTRIBUCIONES LINUX ORIENTADO A LA SEGURIDAD DE REDES DE COMUNICACIÓN (pp. 67 - 83), por Badillo, D. 2015, Pontifica Universidad Católica del Ecuador.

- **Soporte técnico**

Las distribuciones Debian y Ubuntu Server cuenta con versiones con soporte a largo plazo, es decir LTS (Long Time Support) que reciben soporte durante 5 años. Las versiones de openSUSE posteriores a la versión 13.2 se les conoce como openSUSE Leap, y cuentan con un soporte de hasta 3 años. En el caso de CentOS, desde la versión CentOS 5 en adelante, cada una cuenta con un mantenimiento de 10 años.

- **Documentación**

Tomando en consideración la documentación oficial en línea que proporciona cada distribución y la documentación que se proporciona en otros sitios no oficiales, las distribuciones que disponen de una mayor documentación son Debian y CentOS. Dicha documentación es referente a, por dar algunos ejemplos, instalación del sistema, configuración de servicios, foros de ayuda para problemas, manejo de herramientas, etcétera.

- **Consumo de recursos**

La distribución les da un uso a los recursos hardware en mayor medida dependiendo de qué tanta prioridad se le da al entorno gráfico. Esto es algo a considerar ya que entre menos recursos ocupe una distribución, mayor es su capacidad de poder funcionar en cualquier equipo, sin que importe mucho el tipo de equipo y sus características.

La distribución que más recursos consume es Ubuntu Server, esto por utilizar el entorno gráfico Unity. En cambio, las distribuciones más accesibles son Debian y CentOS.

- **Seguridad**

Las distribuciones Linux cuentan con una arquitectura de seguridad (SELinux), la cual se encuentra integrada al kernel del sistema. Sin embargo, es importante que las distribuciones tengan soporte para esta arquitectura.

Además de la arquitectura SELinux, algunas distribuciones hacen uso de AppArmor, siendo esta un equivalente a SELinux. Las distribuciones Debian, Ubuntu y openSUSE soportan AppArmor y SELinux. En cambio, en el caso de CentOS, este solamente soporta SELinux.

Otro factor a tomar en cuenta en la seguridad de las distribuciones es el tiempo de soporte, es decir, el tiempo al que los desarrolladores están dispuestos en publicar actualizaciones de seguridad en los paquetes instalados. Además de la seguridad, cuanto más tiempo de soporte recibe una distribución, es más conveniente realizar proyectos a largo plazo haciendo que una distribución sea menos propensa a fallos. De entre todas las distribuciones, CentOS es la que mayor tiempo de soporte ofrece a los usuarios.

## 2.3 CONCEPTOS DE BASES DE DATOS

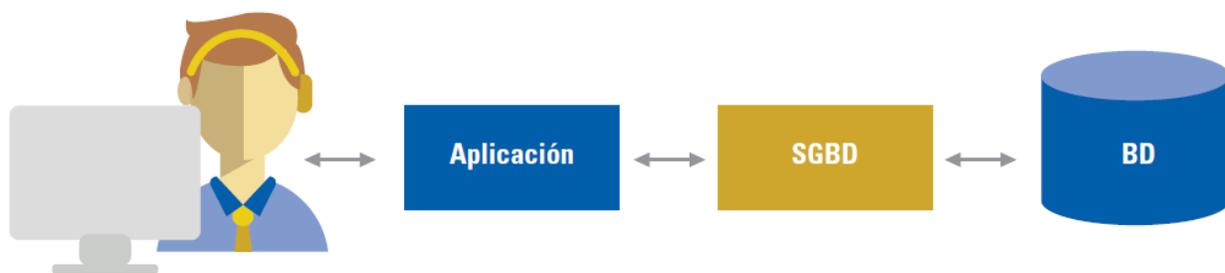
### 2.3.1 Definición de sistema de base de datos

Una base de datos es una colección de datos interrelacionados, los cuales representan información significativa para una organización o empresa. Esta información puede tener diversas aplicaciones. Realizando un análisis de los datos se puede obtener información valiosa que puede ser utilizada para la toma de decisiones, para dirigir y administrar una empresa, o para cualquier otro objetivo.

Una base de datos es el depósito donde se almacenan todos los datos de manera estructurada, el cual suele almacenar una gran cantidad de información. Para poder gestionar toda esa información se utiliza un sistema gestor de base de datos SGBD o DBMS por sus siglas en inglés (Data Base Management System), el cual consiste en un conjunto de programas que se utilizan para almacenar, acceder, y modificar los datos de forma rápida y sencilla. Esto permite tener un mayor control de la información.

Un sistema de base de datos es un conjunto de elementos que permite registrar y mantener la información. Los elementos que integran a un sistema de base de datos son los datos, el hardware, el software y los usuarios.

El programa de aplicación es el medio por el que el usuario va a acceder a la base de datos. El programa de aplicación proporciona al usuario una interfaz gráfica que lo ayudará a visualizar la información, omitiendo muchos de los detalles de la base de datos y mostrando solo aquello que sea relevante para el usuario. El SGBD se encarga de extraer la información de la base de datos y proporcionarla al usuario mediante el programa de aplicación. En la Figura 1.1 se muestra la interacción de estos elementos.



**Figura 2.7** Relación usuario, aplicación de bases de datos, SGBD y base de datos.

Fuente: DISEÑO DE BASE DE DATOS <sup>10</sup>

<sup>10</sup> DISEÑO DE BASE DE DATOS (p. 16), por Merchán, O. 2018, Universidad del Azuay.

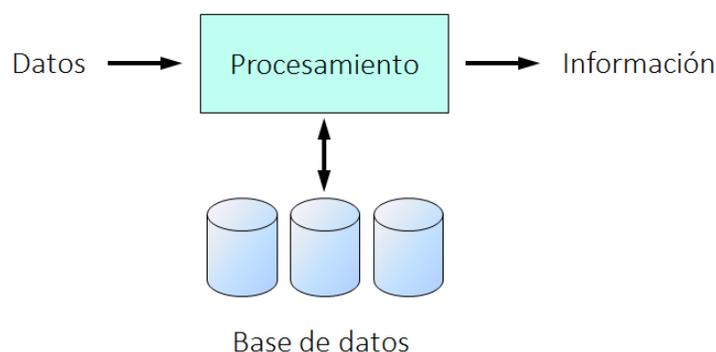
### 2.3.2 Elementos de un sistema de base de datos

Tenemos presente que los cuatro componentes principales que constituyen a un sistema de base de datos son: los datos, el hardware, el software y los usuarios. A continuación, veremos a profundidad en qué consisten cada uno de estos.

- **Datos**

Los datos hacen referencia a una situación del mundo real y representan algo significativo. Entre los recursos de una empresa, los datos son considerados como uno de sus activos más importantes, ya que suelen tener algún tipo de aplicación por parte de los usuarios. Dada su importancia, es imprescindible llevar un manejo adecuado de los datos, ya que al procesarlos se obtiene información que es de utilidad en la toma de decisiones o en la gestión de una empresa.

Podemos considerar tres tipos de datos de acuerdo a la manera en la que interactúan con la base de datos, siendo estos los datos de entrada, datos de salida y datos de operación. Los datos de entrada son aquellos que son ingresados al sistema y pueden generar un cambio en los datos de operación. Los datos de salida son aquellos que se generan a partir de la información que contiene el sistema y que proporcionan un mensaje o cierta información. Los datos de operación son aquellos que forman parte del sistema, describen el funcionamiento de la empresa y estos intervienen en la generación de los datos de salida.



**Figura 2.8** Flujo de información en un sistema de base de datos.

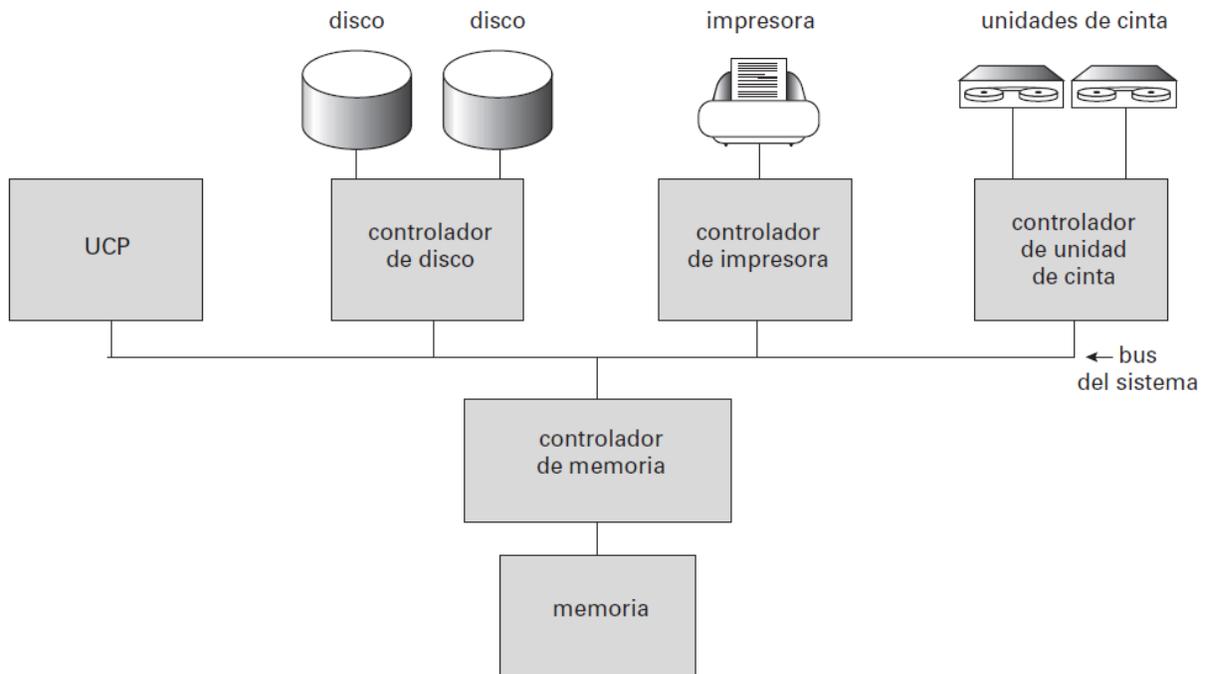
Fuente: Elaboración propia <sup>11</sup>

---

<sup>11</sup> Obtención de la información a través del procesamiento de los datos.

- **Hardware**

Los elementos que constituyen una base de datos varían dependiendo del tipo de base de datos que se necesite usar. En caso de que esta sea una base de datos centralizada, los elementos que la constituyen son aquellos que se muestran en la Figura 2.2. Este tipo de sistemas están constituidos de varias computadoras, las cuales a su vez están conectadas a otros dispositivos. Cada computadora tendrá una unidad central de procesamiento UCP, un cierto número de controladores para los dispositivos y una memoria cache. Las computadoras y los dispositivos están conectados por medio del bus del sistema. Gracias a todos estos elementos, las computadoras pueden interactuar con los dispositivos, los cuales pueden ser monitores, impresoras u otros elementos, además de aquellos elementos en los que se tengan almacenados los datos.



**Figura 2.9** *Un sistema informático centralizado.*  
Fuente: FUNDAMENTOS DE BASES DE DATOS <sup>12</sup>

<sup>12</sup> FUNDAMENTOS DE BASES DE DATOS (p. 446), por Silberschats, A., Korth, H. F. y Sudarshan, S., 2002, McGRAW-HILL/INTERAMERICANA DE ESPAÑA.

- **Software**

El SGBD constituye el software de un sistema de base de datos, el cual se compone de un conjunto de programas que permite a los usuarios realizar todo tipo de actividades que implican el uso de los datos. Para esto el usuario tiene que hacer uso de un lenguaje con el que podrá solicitar al SGBD realizar funciones tales como consulta o actualización de los datos, o especificar el esquema de la base de datos. Los lenguajes que se emplean en un SGBD son:

***Lenguaje de definición de datos***

Se le denomina LDD o DDL por sus siglas en inglés (Data Definition Language). Es un lenguaje especial a partir del cual, mediante el uso de un conjunto de definiciones se especifica el esquema de la base de datos.

Así mismo, el lenguaje DDL permite a los usuarios especificar entidades, atributos y relaciones para definir los esquemas y sus relaciones. Este lenguaje además permite especificar ciertas restricciones para el almacenamiento y modificación de los datos, esto es, restricciones de integridad y acciones de activación referencial.

Una base de datos para poder almacenar los datos en su repositorio, esta dispone de otro tipo de datos cuya función es describir aquellos datos que almacena. Estos datos se denominan metadatos y son almacenados en una tabla especial llamada diccionario de datos (DD).

Los diccionarios de datos se crean a partir de la ejecución de sentencias DDL. Esto permite al sistema tener un depósito en donde pueda almacenar los metadatos, los cuales realizan la descripción de los esquemas de la base de datos. Un sistema de base de datos para realizar las funciones de consulta o actualización debe recurrir al diccionario de datos.

***Lenguaje de control de datos***

El lenguaje de control de datos proporciona un control de acceso a la base de datos. Para poder controlar el acceso se debe identificar el tipo de usuario que ingresa a la base de datos, y posterior a esto, se atienden las solicitudes de los usuarios. A cada usuario se le han delegado privilegios, y son los privilegios los que definen cada una de las acciones que el usuario tiene permitido realizar sobre la base de datos. Por tanto, un usuario entre más privilegios tenga, dispondrá de un mayor control sobre la base de datos.

El usuario administrador de la base de datos es el encargado de definir los tipos de usuarios y de otorgarles privilegios. De igual forma, este también puede denegar privilegios a un usuario al que se le habían concedido privilegios con anterioridad.

Se emplea el concepto de role para designar un nombre a los diferentes tipos de usuarios, de forma que cada role reúne un conjunto de privilegios. Los roles pueden ser asociados a más de un usuario, y pueden conformarse de uno o más privilegios.

### ***Lenguaje de manipulación de datos***

Se le denomina LMD o DML por sus siglas en inglés (Data Manipulation Language). Este lenguaje da lugar a que los usuarios puedan acceder a la base de datos, ya sea que hagan uso de este lenguaje para formular consultas, o que realicen algún tipo de actualización sobre los datos, lo que es inserción, borrado o modificación de los datos. A su vez, existen dos tipos de lenguajes de manipulación de datos, estos son:

#### ***Procedimentales***

El usuario especifica los datos que necesita y como obtenerlos, lo cual requiere que el usuario tenga conocimientos sobre la estructura de los datos y sobre los procesos para acceder a los datos.

El lenguaje procedimental a pesar de ser un lenguaje más difícil de aprender y de usar en comparación con el lenguaje no procedimental, tiene la ventaja de ser un lenguaje con el que se puedan escribir procesos que sean más eficientes, motivo por el cual es un lenguaje utilizado por los usuarios informáticos.

#### ***No procedimentales***

El usuario especifica los datos que necesita, más no se requiere especificar el cómo obtenerlos. En este caso, el sistema es el que define el proceso para acceder a los datos. A este lenguaje también se le conoce como declarativo.

Para la declaración de sentencias DDL, DCL o LMD se utiliza el lenguaje de consultas SQL, el cual forma parte de uno de los lenguajes más ampliamente usado, aunque no es el único. En el caso de las sentencias DML, se dice que SQL es declarativo, aunque también tiene la opción para manejar sentencias del tipo procedimental.

### ***Programa de aplicación***

Un programa de aplicación es un programa que proporciona una interfaz de usuario, el cual funciona como un medio que permite a los usuarios interactuar con la base de datos a través de un entorno gráfico. De forma que estas aplicaciones cuentan con funciones requeridas por los usuarios, y dada la interfaz gráfica que proporciona el sistema mediante diálogos con ventanas, íconos y ratón, los usuarios pueden interactuar con mayor facilidad con la base de datos. Una interfaz de usuario permite a los usuarios normales interactuar con una base de datos sin necesidad de poseer conocimientos informáticos para manipularla. En cambio, los usuarios deben rellenar los campos de un formulario que les proporciona la interfaz.

Para poder desarrollar un programa de aplicación se requiere del uso de un lenguaje de programación. Sin embargo, la interacción con una base de datos mediante un lenguaje de programación puede implementarse de dos formas:

### ***SQL dinámico***

En este caso suele usarse una interfaz para programas de aplicación (API - Application Program Interface), con la que el programa puede crear consultas o actualizaciones en un sistema de base de datos, para lo cual se hace uso de un estándar de conectividad. El estándar de conectividad es una norma con la que se define la interfaz para programas de aplicación. El programa de aplicación utiliza la API para abrir una conexión con el servidor de bases de datos, permitiendo el envío de consultas, actualizaciones y la obtención de resultados.

Entre los lenguajes de programación utilizados en el desarrollo de un programa de aplicación se tienen dos lenguajes en particular, siendo estos el lenguaje C y el lenguaje Java. Esto es porque tanto el lenguaje C como el lenguaje Java disponen de la norma para la conexión con la base de datos, siendo la norma ODBC (Open Database Connectivity - Conectividad abierta de bases de datos) la correspondiente al lenguaje C, y la norma JDBC (Java Database Connectivity - Conectividad de bases de datos con Java) la norma de conectividad del lenguaje Java.

### ***SQL incorporado***

Se le denomina SQL incorporado dado que se adjuntan las instrucciones SQL de forma directa dentro del código del programa de aplicación. Para esto se necesita una sintaxis, la cual permita distinguir las instrucciones SQL del lenguaje de programación. En sí, la sintaxis consiste en el uso de una expresión propia del lenguaje de programación seguido de algún tipo de instrucción LMD. Por esta razón se usa el término lenguaje anfitrión, esto para hacer referencia al lenguaje de programación, el cual admite las consultas SQL dentro de su código. Mientras que al lenguaje SQL se le denomina el lenguaje incorporado u hospedado.

Sin embargo, es necesario hacer uso de un preprocesador denominado precompilador LMD, dado que la ejecución del programa se realiza mediante el compilador del programa anfitrión. Por lo que antes de que el programa sea compilado, es necesario que el precompilador LMD convierta las instrucciones LMD en declaraciones escritas y llamadas a procedimientos escritos en el lenguaje anfitrión.

El SGBD, además de ser un medio que permite a los usuarios interactuar con la base de datos, también dispone de ciertas cualidades que ayudan a que el sistema sea más eficiente en cuanto a su funcionamiento. En la sección 2.2.3 se mencionan a detalle los beneficios que proporciona el SGBD al utilizar un sistema de base de datos.

- **Usuario**

Los usuarios que forman parte de un sistema de base de datos deben su interacción con la base de datos para realizar una tarea en particular que implica el uso de datos. Según sea el tipo de interacción que estos tengan con la base de datos son clasificados en tres grupos.

***Administrador de base de datos***

El administrador de base de datos (ABD), o también denominado DBA por sus siglas en inglés (Data Base Administrator), es el usuario responsable de gestionar la base de datos, el SGBD y los programas asociados a la base de datos. La labor del administrador de la base de datos puede ser desempeñada ya sea por una persona o por un grupo de personas.

Una organización que hace uso de un sistema de base de datos requiere de este tipo de usuarios, puesto que es a él al que se le designan las funciones del diseño, el control y la administración de la base de datos. Cabe mencionar que su labor juega un papel muy importante dentro de la organización y además requiere de un alto nivel de destreza técnica.

Las funciones que desempeña el administrador de base de datos (por mencionar algunas) son las siguientes:

***Seleccionar la información que debe contener la base de datos***

Para esto el administrador de base de datos selecciona aquellas entidades que son relevantes para la organización (esto lo hace en base a las necesidades de la organización y de los usuarios). A su vez, el administrador de base de datos analiza la información asociada a dichas entidades y selecciona la más importante, esta información es la que se almacena en la base de datos.

***Definir el esquema de la base de datos***

Una vez que ha sido seleccionada la información a almacenar, lo siguiente es definir el esquema de la base de datos. Esta labor se realiza a través del registro de la información por medio de instrucciones escritas en lenguaje LDD.

### ***Control de acceso de los usuarios***

Como se ha mencionado en la sección anterior, la función de control de acceso de los usuarios es realizada por el administrador de base de datos. Así pues, a cada usuario se le proporciona la autorización de acceder a solamente una sección de la base de datos y se les proporciona permisos sobre las instrucciones que tienen permitido realizar sobre la base de datos.

### ***Desarrolladores del sistema***

El diseño de un sistema de base de datos requiere de un proceso constituido por etapas, todas ellas están vinculadas con alguno de los elementos que conforman al sistema de base de datos. Recordaremos que los elementos que constituyen al sistema de base de datos son: los datos, el hardware, el software y los usuarios. En cada uno de estos aspectos hay una persona encargada para atender los requerimientos y problemas que puedan surgir. Esto se hace con el fin de que haya un buen funcionamiento del sistema.

Una vez que ha sido seleccionada la información por parte del administrador de base de datos es posible realizar el diseño de la base de datos. Para lo cual se requiere del personal correspondiente para atender los aspectos de hardware, software y usuarios. Por tanto, las personas encargadas del diseño de la base de datos y las funciones que realizan son:

#### ***Diseñador de la base de datos***

El motivo por el cual es necesario realizar el diseño de un sistema de base de datos es para definir una estructura que sea de utilidad para almacenar la información. El diseño que se utiliza inicialmente es el diseño conceptual. Es así que, el diseño conceptual consiste en obtener una estructura de la información de la futura base de datos por medio de un modelo al que se le denomina modelo conceptual.

Seguido del diseño conceptual se realiza un segundo diseño al que se le denomina diseño lógico, y a partir de este se obtiene el modelo lógico. Mientras que el modelo conceptual es un modelo genérico en el que se representa información de interés del mundo real, el modelo lógico es una adaptación del modelo conceptual, y lo que caracteriza al modelo lógico es que es un modelo orientado a los programadores. Esto es porque en este modelo se describen los datos que se van a almacenar y las relaciones entre estos datos, esta información es la que se necesita para poder implementar el modelo sobre la base de datos.

Las organizaciones que disponen de grandes cantidades de información demandan de un mayor número de recursos para poder almacenar la información. Este es un factor a considerar en el diseño físico de la base de datos. Además de esto, otros aspectos a tomar en consideración son: la selección de la estructura física, aspectos relacionados a las

consultas y actualizaciones (tales como vías de acceso y frecuencia de ejecución), aspectos relacionados al almacenamiento (tales como tamaño de memoria y volumen de información), medidas de seguridad, entre otros.

### ***Programador de aplicaciones***

Por último, entre los desarrolladores del sistema se encuentran los programadores de aplicaciones, los cuales se encargan del desarrollo de los programas de aplicación. Estos programas sirven de interfaz a los usuarios para que estos puedan interactuar con la base de datos. Las aplicaciones tienen la función de recibir instrucciones realizadas por los usuarios y desplegar un resultado.

El programador de aplicaciones puede recurrir a ciertos recursos que facilitan el desarrollo de aplicaciones, entre los cuales se tiene un tipo de lenguaje especial denominado lenguaje de cuarta generación. Gracias a este lenguaje puede complementarse una interfaz de usuario con elementos tales como el uso de menús, presentación de datos y diálogos en pantalla, entre otros.

### ***Usuarios finales***

Los usuarios finales son personas que interactúan con la base de datos para realizar consultas, actualizaciones y generación de reportes. En general, este tipo de usuarios suelen ser clasificados en dos grupos en base en cuanto a la forma de acceder a la base de datos. Estos son:

#### ***Usuarios normales***

Son usuarios no informáticos que acceden de forma ocasional a la base de datos para realizar consultas, esto lo hacen a través de un programa de aplicación. El programa de aplicación ayuda a los usuarios a interactuar con la base de datos, de tal forma que no es necesario que el usuario tenga conocimientos sobre el uso del SGBD. En su lugar los programas de aplicación proporcionan una interfaz de formularios, de esta forma los usuarios deberán rellenar los campos que forman parte del formulario.

#### ***Usuarios sofisticados***

Son usuarios que tienen una mayor habilidad en la interacción con la base de datos, estos están familiarizados con las funciones y recursos del SGBD. Además de esto, son usuarios que acceden a la base de datos desde una terminal, en la que realizan las funciones de consulta y actualización empleando un lenguaje de consultas.

### **2.3.3 Cualidades de un sistema de base de datos**

Con el pasar de los años la manera de almacenar la información ha ido cambiando y ha mejorado por mucho desde sus inicios. La forma más básica de almacenar la información es de manera manual, mientras que versiones más actuales emplean recursos tales como discos, cintas o cualquier otro dispositivo de almacenamiento (sistema de archivos manuales y sistemas de archivos dispersos respectivamente). Un sistema de procesamiento de archivos se vale del uso de una computadora para poder almacenar la información en archivos del sistema operativo. Esta forma de almacenar la información es más eficiente a los métodos antes mencionados, pero no es la más óptima.

Debido a sus múltiples beneficios, las bases de datos es la opción más óptima hoy en día para almacenar la información. Consideramos que una base de datos es mejor respecto a un sistema de procesamiento de archivos o algún otro sistema por los siguientes motivos.

- **Redundancia**

Un dato genera redundancia en el sistema cuando se encuentra almacenado más de una vez en diferentes archivos. Los sistemas de procesamiento de archivos tienen una mayor probabilidad de presentar redundancia. Esto es debido a que, dentro de un largo periodo de tiempo, se van generando diversos archivos y programas de aplicación por diversos usuarios, ocurriendo que algunos de los datos estén dispersos en varios de estos archivos y programas. También puede ocurrir que los archivos presenten diferentes formatos entre ellos, y que los programas de aplicación estén escritos en diferentes lenguajes.

No solo los sistemas de procesamiento de archivos presentan redundancia, sino que también las bases de datos presentan esta situación, pero con la diferencia de que una base de datos puede controlar y aprovechar esta situación para buscar tener un mejor funcionamiento.

No se tiene redundancia cuando cada dato solo es almacenado una sola vez, pero esto no es algo que siempre se pueda hacer ya que por motivos de disponibilidad habrá ciertos datos que deban estar presentes en más de un archivo. La redundancia en este caso se controla haciendo uso de una clave primaria (elemento que se explicará en otra sección).

Los inconvenientes de la redundancia son el consumo de espacio de memoria y el costo de almacenamiento. Estos no representan un mayor problema comparado con la inconsistencia de datos, que es lo que más gravemente llega a afectar a una empresa.

- **Consistencia**

La consistencia de datos es cuando coinciden las múltiples copias que representa la misma información de un dato. La consistencia se obtiene llevando un control de la redundancia, esto es si al actualizar uno de los datos, se actualizan las múltiples copias de ese mismo dato.

Para llevar un control de la redundancia el sistema debe estar al tanto de las múltiples copias, propagando la actualización de los datos cuando se modifica su información. Aunque, si el sistema no está al tanto de esto, una actualización hará que se generen diversas versiones de un mismo dato, lo cual generará inconsistencia de los datos.

La inconsistencia se puede presentar cuando de la base de datos se utilizan algunos de sus datos para generar nuevos datos, y estos nuevos datos se registran en la misma base de datos. Por lo que al actualizar los datos previamente utilizados no se actualizan los datos derivados volviéndolos a generar.

- **Integridad**

La integridad de datos se debe a que los datos cumplen con ciertas reglas o restricciones de integridad, de forma que toda la información almacenada es acertada, garantizando un funcionamiento adecuado en aquella aplicación en la que requiere ser usada.

Para garantizar la integridad en la base de datos se puede agregar al programa de aplicación las restricciones de funcionamiento para validar la información (estas restricciones hacen referencia a los valores de los datos), para que así el programa solo admita aquellos datos que son apropiados para la aplicación cuando el usuario realice las operaciones de inserción o actualización de datos.

Se puede perder la integridad debido a la inconsistencia de los datos (provocada por un mal manejo de la redundancia), o debido a otros factores tales como fallas del programa, fallas por el usuario, realizar transacciones de manera simultánea, entre otras.

- **Seguridad**

Establecer un control de acceso, proporcionar a los usuarios un acceso limitado a los datos y establecer permisos sobre las acciones que cada usuario tiene permitido hacer son los mecanismos de seguridad con los que cuenta una base de datos. En este caso, para poder tener acceso a la base de datos, cada usuario deberá acceder un nombre de usuario y una contraseña lo cual protege a la base de datos de accesos no autorizados. También habrá cierta información que sea sensible, por lo que esta información debe ser confidencial para los usuarios, y solo podrán acceder a aquella información a la que están autorizados. Además, dependiendo de los permisos que se le den al usuario, este podrá realizar algunas operaciones sobre los datos tales como selección, actualización, inserción y borrado.

- **Independencia de datos**

La independencia de datos es la propiedad que permite a los usuarios modificar la estructura de la base de datos sin tener que realizar cambios en las aplicaciones existentes, o tener que efectuar las consultas o actualizaciones de manera distinta. Cabe mencionar que la estructura de la base de datos a su vez presenta independencia física e independencia lógica. Esto es en relación al tipo de elementos de la base de datos a los que se hace referencia.

La independencia física es la propiedad en la que las aplicaciones o los esquemas (conceptual y externo) se mantienen sin cambios cuando se realizan modificaciones en la organización física de la base de datos. Algunos de estos cambios en el nivel físico son modificar o eliminar índices, cambios en las direcciones de almacenamiento, cambios de tamaño de bloques, etc.

La independencia lógica es la propiedad que permite modificar el esquema lógico sin afectar los esquemas externos o las aplicaciones, lo cual proporciona a la base de datos flexibilidad a los cambios. Algunos de estos cambios son incluir/eliminar entidades, atributos o relaciones, realizar cambios de nombre y tipo, etc.

- **Compartida**

Una base de datos compartida es aquella que permite que dos o más usuarios puedan compartir partes individuales de la base de datos, permitiendo a los usuarios acceder de forma simultánea y estar visualizando la misma sección de la base de datos (acceso concurrente). Esto es gracias a un control de concurrencia integrado en el software de la base de datos.

Otro beneficio de este tipo de base de datos es que diferentes aplicaciones pueden compartir los mismos datos. Incluso se puede compartir los datos almacenados con nuevas aplicaciones, sin necesidad de generar nuevos archivos para dichas aplicaciones.

El acceso concurrente no representa un problema cuando los usuarios acceden a la base de datos para visualizar los datos. Los únicos inconvenientes que pueden llegar a presentarse son en cuanto al bajo rendimiento causados por las limitaciones de sus elementos. No obstante, cuando múltiples usuarios acceden a la vez para realizar actualizaciones puede derivar en inconsistencia de los datos, haciendo que se pierda la integridad en la base de datos.

- **Integrada**

Una base de datos integrada hace referencia a que algunas bases de datos son el resultado de la unión de diferentes repositorios donde se almacenan los datos. Un ejemplo de esto es cuando un programa de aplicación ha tenido que realizar la unión de dos o más datos que estaban almacenados de forma separada. Esto para después mostrarlos al usuario de forma conjunta.

### **2.3.4 Clasificación de un sistema de base de datos**

Los sistemas de base de datos tienen una amplia gama de variantes en cuanto a los elementos que lo componen o en cuanto a su funcionamiento. Por lo cual se tendrán diversos criterios a tomar en consideración para realizar su clasificación, derivando en diversas categorías.

La forma en que se almacena la información de manera física en una base de datos es un criterio a tomar en consideración en cuanto a su clasificación. Dentro de esta categoría las bases de datos se dividen en dos grupos: las bases de datos centralizadas y las bases de datos distribuidas.

Adicional a esto, un sistema de base de datos también puede ser clasificado en base al número de usuarios que admite para generar consultas de manera simultánea. Es así que, un sistema que admite un usuario a la vez para realizar la consulta de datos se le denomina sistema monousuario. En cambio, un sistema multiusuario es aquel que permite el acceso a la base de datos a un grupo de usuarios conectándose al sistema a través de terminales.

Tomando esto en cuenta un sistema puede ser catalogado como centralizado o distribuido, y ser catalogado como un sistema monousuario o multiusuario. Un sistema de base de datos centralizado puede estar constituido de más de una computadora, un servidor de base de datos, y de un solo elemento de almacenamiento que conformará la base de datos.

Un sistema distribuido es un sistema en el que la base de datos está distribuida en diferentes áreas geográficas. Por ejemplo, en el caso de una organización que está compuesta de diferentes sedes o departamentos, esta puede almacenar múltiples copias de una sección de la base de datos en algunas de sus sedes, de esta forma la organización puede disponer en la medida de lo posible de dicha información sin que se vea afectada ante el fallo de alguno de los sistemas.

Además, un sistema distribuido de base de datos, a pesar de ser un sistema más complejo, tiene otras características que lo vuelven más eficiente para una organización respecto de otros sistemas. Entre estas características tenemos las que se presentan a continuación.

- **Datos compartidos**

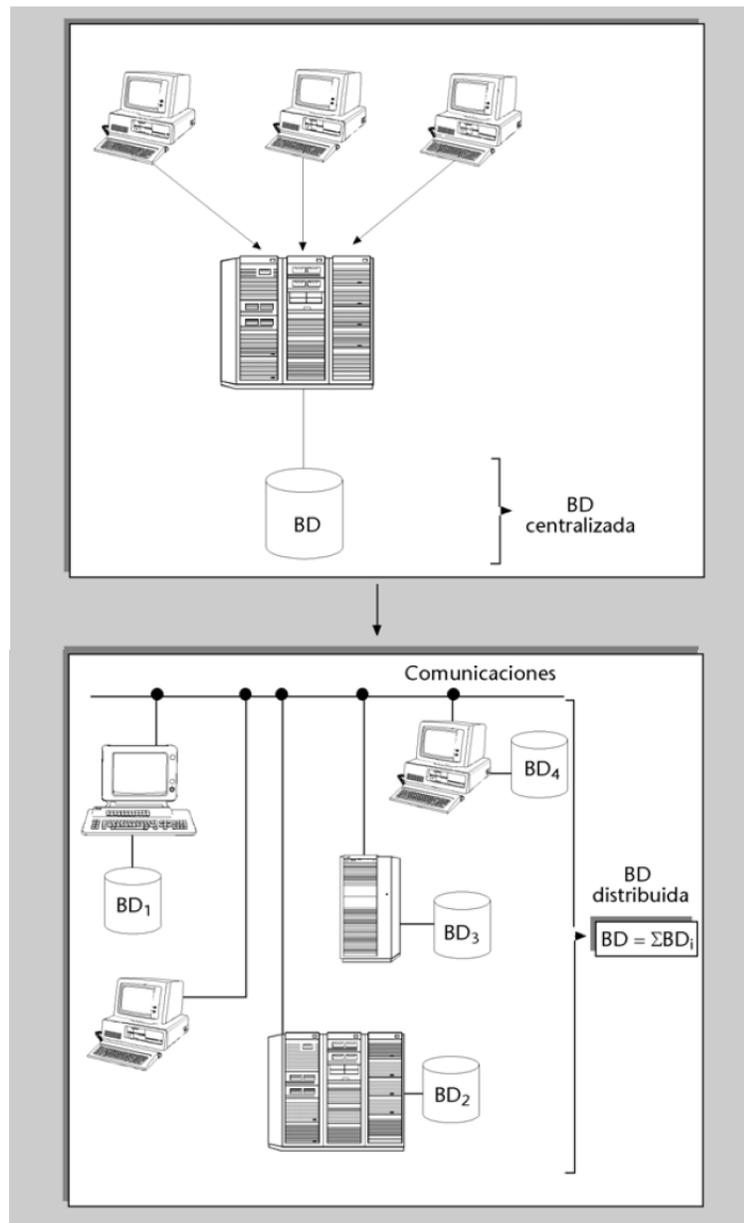
Los usuarios de un sistema distribuido pueden acceder a los datos que estén almacenados en otras localidades, ya que cada sistema puede contener datos que solamente se encuentran almacenados en esa base de datos.

- **Autonomía**

En un sistema distribuido se asigna un administrador local por cada sistema que forme parte de la organización, por lo que cada sistema se puede gestionar de manera autónoma. Sin embargo, el que tiene el control de todo el sistema es el administrador global de la base de datos, por lo

que la autonomía de cada sistema dependerá de la responsabilidad que se le delegue a su administrador local.

La representación de los sistemas de base de datos centralizados y distribuidos se muestra en la Figura 2.4.



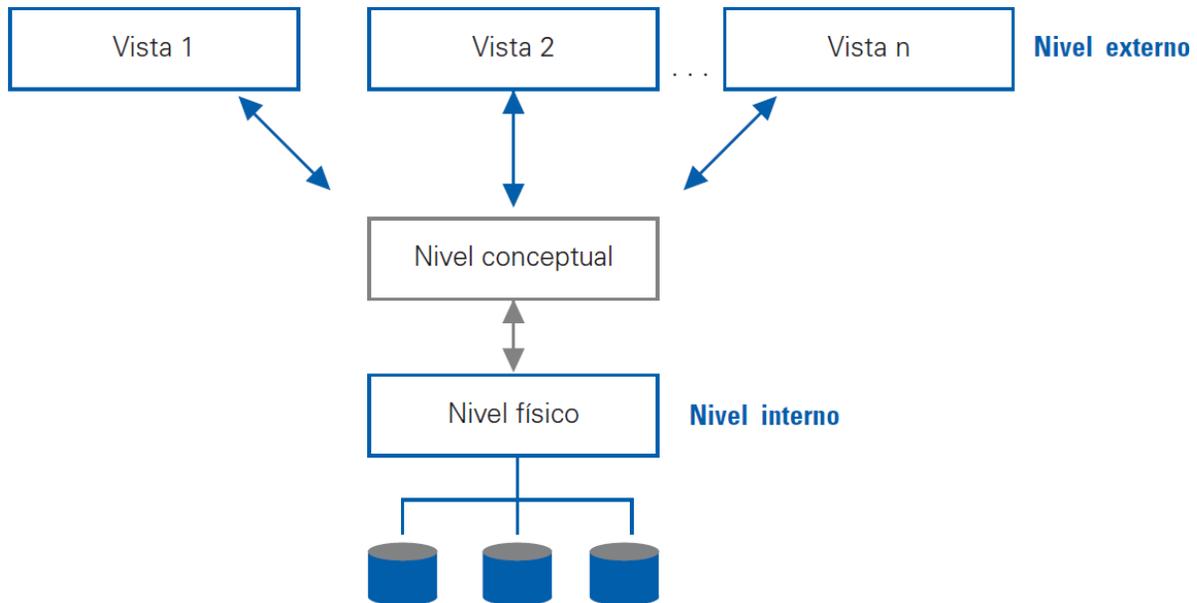
**Figura 2.10** Organización de un sistema de base de datos.

Fuente: Bases de datos <sup>13</sup>

<sup>13</sup> Bases de datos (p. 21), por Camps, R. 2005, UOC.

### 2.3.5 Arquitectura de un sistema de base de datos

Un sistema de base de datos se representa por medio de una estructura, la cual se divide en tres niveles: interno, conceptual y externo. Cada nivel es una abstracción de la base de datos y dependiendo del nivel es la descripción que se hace de la base de datos.



**Figura 2.11** *Arquitectura de tres niveles de abstracción de datos.*

Fuente: DISEÑO DE BASE DE DATOS <sup>14</sup>

- **Nivel interno**

En el nivel interno o físico se describe el almacenamiento físico de los datos haciendo referencia a las unidades de almacenamiento. Es decir, como están distribuidos los datos de manera física en la estructura de almacenamiento. Este nivel es el más bajo de abstracción.

- **Nivel conceptual**

El siguiente nivel es el conceptual o nivel lógico. En este nivel se encuentra la vista conceptual, la cual muestra una relación entre el nivel físico y el nivel externo. Una base de datos cuenta solamente con una vista conceptual, en la cual se visualiza la base de datos completa, esto es todos sus datos y la relación que hay entre esos datos. En el nivel conceptual se detalla información en relación a los datos y el funcionamiento de la base de datos. Se define los aspectos de entidades, atributos y sus relaciones, aspectos de integridad y seguridad de los datos.

<sup>14</sup> DISEÑO DE BASE DE DATOS (p. 26), por Merchán, O. 2018, Universidad del Azuay.

- **Nivel externo**

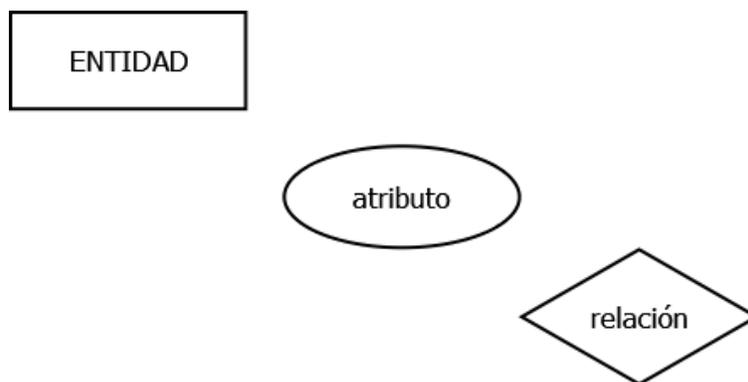
El nivel externo, también llamado nivel de vistas. Es el nivel más alto de abstracción y representa solo una parte de la base de datos completa. En este nivel se encuentran las vistas externas, las cuales son un medio para mostrar al usuario la información almacenada en el sistema. El usuario solo se interesará en cierta información en particular, y no en toda la información de la base de datos. Por lo cual siempre se cuenta con múltiples vistas, cada una para un usuario en particular. Por otra parte, las vistas ayudan a proporcionar solamente la información relevante para el usuario, omitiendo los detalles del nivel lógico. O el dar una mayor protección a los datos, evitando que los usuarios irrumpían en alguna otra sección de la base de datos.

Al diseño de la base de datos se le denomina esquema y cada nivel cuenta con uno. Se cuenta con un esquema interno para el nivel interno. Un esquema conceptual para el nivel conceptual. Y en el caso del nivel externo, esta cuenta con múltiples esquemas los cuales se denominan sub esquemas, que definen las diversas vistas externas con las que se cuenta de la base de datos.

### 2.3.6 Modelos de datos

- **Modelo entidad relación**

El modelo ER está conformado principalmente por entidades, atributos y relaciones que en conjunto forman una representación diagramática de la de la base de datos. Lo que se representa es su estructura y sus restricciones de integridad (derivadas de las reglas del mundo real). Este modelo es el que más se utiliza en el diseño conceptual de la base de datos.



**Figura 2.12** Elementos básicos de un modelo entidad relación.

Fuente: Elaboración propia <sup>15</sup>

---

<sup>15</sup> Simbología o nomenclatura básica de un modelo de datos según notación Chen.

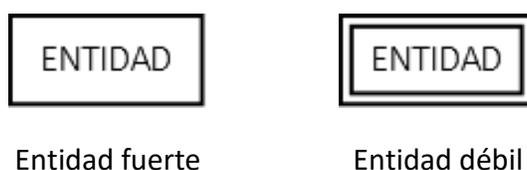
La simbología de los elementos que conforman el esquema conceptual de la base de datos son los que se muestran en la Figura 2.12. Dicho esto, a continuación, se presenta una descripción de cada uno de estos elementos, con el objetivo de dar una idea clara de cada uno de ellos ya que juegan un papel muy importante en el diseño de la base de datos.

### ***Entidades***

Las entidades representan aquellos objetos o conceptos del mundo real que son de especial interés respecto de otros objetos. Una entidad se representa de forma gráfica mediante un rectángulo con el nombre del objeto escrito en mayúsculas en su interior, y el nombre suele expresarse mediante un sustantivo en singular. Dentro del modelo, el nombre de cada entidad debe estar escrito una sola vez.

Se utiliza el nombre de la entidad para expresar que se trata de un conjunto de objetos similares, haciendo referencia a ellos de forma general, a lo cual se le denomina tipos o clases de entidades. A su vez, las entidades al ser la unión de varios objetos, la representación unívoca de cada uno de estos recibe el nombre de instancia u ocurrencia de una entidad.

Dos entidades pueden estar relacionadas de forma que una de ellas necesita de otra para que pueda existir. En este caso las entidades se dividen en:



**Figura 2.13** Clasificación de las entidades.

**Fuente:** Elaboración propia <sup>16</sup>

### ***Entidad fuerte***

Una entidad fuerte o dominante es aquella que tiene existencia propia y posee un atributo clave.

### ***Entidad débil***

Una entidad débil o subordinada es aquella que necesita estar relacionada con una entidad fuerte para poder existir. Además, una entidad débil se caracteriza por carecer de un atributo clave. En su lugar se utiliza algún atributo que tenga asociado para formar una clave parcial o discriminante.

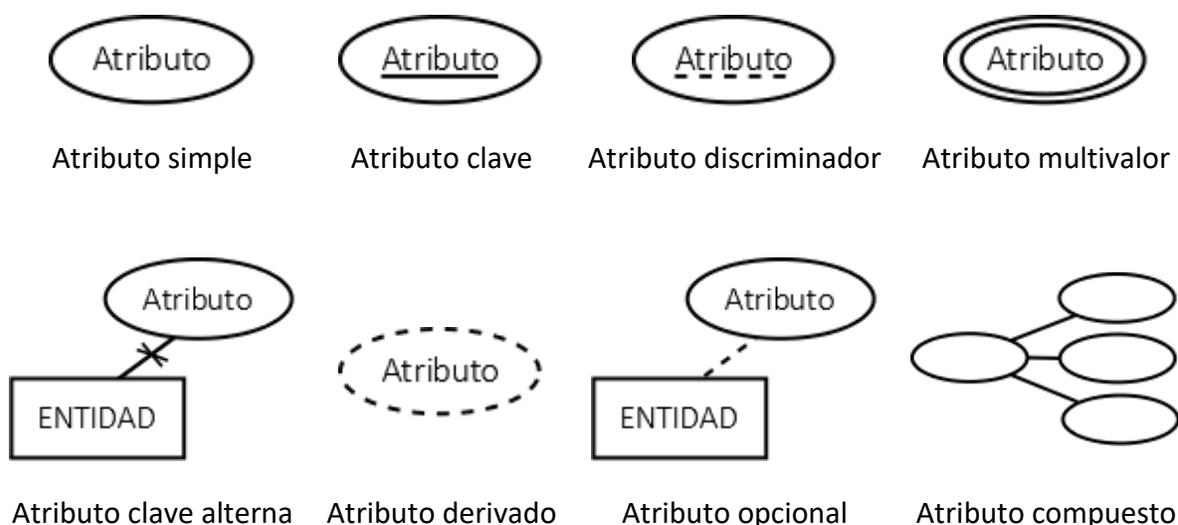
---

<sup>16</sup> Figura ejemplificativa sobre los tipos de entidades y su representación en el diagrama ER.

## Atributos

Los atributos son propiedades o características que sirven para realizar una descripción de las entidades (adquiriendo diferentes valores para cada instancia de la entidad). Para lo cual se consideran únicamente aquellas propiedades que son significativas de la entidad. Los atributos se representan de forma gráfica mediante el nombre de la propiedad en minúsculas contenida dentro de un óvalo. Un atributo se encuentra asociado a una entidad cuando están unidos por una línea, de igual forma las relaciones pueden tener atributos asociados a estas.

Cada atributo consta de un dominio o conjunto de valores, indicando así que el atributo puede tomar los valores que estén dentro del conjunto. Tomando en consideración la función que desempeña cada atributo, estos se dividen en diferentes tipos, los cuales se muestran a continuación.



**Figura 2.14** Clasificación de los atributos.

Fuente: Elaboración propia <sup>17</sup>

### *Atributo simple*

Un atributo simple es aquel cuyos valores no pueden ser divisibles en partes más pequeñas (valor atómico). Además de esto, sus valores tienen una relación unívoca con cada una de las instancias de la entidad con la que está asociada.

<sup>17</sup> Figura ejemplificativa sobre los tipos de atributos y su representación en el diagrama ER.

### ***Atributo clave***

Un atributo clave es aquel en el que cada valor es único, de forma que sus valores permiten identificar de manera unívoca a cada instancia de la entidad a la que está asociada. Se considera clave primaria cuando el diseñador decide utilizar dicho atributo para identificar las instancias de la entidad, en cuyo caso es importante que el atributo tenga valores que no sean variables.

### ***Atributo discriminador***

Un atributo discriminador es aquel que está asociado a una entidad débil. Además, tiene una clave denominada clave parcial, débil o discriminante que puede formar parte de una clave primaria.

### ***Atributo multivalor***

Un atributo multivalor es aquel que relaciona un conjunto de valores con cada una de las instancias de la entidad con la que está asociada. Además, se puede asignar un límite inferior y superior, de modo que no es obligatorio que el número de valores que tome alguna de las instancias deba ser el mismo para el resto. Incluso se puede considerar que el límite inferior de atributos sea de cero para la entidad asociada.

### ***Atributo clave alterna***

Un atributo que puede ser utilizado como clave primaria sin serlo se le denomina clave alterna. Además de esto, para considerar que un atributo es una clave alterna es importante tener la garantía de que sus valores siempre serán únicos en todo momento.

### ***Atributo derivado***

Un atributo derivado es aquel que contiene valores que pueden ser calculados u obtenidos de otros atributos, para lo cual se puede recurrir a uno o a varios atributos. Al poder obtener dichos valores recurriendo a otros atributos, estos no necesitan ser almacenados, sino que son calculados cuando es necesario.

### ***Atributo opcional***

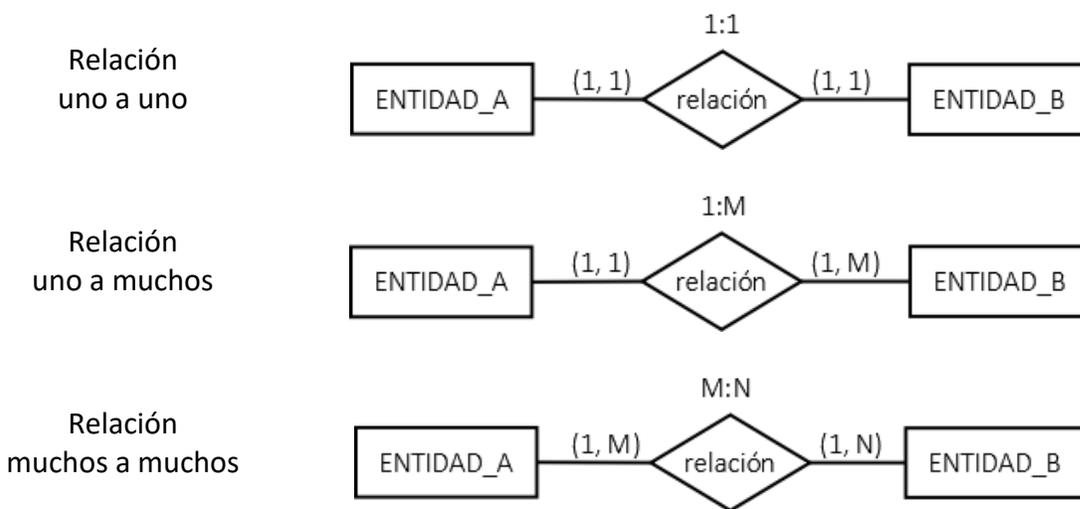
Un atributo opcional se utiliza cuando los atributos de una entidad requieren de valores nulos (esto es solamente para algunas instancias de la entidad). El valor nulo representa ausencia de valor y se utiliza cuando el valor es desconocido o no aplicable.

### ***Atributo compuesto***

Un atributo compuesto es aquel que se compone de otros atributos.

## Relaciones

Una relación es una asociación entre entidades. Las relaciones se representan de forma gráfica mediante el nombre de la relación contenida dentro de un rombo. En el caso de las relaciones, su nombre se representa con algún verbo o alguna frase corta que incluya un verbo, y la forma en que se realiza el vínculo es mediante líneas. Cada línea deberá estar situada entre la entidad asociada y la relación que la asocia.



**Figura 2.15** Cardinalidad de las relaciones.

Fuente: Elaboración propia <sup>18</sup>

Una relación se caracteriza en base al número de entidades que participan en la relación (grado), y en cuanto al número de instancias con las que la instancia de una entidad guarda relación con otra entidad (Cardinalidad).

Considerando una relación entre dos entidades (relación binaria), el tipo de correspondencia que relaciona a ambas entidades puede ser alguna de las que se muestran en la Figura 2.15, las cuales se describen a continuación.

### *Uno a uno*

La relación uno a uno se representa mediante la expresión 1:1, lo cual quiere decir que una instancia de la entidad A se asocia máximo con una instancia de la entidad B y viceversa.

### *Uno a muchos*

La relación uno a muchos se representa mediante la expresión 1:M, lo cual especifica el máximo de relaciones que pueden tener las instancias de cada entidad, pero no especifica el mínimo, pudiendo ser el mínimo uno o cero. También se puede utilizar la expresión (1, M)

<sup>18</sup> Figura ejemplificativa sobre los tipos de cardinalidad de una relación y su representación en el diagrama ER.

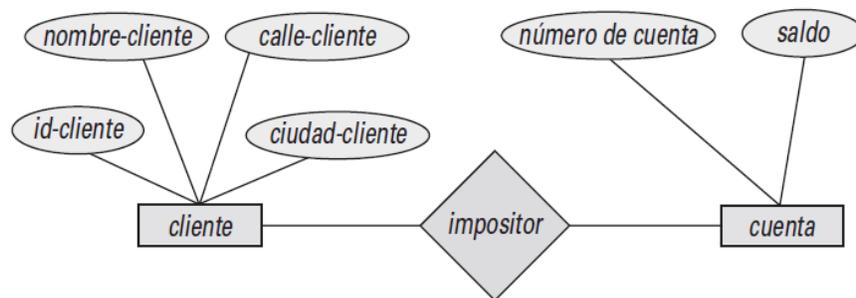
en el lado de la entidad B para indicar que una instancia de la entidad A puede estar asociada con mínimo una instancia de la entidad B y máximo con muchas. Y la expresión (1, 1) en el lado de la entidad A para indicar que una instancia de la entidad B mínimo puede estar asociada con una instancia de la entidad A y máximo con una.

Sin embargo, se puede remplazar la letra M con un número que sea mayor a uno para expresar que la relación es uno a varios o las letras M y N para representar una relación de varios a varios, siendo el máximo el número designado en la cardinalidad de la relación.

### ***Muchos a muchos***

La relación muchos a muchos se representan mediante la expresión M: N, lo cual quiere decir que una instancia de la entidad A se asocia máximo con muchas instancias de la entidad B y viceversa.

El diagrama mostrado en la Figura 2.16 es un ejemplo de un modelo entidad relación, en el cual se muestra un segmento del diagrama entidad relación de una base de datos de un sistema bancario. Dicho segmento consiste en registrar información personal sobre los clientes del banco, así como de sus cuentas.



**Figura 2.16** Ejemplo de diagrama E-R.  
Fuente: FUNDAMENTOS DE BASES DE DATOS <sup>19</sup>

### ▪ **Modelo relacional**

El modelo relacional es una representación en la cual se muestra la base de datos como una estructura lógica, de forma que los datos y sus relaciones son representados mediante un conjunto de tablas.

En el proceso de diseño de una base de datos se requiere realizar ambos modelos (el modelo ER y el modelo relacional). Para lo cual primero se debe realizar el diagrama entidad relación para que de este se pueda obtener las tablas que conforman el esquema relacional de la base de datos.

<sup>19</sup> FUNDAMENTOS DE BASES DE DATOS (p. 6), por Silberschats, A., Korth, H. F. y Sudarshan, S., 2002, McGRAW-HILL/INTERAMERICANA DE ESPAÑA.

Cabe mencionar que hay una terminología para hacer referencia a las partes que integran un esquema del modelo relacional. Es así que para referirse a una tabla se puede utilizar el término relación o archivo. Una tabla a su vez se compone de filas y columnas, de forma que para hacer referencia a estos se utilizan los términos tupla o registro (en el caso de las filas), y campos o atributos (en el caso de las columnas). La siguiente figura muestra el modelo relacional del diagrama ER del ejemplo anterior.

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsaín
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

(a) La tabla *cliente*

<i>número-cuenta</i>	<i>saldo</i>	<i>id-cliente</i>	<i>número-cuenta</i>
C-101	500	19.283.746	C-101
C-215	700	19.283.746	C-201
C-102	400	01.928.374	C-215
C-305	350	67.789.901	C-102
C-201	900	18.273.609	C-305
C-217	750	32.112.312	C-217
C-222	700	33.666.999	C-222
		01.928.374	C-201

(b) La tabla *cuenta*

(b) La tabla *impositor*

**Figura 2.17** Ejemplo de base de datos relacional.

Fuente: FUNDAMENTOS DE BASES DE DATOS <sup>20</sup>

En términos generales, para realizar la transición del modelo ER al modelo relacional hay que tomar en cuenta algunos aspectos, tales como:

### **Entidades**

En el modelo relacional cada entidad se representa por medio de una relación (tabla), a cada una de las cuales se les asigna un nombre.

### **Atributos**

Los atributos de una entidad se representan en una relación mediante campos (columnas), especificando el nombre del atributo en la parte de la cabecera.

Por tanto, una relación es una colección de tuplas, donde todas estas comparten los mismos atributos, y para cada tupla se tiene un valor registrado en cada uno de sus campos. Las tuplas

<sup>20</sup> FUNDAMENTOS DE BASES DE DATOS (p. 6), por Silberschats, A., Korth, H. F. y Sudarshan, S., 2002, McGRAW-HILL/INTERAMERICANA DE ESPAÑA.

corresponden a objetos o conceptos del mundo real, de los cuales es importante registrar información.

Una clave primaria en una relación se representa como un atributo o un conjunto de atributos, donde los valores asignados a una tupla como clave primaria deben ser únicos, y deben ser valores válidos (no nulos), lo cual permite que cada tupla pueda ser identificada de manera unívoca.

En un diagrama relacional, para establecer una relación entre relaciones se utilizan la clave primaria de alguna de las dos relaciones o de ambas. Por tanto, cuando una relación entre sus atributos cuenta con la clave primaria de otra relación, las tuplas de ambas relaciones presentan algún tipo de correspondencia entre sí.

Cabe mencionar que cuando el atributo designado como clave primaria de una relación está presente en otra relación para establecer una relación, esto da lugar a la formación de una clave en la otra relación denominada clave foránea o externa. La relación a la que pertenece la clave primaria recibe el nombre de relación referenciada. Mientras que la relación a la que pertenece la clave foránea recibe el nombre de relación referenciante.

### ***Relaciones***

Una relación entre dos entidades requiere ser expresada en el modelo relacional en base a su cardinalidad.

Una relación 1:1 se representa añadiendo una clave foránea en cualquiera de las dos relaciones para hacer referencia a la otra relación.

Una relación 1: N se representa añadiendo una clave foránea a la relación que corresponde a la entidad del lado N. Esto es para que cada una de las tuplas de dicha relación tenga asignada una tupla de la otra relación.

Una relación M: N se representa mediante otra relación (tabla), la cual tendrá como clave primaria la unión de las claves primarias de las entidades relacionadas.

#### ▪ **Modelo jerárquico**

Una base de datos jerárquica para almacenar la información emplea una estructura jerárquica. Esta estructura es similar a la de un árbol, en la cual se hace referencia a que los datos deben ser almacenados en base a un orden de importancia.

Para implementar este modelo, la estructura se construye por medio de segmentos, los cuales pueden ser nodos o ramas. A su vez, dichos segmentos se clasifican en tres grupos, tal como se muestra a continuación.

### ***Nodo padre***

Es un nodo del cual surgen otros nodos. Los nodos padres están situados en el mismo nivel lo cual significa que todos tienen el mismo nivel de importancia.

### ***Nodo hijo***

Este nodo es una derivación del nodo padre, por lo cual el nodo hijo depende del nodo padre.

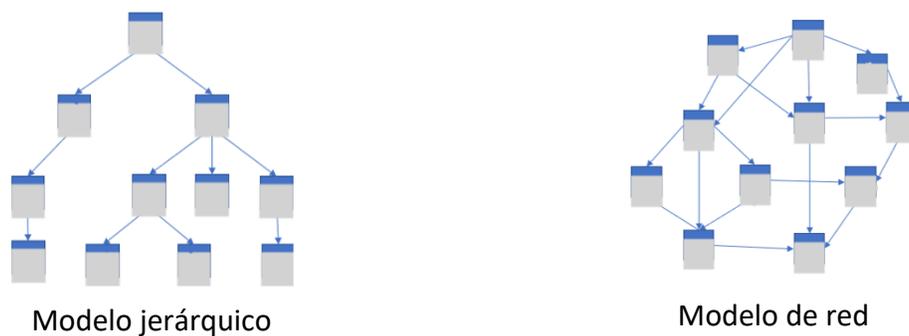
### ***Raíz***

Se encuentra en el nivel superior, por lo que no tiene un nodo padre. Es el origen de los datos a partir del cual se desprenden los demás nodos.

## ▪ **Modelo de red**

Una base de datos de red al igual que las bases de datos jerárquicas también está compuesta por nodos, aunque la composición de los nodos la diferencia de la estructura jerárquica, ya que los nodos pueden tener diversos padres.

La siguiente figura muestra un ejemplo de los esquemas de las bases de datos jerárquicas y las bases de datos de red.



**Figura 2.18** Ejemplo de una base de datos jerárquica y de red.

Fuente: Elaboración propia <sup>21</sup>

### **2.3.7 Normalización**

La normalización es un proceso en el cual se determina si en el modelo relacional de una base de datos se tienen relaciones cuyas estructuras resultan ser poco eficaz o inadecuadas.

Este proceso da como resultado obtener un modelo que sea menos sensible a las anomalías de actualización. Dichas anomalías se presentan cuando se modifican datos en un modelo que ha sido diseñado de forma inadecuada.

Para realizar este proceso se tiene que seguir una secuencia de pasos en el que cada paso del proceso corresponde a una forma normal, siendo la primera forma normal (1FN), la segunda forma

---

<sup>21</sup> Figura ejemplificativa sobre los esquemas de los tipos de base de datos jerárquica y de red.

normal (2FN), y la tercera forma normal (3FN). Siendo estas las formas normales más comúnmente utilizadas.

Entre mayor sea la forma normal de una relación, su estructura es más restringida y menos sensible a las anomalías.

A continuación, se presenta el proceso de normalización mediante las tres primeras formas normales de una relación.

- **Primera forma normal**

Toda relación está en primera forma normal cuando los valores de sus atributos son atómicos. Es decir, solamente puede registrarse un valor en cada una de las celdas que conforman a un atributo.

ESTUDIANTE		ESTUDIANTE	
Ci	Carrera	Ci	Carrera
0101010101	Contabilidad	0101010101	Contabilidad
0123451234, 0133556678	Economía	0123451234	Economía
0112222222, 0133388809, 0198765432	Informática	1112222222	Informática
		0133388809	Informática
		0133556678	Economía
		0198765432	Informática

**Figura 2.19** (a) Relación no normalizada. (b) Relación en primera forma normal.

Fuente: DISEÑO DE BASE DE DATOS <sup>22</sup>

- **Segunda forma normal**

Toda relación que está en primera forma normal pasa a estar en segunda forma normal si se cumple que todos sus atributos que no son clave primaria dependen funcionalmente de manera completa respecto de la clave principal.

La dependencia funcional se presenta si por ejemplo un atributo *X* depende funcionalmente del atributo *Y*. Esto porque es posible determinar los valores de *X* a partir de los valores de *Y*. Así pues, se dice que *Y* determina a *X* lo cual se expresa como  $Y \rightarrow X$ , para lo cual en todo momento *Y* debe estar asociado con un único valor de *X* (*X* y *Y* pueden estar formados por uno o más atributos).

<sup>22</sup> DISEÑO DE BASE DE DATOS (p. 294), por Merchán, O. 2018, Universidad del Azuay.

## ARTÍCULO

Fabricante	Descripción	Nro_Casilla	Bodega	Estantería
Concretos S.A.	Cemento	8765	1	1
Concretos S.A.	Cementina	8765	1	1
JMHierro	Malla	2659	1	2
JMHierro	Hierro	2659	1	2
JMHierro	Alambre	2659	1	2
Maderesa	Madera cerezo	4170	2	3
Maderesa	Madera roble	4170	2	3
Granito	Cerámica	1234	2	4
Granito	Porcelanato	1234	2	4
Grupo TPVC	Tubería PVC	9876	3	5
Accelec	Accesorios eléctricos	1357	3	5
PlastiMex	Tubería PVC	2468	3	6
Aluminio S.A.	Perfil de aluminio	9483	2	3

**Figura 2.20** Instancia de la tabla ARTÍCULO.

Fuente: DISEÑO DE BASE DE DATOS <sup>23</sup>

En la figura 2.20 se muestra el ejemplo de una relación de nombre ARTÍCULO, la cual presenta ciertos inconvenientes. A continuación, se hace mención de estos inconvenientes con la finalidad de normalizar la relación para que pase a estar en segunda forma normal.

- Los atributos de la relación corresponden a dos temas diferentes. Esto porque los atributos proporcionan información relacionada con el fabricante, pero también proporcionan una descripción de los productos de cada fabricante.
- Ni uno de los atributos de la relación se representa como clave primaria.

A continuación, se muestra los pasos a seguir con los cuales la relación ARTÍCULO puede ser representada en segunda forma normal.

- Se sustituye la relación ARTÍCULO con dos nuevas tablas. Una denominada ARTÍCULO1 para almacenar información de los fabricantes y sus artículos, y otra FABRICANTE1 para almacenar exclusivamente información relacionada a los fabricantes.

<sup>23</sup> DISEÑO DE BASE DE DATOS (p. 297), por Merchán, O. 2018, Universidad del Azuay.

- Se designa como clave primaria el atributo Fabricante.

La figura 2.21 y 2.22 muestra las tablas ARTÍCULO1 y FABRICANTE1, las cuales representan la segunda forma normal de la relación ARTÍCULO.

Fabricante	Descripción
Concretos S.A.	Cemento
Concretos S.A.	Cementina
JMHierro	Malla
JMHierro	Hierro
JMHierro	Alambre
Maderesa	Madera cerezo
Maderesa	Madera roble
Granito	Cerámica
Granito	Porcelanato
Grupo TPVC	Tubería PVC
Accelec	Accesorios eléctricos
PlastiMex	Tubería PVC
Aluminio S.A.	Perfil de aluminio

**Figura 2.21** Instancia de la tabla ARTÍCULO1.

Fabricante	Descripción	Bodega	Estantería
Concretos S.A.	8765	1	1
JMHierro	2659	1	2
Maderesa	4170	2	3
Granito	1234	2	4
Grupo TPVC	9876	3	5
Accelec	1357	3	5
PlastiMex	2468	3	6
Aluminio S.A.	9483	2	3

**Figura 2.22** Instancia de la tabla FABRICANTE1.

Fuente: DISEÑO DE BASE DE DATOS <sup>24</sup>

### ▪ Tercera forma normal

Toda relación que está en segunda forma normal pasa a estar en tercera forma normal si esta no tiene dependencias transitivas.

La dependencia transitiva se presenta si por ejemplo un atributo Z tiene dependencia transitiva respecto de X a través de Y. Esto porque Z depende funcionalmente de Y, pero a su vez Y depende funcionalmente de X. Así pues, las dependencias funcionales entre X y Y, y entre Y y Z son  $X \rightarrow Y$ , y  $Y \rightarrow Z$  de manera respectiva.

La tercera forma normal elimina la dependencia transitiva para evitar que una relación con este tipo de dependencia presente anomalías de actualización. Así, por ejemplo, en el caso de la tabla

<sup>24</sup> DISEÑO DE BASE DE DATOS (p. 303), por Merchán, O. 2018, Universidad del Azuay.

FABRICANTE1 (figura 2.22), esta presenta una dependencia transitiva. Esto porque los atributos Fabricante y Estantería, y los atributos Estantería y Bodega presentan dependencia funcional (Fabricante → Estantería y Estantería → Bodega respectivamente), ya que cada fabricante tiene asignado una estantería y una estantería pertenece a una sola bodega.

A continuación, se muestra los pasos a seguir con los cuales la tabla FABRICANTE1 puede ser representada en tercera forma normal.

- Se sustituye la tabla FABRICANTE1 con dos nuevas tablas. Una denominada FABRICANTE2 para almacenar información de los atributos Fabricante, Num\_casilla y Estantería, y otra denominada LOCAL para almacenar información de los atributos Estantería y Bodega.

La figura 2.23 muestra las tablas FABRICANTE2 y LOCAL, las cuales representan la tercera forma normal de la relación FABRICANTE1.

FABRICANTE2			LOCAL	
Fabricante	Nro_Casilla	Estantería	Estantería	Bodega
Concretos S.A.	8765	1	1	1
JMHierro	2659	2	2	1
Maderesa	4170	3	3	2
Granito	1234	4	4	2
Grupo TPVC	9876	5	5	3
Accelec	1357	5	6	3
PlastiMex	2468	6		
Aluminio S.A.	9483	3		

**Figura 2.23** Tablas en tercera forma normal.

Fuente: DISEÑO DE BASE DE DATOS <sup>25</sup>

## 2.4 MANEJADORES DE BASES DE DATOS

### 2.4.1 Manejadores de bases de datos (SQL y NoSQL)

En un sistema de base de datos, un manejador de base de datos (SGBD) es el motor de búsqueda con el que los usuarios acceden a los elementos que alberga una base de datos. Es así que con el aumento de las tecnologías de la información y la dependencia que se ha generado por estas, han surgido en los últimos años diferentes proveedores de manejadores de bases de datos. Algunos de

<sup>25</sup> DISEÑO DE BASE DE DATOS (p. 308), por Merchán, O. 2018, Universidad del Azuay.

estos están disponibles en internet sin costo alguno. Mientras que otros están disponibles en el mercado bajo un costo.

En tanto que la mayor parte de manejadores de bases de datos disponibles son relacionales, en ciertas aplicaciones se ha optado por utilizar manejadores de bases de datos no relacionales, resultando ser más eficientes en ciertas ocasiones. Por tal motivo, se debe de prestar atención sobre la función que tiene cada uno de estos manejadores, y así poder identificar sus aplicaciones. A continuación, se da una descripción sobre estos tipos de base de datos y sus aplicaciones.

#### ▪ **Bases de datos relacionales**

Se define como base de datos relacional a aquellas bases de datos que emplean el modelo relacional como forma de almacenamiento de sus datos. Por lo cual, la forma en que este tipo de base de datos presenta los datos es en forma de tablas, cada una de las cuales se relacionan entre sí.

Adicional a esto, una base de datos relacional para poder almacenar, administrar y recuperar los datos almacenados utiliza un conjunto de programas. A este conjunto de programas se le denomina como sistema de gestión de base de datos relacional (RDBMS).

A continuación, se presentan las cualidades de las bases de datos relacionales.

- El uso de claves en una relación permite acceder a su información de forma sencilla y en todo momento.
- La sencillez con la cual se muestra la información a los usuarios, permitiendo organizar grandes cantidades de información por medio de tablas, las cuales pueden estar relacionadas entre sí.
- Mantiene un control sobre la duplicidad de los registros y proporciona integridad referencial. Es decir, siempre que se elimine un registro, la integridad de los demás registros no se verá afectada.

A pesar de las cualidades de las bases de datos relacionales, estas se encuentran limitadas en el manejo de datos gráficos, multimedia, CAD y sistemas de información geográfica.

#### ▪ **Bases de datos no relacionales**

Se le denomina como base de datos no relacional a los sistemas de almacenamiento que no cumplen con el esquema entidad relación, ni tampoco emplean tablas para almacenar la información. Las bases de datos no relacionales para almacenar la información, en su lugar hacen uso de otro tipo de modelos tales como clave-valor, mapeo de columnas o grafos.

A las bases de datos no relacionales también se les llama NoSQL (Not only SQL). Esto porque este tipo de bases de datos no usan el lenguaje SQL. Si acaso lo utilizan de apoyo, pero no para realizar las consultas. Para realizar las consultas, las bases de datos no relacionales emplean lenguajes tales como JSON, CQL o GQL.

### ***Bases de datos clave-valor***

El modelo clave-valor es de los más populares entre las bases de datos no relacionales. En este modelo cada elemento está identificado por una llave única, lo cual agiliza la recuperación de la información.

### ***Bases de datos documentales***

Las bases de datos documentales almacenan la información en forma de documentos. Para lo cual utilizan estructuras simples como JSON o XML. Además, utiliza una clave única para cada registro.

### ***Bases de datos en grafo***

Las bases de datos de este tipo utilizan los nodos de los grafos para almacenar la información. Mientras que las relaciones las representa como aristas de los mismos. En este caso se puede utilizar la teoría de grafos para recorrerla.

### ***Bases de datos orientadas a objetos***

Las bases de datos en este caso representan la información en forma de objetos, tal como se hace con los lenguajes de programación orientada a objetos (POO), como es el caso de los lenguajes de programación Java, C# y Visual Basic.NET.

A continuación, se presentan las cualidades por las cuales las bases de datos no relacionales han resultado ser más funcionales que las bases de datos relacionales. Esto en base al tipo de aplicación en que se emplean.

- Cuando se llegan a generar grandes cantidades de información de forma muy rápida en momentos puntuales. Incluso llegando a superar el Terabyte de información.
- Cuando al utilizar una base de datos relacional la escalabilidad deja de ser viable en el nivel de costos y a nivel técnico.
- Cuando varias veces se presentan elevados picos en el uso del sistema por parte de los usuarios.
- Cuando se requiere de un esquema no homogéneo. Esto es cuando la información que se almacena puede tener campos distintos.

## 2.4.2 Principales manejadores de bases de datos



Figura 2.24 Manejadores de bases de datos.

Fuente: Elaboración propia <sup>26</sup>

### ▪ Manejadores de bases de datos de paga

Los manejadores de bases de datos que son de paga están orientados para uso empresarial, los cuales brindan una mayor calidad en almacenamiento, procesamiento y seguridad de la información. Los manejadores de bases de datos disponibles en el mercado, por mencionar algunos, tenemos los siguientes:

#### *Oracle Database*

Se trata de un manejador de base de datos relacional, el cual cuenta con diferentes ediciones, cada una con diferentes cualidades para adaptarse de acuerdo a cada necesidad. Incluso cuenta con una versión gratuita, aunque limitada, y una versión para dispositivos móviles.

#### *Microsoft SQLServer*

Microsoft SQL forma parte de aquellos manejadores de bases de datos que predominan en el mercado, compitiendo con Oracle DB y otros manejadores de alto nivel. Este cuenta con un motor para el almacenamiento, procesamiento y seguridad de los datos. Otro motor para procesar comandos y consultas, y otro más para gestionar archivos, tablas, páginas e índices.

#### *Microsoft Access*

Microsoft Access es un programa utilizado para gestionar bases de datos en equipos con sistema Windows, en aplicaciones de pequeña escala. Su importancia radica en su entorno gráfico, el cual es fácil de utilizar, y permite la creación de tablas, formularios e informes.

<sup>26</sup> URL de los logos: <https://www.pinterest.com.mx/>

### *Informix*

Informix surgió como una compañía en 1980 dedicada al desarrollo de productos relacionados al manejo de bases de datos. Después de algunos problemas en su gestión fue adquirida por IBM en 2001. Pero debido a sus cualidades la empresa Walmart actualmente emplea Informix para gestionar información sobre sus recursos.

### *Sybase*

Sybase es un manejador de bases de datos desarrollado por una compañía del mismo nombre. También se le denomina Sybase ASE (Adaptive Server Enterprise). Este se caracteriza por ser el primer sistema de bases de datos de nivel empresarial, diseñado para administrar grandes cantidades de datos y miles de usuarios de forma simultánea.

#### ▪ **Manejadores de bases de datos gratuitos**

A pesar de que los manejadores de bases de datos comerciales garantizan una mayor calidad en sus productos. Existen ciertas aplicaciones en las que suele ser más óptimo el uso de manejadores de bases de datos de código abierto que puedan ser descargados de forma gratuita. Esto porque además de ser gratuitos ofrecen cualidades equiparables a los manejadores de bases de datos comerciales. A continuación, se mencionan algunos de los manejadores de bases de datos de código abierto más populares.

### *PostgreSQL*

PostgreSQL forma parte de los manejadores de bases de datos de código abierto, siendo uno de los mejores calificados. Sin embargo, uno de sus inconvenientes es que puede mostrarse lento en el manejo de bases de datos de pequeño y mediano tamaño.

### *MySQL*

MySQL es un manejador mayormente utilizado en entornos que requieren ser modificados ocasionalmente. Un ejemplo de esto es en el desarrollo de aplicaciones web. Esto es debido a que los usuarios utilizan dichas aplicaciones para lectura de datos.

### *MaríaDB*

MaríaDB es una versión mejorada de MySQL para hacer frente a algunas limitaciones que este presenta. Además, MaríaDB busca compensar la adquisición de MySQL por parte de Oracle, lo cual representa un problema debido a que MySQL ya no es del todo Open Source.

### 2.4.3 Comparativa entre manejadores de bases de datos

La tabla 2.2 muestra una comparativa de algunos de los manejadores de bases de datos mencionados en la sección anterior. Para lo cual se tomó en cuenta ciertas características de los manejadores de bases de datos tales como el consumo de recursos, el soporte, la configuración, el espacio de almacenamiento, el uso de múltiples usuarios y su implementación en diferentes sistemas operativos.

	MySQL	ORACLE	PostgreSQL	SQL Server
<b>Consumo de recursos</b>	Bajo	Alto	Bajo	Alto
<b>Soporte</b>	Completo	Completo	Completo	Reducido
<b>Configuración</b>	Totalmente Accesible	Accesible	Totalmente Accesible	Totalmente Accesible
<b>Almacenamiento</b>	Ilimitado	Ilimitado	Ilimitado	Ilimitado
<b>Multiplataforma</b>	Sí	Sí	Sí	Sí
<b>Multiusuario</b>	Sí	Sí	Sí	Sí

**Tabla 2.2** Comparativa entre manejadores de bases de datos.

Fuente: Elaboración propia <sup>27</sup>

- **Consumo de recursos**

MySQL: Su consumo de recursos es bajo, por lo que puede ser implementado en cualquier máquina sin que sea una limitante las características del equipo. Entre las características que debe tener el equipo para su instalación se debe considerar un tamaño de 512 Mb de memoria RAM y 1 GB del disco duro.

Oracle: El consumo de recursos de un SGBD Oracle depende de la versión del software, por lo que el consumo es menor cuanto más básica sea la versión. Así, por ejemplo, el consumo de recursos de versiones gratuitas de Oracle, tal como la Express Edition es de 1 GB de memoria RAM. Mientras que las versiones de paga como la Standard Edition y Enterprise Edition, el consumo de memoria RAM de estas es la máxima que proporciona el sistema operativo, y 5 GB de disco duro para la instalación de las tres versiones.

<sup>27</sup> Tabla comparativa entre manejadores de base de datos.

PostgreSQL: Los usuarios pueden configurar el servidor de acuerdo a las características de su equipo. Por lo que los usuarios pueden configurar la cantidad de CPU, y la cantidad de memoria a utilizar. Con lo cual se puede obtener un mejor rendimiento del equipo.

SQL Server: El espacio en disco que utiliza un SGBD SQL Server depende de la versión y de sus características. Este suele ser de 4 GB para aquellas ediciones de gama alta. Además, utiliza mucho la memoria RAM para el almacenamiento de páginas. Estas páginas se consideran como la unidad mínima de E / S para llevar a cabo las operaciones de SQL Server, tales como el almacenamiento de metadatos, administración del espacio de almacenamiento de la base de datos, etc.

- **Soporte**

MySQL: La compañía Oracle es la responsable de brindar soporte a aquellas versiones de MySQL que aún se encuentran disponibles a los usuarios. Por lo cual, el tipo de soporte con el que cuentan las bases de datos implementadas en MySQL es el mismo que el de las bases de datos implementadas en Oracle.

Oracle: El periodo de tiempo durante el cual la compañía ofrece soporte depende del tipo de soporte mientras se cuente con la licencia de los productos Oracle. Así, por ejemplo, el soporte de una base de datos implementada en Oracle puede ser de cinco años por medio de la versión Premier. Dicho periodo puede ser aplazado tres años más por medio de la versión extendida. O bien, se puede recibir un soporte de forma indefinida por medio de un soporte de mantenimiento.

PostgreSQL: Los usuarios pueden acceder a la página de PostgreSQL, y hacer uso de los recursos que se brindan a los usuarios de forma gratuita como apoyo. En este tipo de soporte se ofrece documentación o atención a los usuarios, la cual proporciona la comunidad Postgres. Por si fuera poco, los usuarios pueden acceder a un soporte más personalizado mediante el soporte comercial que brinda PostgreSQL.

SQL Server: No se ofrece soporte para versiones con compatibilidad en plataformas Linux.

- **Configuración**

MySQL: Es fácil de instalar y configurar. Aunado a esto, la compañía Oracle ofrece un servicio denominado Oracle Service Cloud, el cual facilita la gestión de las bases de datos. Dicho servicio ofrece parches, copias de seguridad y recuperación, ajustes y monitoreo.

Oracle: Se proporciona un servicio a los usuarios denominado OCM (Oracle Configuration Manager), el cual recopila la información de la configuración de los usuarios para ofrecer un

mejor servicio de atención al cliente. Por otro lado, una configuración inadecuada puede afectar el funcionamiento del SGBD volviéndolo más lento.

PostgreSQL: En la documentación que se brinda acerca del SGBD PostgreSQL se detallan los diferentes procesos de instalación y configuración. Proporcionando la explicación del proceso de instalación de acuerdo al sistema operativo, y ofreciendo diferentes opciones de instalación y paquetes para que el usuario pueda configurar el servidor de acuerdo a sus necesidades.

SQL Server: Es fácil de instalar, configurar, administrar, e incluso cuenta con una herramienta denominada SQL Server Management Studio, la cual ofrece al usuario diferentes funciones para configurar y administrar diferentes componentes de SQL Server.

- **Almacenamiento**

MySQL: La cantidad de información que puede almacenar una base de datos en MySQL es indefinida. Así pues, el usuario puede implementar bases de datos de gran capacidad siempre y cuando no se exceda los límites en cuanto a la cantidad máxima de directorios que pueda manejar el sistema operativo, y en cuanto a el almacenamiento del disco duro.

Oracle: Una base de datos Oracle cuenta con un espacio de almacenamiento de 4 GB para la Express Edition. En tanto que el espacio de almacenamiento de la Standard Edition y la Enterprise Edition es ilimitado.

PostgreSQL: Su capacidad de almacenamiento es alta. Sus características de almacenamiento son un espacio límite en memoria de una tabla de 32 TB, un número máximo de bases de datos de 4,294,950,911, y un número máximo de tablas por base de datos de 1,431,650,303.

SQL Server: La versión más limitada (Express Edition) permite almacenar bases de datos de hasta 10 GB. Mientras que las versiones de uso empresarial, tales como SQL Standard Edition y SQL Enterprise Edition pueden llegar a almacenar bases de datos de hasta 524 petabytes.

- **Multiplataforma**

MySQL, Oracle y PostgreSQL se encuentran disponibles para sistemas operativos Windows y Linux. En cambio SQL Server se encuentra disponible principalmente para sistema operativo Windows. Solamente unas cuantas versiones están disponibles para Linux.

- **Multiusuario**

MySQL: La herramienta MySQL Enterprise Scalability proporciona a una base de datos rendimiento y escalabilidad, favoreciendo en el acceso concurrente y manteniendo un control en las consultas.

Oracle: El acceso concurrente a los usuarios está dado por diversos mecanismos, claramente uno de estos mecanismos es el de bloqueo. Por lo general este mecanismo es administrado de forma automática por el SGBD y puede ser de dos tipos: el bloqueo exclusivo y el bloqueo compartido. En el caso del bloqueo exclusivo, este se activa cuando un usuario modifica los datos, y se libera cuando termina de modificar los datos. En cambio, el bloqueo compartido se activa cuando uno o varios usuarios necesitan leer y compartir datos. Esto para evitar una activación simultánea de este mecanismo con el acceso concurrente. Suponiendo que en ese momento otro usuario desea acceder a la base de datos para modificar los datos (estos bloqueos pueden ser de una sola fila o de toda una tabla).

PostgreSQL: Ofrece un mecanismo denominado control de concurrencia de múltiples versiones (MVCC), el cual permite a los usuarios ejecutar múltiples transacciones de forma simultánea manteniendo dichas transacciones aisladas entre sí. El aislamiento de las transacciones se obtiene por medio de múltiples copias de la base de datos denominadas instantáneas. En este caso a cada consulta le corresponde una instantánea, por lo que los cambios que se realizan durante una consulta solamente son efectuados sobre la copia que le fue asignada. A su vez, esos cambios se efectúan sobre la base de datos una vez que la consulta ha sido concluida.

SQL Server: Ofrece un mecanismo el cual permite acceso concurrente a los usuarios a la vez que mantiene la integridad de los datos. Esto es mediante dos tipos de concurrencia denominadas concurrencia pesimista y concurrencia optimista. La concurrencia pesimista consiste en controlar el acceso concurrente por medio de bloqueos. La concurrencia optimista en cambio, permite realizar actualizaciones sin necesidad de bloqueos. En su lugar se crean dos versiones de la fila que se quiere actualizar. La versión actual de la fila se almacena en la base de datos, y la versión anterior de la fila se almacena en otra base de datos.

## **2.5 LENGUAJE DE CONSULTAS SQL**

### **2.5.1 Definición de SQL**

Se denomina SQL (Structured Query Language) al lenguaje utilizado para efectuar operaciones sobre la base de datos. Para lo cual, por medio del SGBD los usuarios pueden interactuar con la base de datos, al formular sus peticiones al SGBD mediante la declaración de sentencias SQL.

SQL apareció por primera vez en el año de 1970, su nombre inicial era SEQUEL (Structured English Query Language), y fue creado en los laboratorios IBM como parte de un proyecto denominado System R, cuyo objetivo era implementar el prototipo de un SGBD relacional, por lo que inicialmente desarrollaron un lenguaje de consultas. Posterior a esto, este lenguaje pasó a llamarse SQL. Tras su desarrollo se han ido creando nuevas versiones, las cuales incorporan nuevas funciones, por lo que SQL se ha convertido en el lenguaje estándar de los SGBD.

## 2.5.2 Sentencias SQL

### ▪ Sentencias DDL

El lenguaje de definición de datos o DDL es un lenguaje orientado a la creación y modificación de objetos que conforman la estructura de la base de datos. Para esta sección se consideran objetos tales como esquemas, tablas y vistas. Además, se proporciona una descripción de las sentencias asociadas a estos objetos.

#### *Sentencia para la creación de bases de datos*

La sentencia para la creación de una base de datos se efectúa por medio de la instrucción CREATE DATABASE, y se utiliza para agrupar un conjunto de objetos propios de una base de datos bajo un esquema (ya sea que se utilice esta sentencia, o una sentencia para la creación de un esquema). Para esto en la sentencia se debe especificar el nombre de la base de datos. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
CREATE DATABASE base de datos;
```

#### *Sentencia para la eliminación de bases de datos*

La sentencia para la eliminación de una base de datos se efectúa por medio de la instrucción DROP DATABASE. Para esto en la sentencia se debe especificar el nombre de la base de datos que requiere ser eliminada. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
DROP DATABASE base de datos;
```

#### *Sentencia para la creación de tablas*

La sentencia para la creación de una tabla se efectúa por medio de la instrucción CREATE TABLE, la cual da lugar a una nueva tabla correspondiente a la base de datos. Para esto, en la sentencia se debe especificar el nombre de la tabla, y la definición de cada una de las columnas de la tabla. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
CREATE TABLE tabla (  
  columna 1    dominio 1    restricción 1,  
  columna 2    dominio 2    restricción 2,  
  ⋮            ⋮            ⋮  
  columna n    dominio n    restricción n );
```

El formato empleado para la definición de una columna debe especificar el nombre de la columna, el dominio de la columna de acuerdo al tipo de dato que debe almacenar, y una restricción (opcional) en base a cada columna. A continuación, se muestra los diferentes dominios con los que puede definirse el tipo de dato correspondiente a cada columna.

Para almacenar una cadena de caracteres se utiliza un dominio ya sea del tipo CHAR(n) o VARCHAR(n) (donde n es la longitud fijada por el usuario, con la cual se establece un límite en cuanto al número de elementos que puede tener una cadena de caracteres). En el caso de que se introdujese una cadena con una longitud menor a n, cuando la columna es de tipo CHAR(n), para que la cadena ocupe toda la longitud especificada en el dominio, a esta se añaden espacios en blanco. En cambio, cuando la columna es de tipo VARCHAR(n), se almacena la cadena más no se incluyen espacios en blanco a esta.

Otro dominio para almacenar cadenas de caracteres es el de tipo TEXT, el cual permite almacenar cadenas de caracteres de diferentes longitudes. A continuación, se muestra una tabla con los dominios de tipo caracter.

<b>Dominio</b>	<b>Tipo de dato</b>
CHAR (n)	Cadena de caracteres de longitud fija
VARCHAR (n)	Cadena de caracteres de longitud variable
TEXT	Cadena de caracteres de cualquier longitud

**Tabla 2.3** Datos tipo caracter.

**Fuente:** Elaboración propia <sup>28</sup>

Para almacenar un número entero se utiliza un dominio de tipo entero, para lo cual se dispone de diferentes tipos de dominio, según el tamaño en memoria que estos ocupan para almacenar los datos. Por lo cual, los dominios utilizados para almacenar números enteros son SMALLINT para un número de 2 bytes, INT para un número de 4 bytes, y BIGINT para un número de 8 bytes, siendo más comúnmente utilizado el dominio del tipo INT.

Para almacenar números decimales se utiliza un dominio ya sea del tipo NUMERIC (p, d) o DECIMAL (p, d). Un dominio de este tipo para poder almacenar números decimales cuenta con una variable denominada precisión (p), y una variable denominada escala (d), las cuales son definidas por el usuario. La precisión representa el número de dígitos que componen el número decimal (tomando en cuenta la parte entera y la parte decimal). Mientras que la escala solamente considera el número de dígitos de la parte decimal. En este caso los dominios NUMERIC y DECIMAL son equivalentes. Sin embargo, al usar un dominio NUMERIC sin especificar la escala y la precisión permite almacenar valores numéricos sin importar su escala y precisión, además de que no forzará sus valores de entrada a una escala.

<sup>28</sup> Tabla ejemplificativa de los diferentes dominios de datos de tipo caracter.

Para almacenar un número con coma flotante se utiliza un dominio ya sea del tipo FLOAT(p), REAL o DOUBLE PRECISION. La diferencia entre cada dominio se debe a su precisión, por lo que la precisión de un dominio de tipo REAL es de al menos 6 dígitos, la de un dominio de tipo DOUBLE PRECISION es de al menos 15 dígitos. En el caso de un dominio de tipo FLOAT(p), este puede ser equivalente a un dominio de tipo REAL cuando el valor de p está dentro del rango entre 1 y 25, o ser equivalente a un dominio de tipo DOUBLE PRECISION cuando el rango está entre los valores 25 y 53. Incluso si se omite el valor de p un dominio de tipo FLOAT es equivalente a un dominio de tipo DOUBLE PRECISION. A continuación, se muestra una tabla con los diferentes dominios de tipo numérico.

<b>Dominio</b>	<b>Descripción</b>
INT	Número entero
NUMERIC (p, e)	Número con precisión arbitraria
DECIMAL (p, e)	Número con precisión arbitraria
FLOAT (p)	Número de punto flotante de precisión p
REAL	Número de punto flotante de baja precisión
DOUBLE PRECISION	Número de punto flotante de alta precisión

**Tabla 2.4** Datos tipo numérico.

**Fuente:** Elaboración propia <sup>29</sup>

Para almacenar la fecha o la hora se utilizan dominios en relación a este tipo de datos. Por lo cual, para almacenar datos de tipo fecha se utiliza un dominio de tipo DATE (la fecha es expresada por el año, mes y día. Por ejemplo: '2022-03-12'). Para almacenar datos de tipo hora se utiliza un dominio de tipo TIME (la hora es expresada en horas, minutos y segundos. Por ejemplo: '06:30:00'). O bien, es posible almacenar la unión de ambos datos (fecha y hora) utilizando un dominio de tipo TIMESTAMP (cuyo formato es la unión de la fecha y hora. Por ejemplo: '2022-03-12 06:30:00'). A continuación, se muestra una tabla con los dominios de tipo fecha y hora.

<sup>29</sup> Tabla ejemplificativa de los diferentes dominios de datos de tipo numérico.

<b>Dominio</b>	<b>Descripción</b>
DATE	Dato de tipo fecha
TIME	Dato de tipo hora
TIMESTAMP	Dato de tipo fecha y hora

**Tabla 2.5** Datos tipo fecha y hora.

Fuente: Elaboración propia <sup>30</sup>

Una vez que se ha asignado el nombre y dominio a una columna es conveniente definir algún tipo de restricción, debido a que en ocasiones los usuarios pueden realizar actualizaciones que comprometan la integridad de los datos. Por esta razón, a este tipo de restricciones se les denomina restricciones de integridad, las cuales se muestran a continuación.

<b>Restricción</b>	<b>Descripción</b>
NOT NULL	No permite introducir valores nulos
UNIQUE	No permite introducir valores repetidos
DEFAULT	Establece un valor como predeterminado
CHECK	No permite introducir valores si estos no cumplen con las condiciones especificadas

**Tabla 2.6** Restricciones de integridad.

Fuente: Elaboración propia <sup>31</sup>

Una restricción NOT NULL garantiza que todos los valores de la columna son válidos. Por otro lado, si se omite esta restricción o si se escribe la restricción DEFAULT NULL, el valor predeterminado de la columna es NULL.

Una restricción UNIQUE garantiza que todos los valores de la columna son únicos. Una columna con esta restricción es designada como clave alternativa.

Una restricción DEFAULT (valor) permite definir un valor como predeterminado. De esta forma, el valor indicado entre paréntesis es asignado en cada fila de la columna que no tiene un valor asignado.

<sup>30</sup> Tabla ejemplificativa de los diferentes dominios de datos de tipo fecha y hora.

<sup>31</sup> Tabla ejemplificativa de las diferentes restricciones de columna.

Una restricción CHECK (condición) permite especificar una condición de validez. De esta forma, solo se registran aquellos datos que cumplen la condición especificada entre paréntesis. El planteamiento de tales condiciones se realiza de la misma forma como se realiza las condiciones de búsqueda utilizadas para la consulta de datos, como es el caso del uso de los operadores de comparación o de una lista especificando los valores admitidos por la columna.

Una columna con un dominio de tipo fecha y hora puede llevar una restricción de tipo DEFAULT, de forma que el valor predeterminado de la columna sea la fecha actual para un dominio de tipo DATE, la hora actual para un dominio de tipo TIME, o la fecha y la hora actual para un dominio TIMESTAMP. En este caso se utiliza como argumento de la restricción alguna de las funciones establecidas según el tipo de dominio. Tales funciones se muestran a continuación.

<b>Función</b>	<b>Descripción</b>
CURRENT_DATE	Registra la fecha actual
CURRENT_TIME	Registra la hora actual
CURRENT_TIMESTAMP	Registra la fecha y hora actual

**Tabla 2.7** *Funciones de tipo fecha y hora.*

Fuente: Elaboración propia <sup>32</sup>

Para especificar una relación de correspondencia, donde las filas de una tabla guardan una relación con las filas de otra tabla, se debe designar una clave primaria y una clave foránea. Suponiendo que dos tablas presentan una relación de correspondencia, se designa una tabla (tabla padre) para que cada una de sus filas de la clave primaria estén asociadas con cero o más filas de una de las columnas de la otra tabla (tabla hijo). Por otro lado, la otra tabla debe tener designada una clave foránea (la cual es una columna, donde cada una de sus filas están asociadas exactamente con una de las filas de la clave primaria de la tabla padre).

Para conformar una clave primaria se utiliza la instrucción PRIMARY KEY (columnas), lo cual implica que las columnas especificadas no admiten valores nulos, y no permiten que dos o más filas de cada columna tengan valores repetidos (ya sea que la clave primaria esté constituida por más de una columna, para lo cual deben estar separadas por comas).

Para conformar una clave foránea se utiliza una sentencia con la instrucción FOREIGN KEY. Esta va acompañada de las columnas de la tabla que constituyen la clave foránea, además debe llevar

---

<sup>32</sup> Tabla ejemplificativa de las diferentes funciones de los datos de tipo fecha y hora.

el nombre de la tabla padre (el nombre de la tabla padre puede estar acompañado de sus columnas que conforman su clave primaria siendo algo opcional). El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**FOREIGN KEY (columnas) REFERENCES tabla (columnas) ON DELETE restricción**

Las sentencias para establecer una clave primaria o clave foránea son especificadas después de haber declarado cada una de las columnas de la tabla.

Al igual que las columnas, las tablas también pueden ser complementadas con restricciones que permitan actualizar los datos sin que se pierda su integridad. Estas son opcionales y son incluidas dentro de la sentencia de clave foránea mediante la frase ON DELETE.

La integridad referencial es la que se ve afectada si alguna de las filas de la tabla hijo pierde su relación de correspondencia tras haber actualizado la fila de la tabla padre con la cual está asociada. Por lo cual es recomendable especificar algún tipo de restricción (también conocidas como acciones de activación referencial). Para determinar lo que debe suceder con las filas de la tabla hijo cuando son actualizadas las filas de la tabla padre se debe de escoger algún tipo de restricción. Los diferentes tipos de restricción son los que se presentan a continuación.

Restricción	Descripción
RESTRICT	No permite eliminar una fila o actualizar su clave primaria si es que la fila tiene hijos
CASCADE	Transmite actualizaciones en cascada entre filas cuando presentan una relación padre/hijo
SET NULL	Asigna un valor nulo en la clave foránea de una fila si la fila que tiene asignada de la tabla padre es actualizada
SET DEFAULT	Asigna un valor predeterminado en la clave foránea de una fila si la fila que tiene asignada de la tabla padre es actualizada

**Tabla 2.8** Acciones de activación referencial.

Fuente: Elaboración propia <sup>33</sup>

<sup>33</sup> Tabla ejemplificativa de las diferentes restricciones de tabla.

En una tabla con una restricción RESTRICT está permitido eliminar una fila de la tabla padre o modificar su clave primaria, solo si la clave de la fila no está asociada con alguna de las filas de la tabla que contiene la clave foránea.

Una tabla con la restricción CASCADE al eliminar una fila de la tabla padre o actualizar su clave primaria, elimina en cascada cada una de sus filas que están asociadas con la fila de la tabla padre, o actualiza el valor de la clave primaria referenciada de cada una de sus filas.

Una tabla con la restricción SET NULL al eliminar una fila de la tabla padre o actualizar su clave primaria, cambia el valor de la clave foránea de cada una de sus filas que están asociadas con la fila de la tabla padre por un valor NULL.

Una tabla con la restricción SET DEFAULT al eliminar una fila de la tabla padre o actualizar su clave primaria, cambia el valor de la clave foránea de cada una de sus filas que están asociadas con la fila de la tabla padre por el valor que tienen designado por defecto.

En caso de que no sea especificada algún tipo de restricción, la restricción que es asignada a la tabla por defecto es del tipo RESTRICT.

La restricción de integridad sigue presente aun cuando si al insertar una fila en la tabla padre esta no tiene hijos, o cuando en una tabla hijo se actualiza una clave foránea o se inserta una nueva fila. En ambos casos el valor de la clave foránea de esa fila debe coincidir con un valor de la clave primaria de la tabla padre. Incluso la integridad referencial permanece si se elimina una fila de la tabla hijo, en cuyo caso la fila de la tabla padre asociada a esa fila solamente tendrá un hijo menos.

Una vez definida la tabla, esta se encuentra vacía. Por lo cual la sentencia de inserción del lenguaje DML permite introducir valores a la tabla (la cual es detallada más adelante).

### ***Sentencia para listar tablas***

La sentencia para listar tablas se efectúa por medio de la instrucción SHOW TABLE, y se utiliza para desplegar los nombres de todas las tablas que componen a la base de datos. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

#### **SHOW TABLES;**

La sentencia también puede ser escrita como SHOW FULL TABLES para desplegar el nombre de cada una de las tablas y su tipo de tabla correspondiente, cuya clasificación está basada en dos tipos: Esto es VIEW (en el caso de una vista) o BASE TABLE (en el caso de una tabla).

### ***Sentencia para la modificación de tablas***

La sentencia para la modificación de una tabla de la base de datos se efectúa por medio de la instrucción ALTER TABLE, lo cual resulta en la modificación de la estructura de la tabla. Para esto, en la sentencia se debe especificar el nombre de la tabla junto con alguna sentencia de modificación. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**ALTER TABLE *tabla* (*sentencia de modificación*);**

Por medio de la sentencia de modificación se especifica algún tipo de acción, ya sea la adición, modificación o eliminación de un elemento, para lo cual se utilizan las palabras clave ADD, ALTER o DROP. Así pues, las modificaciones que pueden ser realizadas por medio esta sentencia están la adición o eliminación de una columna, y la adición o eliminación de una restricción, ya sea una restricción de tabla o de columna.

A continuación, se especifica la forma en que debe realizarse cada declaración según el elemento a modificar y el tipo de modificación.

- Columnas

**ADD *columna dominio*;**  
**ALTER *columna* [SET | DROP] *restricción*;**  
**DROP *columna* [RESTRICT | CASCADE];**

- Restricciones

**ADD *restricción*;**  
**DROP CONSTRAINT *restricción* [RESTRICT | CASCADE];**

En una sentencia de eliminación, para evitar eliminar un elemento que está referenciado se utiliza la palabra clave RESTRICT. Por otro lado, la palabra clave CASCADE elimina dicho elemento junto con todo aquello que referencia a dicho elemento.

Puede darse el caso de que, la sentencia para la creación de una tabla contemple la creación de una clave foránea que haga referencia a la clave primaria de otra tabla que aún no ha sido creada. Esta situación generará un error, por lo cual primero es necesario crear cada una de las tablas de la base de datos, evitando incluir en la sentencia la creación de claves foráneas.

Una vez que las tablas han sido declaradas, mediante una sentencia de modificación de tabla es posible definir una clave foránea para cada una de las tablas que lo requieran.

### ***Sentencia para la eliminación de tablas***

La sentencia para la eliminación de una tabla de la base de datos se efectúa por medio de la instrucción DROP TABLE. Para esto, en la sentencia se debe especificar el nombre de la tabla junto con alguna de las restricciones antes mencionadas para definir si la tabla está referenciada y que hacer al respecto si es el caso. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**DROP TABLE *tabla* [RESTRICT | CASCADE];**

Una vez que la tabla fue eliminada no hay forma de recuperar su esquema o cualquiera de sus datos. Suponiendo que alguien desee ingresar datos en una tabla que ha sido borrada, este tendrá que crear nuevamente la tabla.

### ***Sentencia para la creación de vistas***

La sentencia para la creación de vistas en una base de datos se efectúa por medio de la instrucción CREATE VIEW y la palabra AS, y se utiliza para establecer consultas predefinidas por medio de las vistas, para que los usuarios puedan acceder a un subconjunto de la base de datos. Para esto, en la sentencia se debe seleccionar un nombre para vista, se colocan los nombres que tendrán las columnas en la vista (si es que se desea cambiar su nombre original), y por último se declara una sentencia de selección para incluir en la vista solamente aquellos datos que sean relevantes para los usuarios. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**CREATE VIEW *vista* (*columna(s)*) AS (*consulta*) [WITH CHECK OPTION];**

Para crear una vista se utiliza la opción WITH CHECK OPTION al final si es que dentro de la sentencia se utiliza una consulta compuesta. Esto se debe a que cuando se omite esta opción, la vista se efectúa sin tomar en cuenta la condición de selección que realiza la consulta (condición que es especificada después de la palabra clave WHERE).

### ***Sentencia para la eliminación de vistas***

La sentencia para la eliminación de vistas en una base de datos se efectúa por medio de la instrucción DROP VIEW, lo cual resulta en la eliminación de la vista solicitada. Para esto, en la sentencia se debe especificar el nombre con el cual fue creada la vista. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**DROP VIEW *vista* [RESTRICT | CASCADE];**

De forma opcional la sentencia puede ir acompañada por la palabra clave RESTRICT para evitar que la vista sea eliminada si es que está referenciada. En caso de que se desee eliminar la vista junto con todo aquello que la referencia se utiliza la palabra clave CASCADE.

- **Sentencias DML**

El lenguaje de manipulación de datos o DML es un lenguaje orientado a la consulta y manipulación de los datos contenidos en la base de datos. Para esta sección se proporciona una descripción de las sentencias para efectuar consultas de datos (simples y compuestas), así como para insertar, modificar o borrar los datos de una tabla de la base de datos.

***Sentencia para la selección de datos***

La sentencia de selección se efectúa por medio de la instrucción SELECT y la palabra FROM, y se utiliza para consultar los datos de una tabla de la base de datos. Para esto, en la sentencia se debe especificar el nombre de las columnas que se desean consultar (en este caso se requiere que las columnas estén separadas por comas), y el nombre de la tabla a la cual pertenecen dichas columnas. Como resultado la sentencia muestra una tabla que contiene todas las columnas que se le hayan especificado junto con todas sus filas. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**SELECT *columna(s)* FROM *tabla*;**

Sin embargo, habrá ocasiones en las que se requiera acceder a todas las columnas de la tabla. Para lo cual no es necesario escribir el nombre de cada una de las columnas, en su lugar se utiliza el símbolo (\*). El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**SELECT \* FROM *tabla*;**

En caso de que se requiera acceder a las columnas de una tabla omitiendo valores repetidos se utiliza la palabra clave DISTINCT después de la palabra SELECT. Si por el contrario se requiere desplegar todos los valores de la columna no importando que en el resultado se tengan valores repetidos se omite la palabra DISTINCT o se sustituye por la palabra clave ALL. El aspecto que debe tener una sentencia con la palabra clave DISTINCT es el que se muestra a continuación.

**SELECT DISTINCT *columna(s)* FROM *tabla*;**

Cabe mencionar que las sentencias hasta ahora mencionadas obtienen datos de forma generalizada, incluyendo datos que podrían estar por demás en la consulta. Por tanto, para obtener datos que cumplan con una condición más específica se realiza una consulta compuesta. Es decir, una consulta que haga uso de la palabra clave WHERE seguida de la condición que especifica las características que deben tener los datos. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**SELECT *columna(s)* FROM *tabla* WHERE *condición*;**

Una condición con un operador de comparación se utiliza para realizar una consulta en la que se compara el valor de una fila (correspondiente a la columna especificada en la condición) con

el valor mostrado al otro lado de la expresión de comparación. Como resultado se despliega todas las filas que cumplan con la condición. El aspecto de tal condición es el que se muestra a continuación.

```
columna = valor;  
columna < valor;  
columna > valor;  
columna <= valor;  
columna >= valor;  
columna <> valor;
```

Por lo general, el valor de comparación es un valor numérico, y los operadores de comparación son =, <, >, <=, >= y <> cuyo significado de cada uno es igual, menor que, mayor que, menor o igual que, mayor o igual que y diferente respectivamente.

Una condición puede estar conformada por dos o más restricciones unidas mediante operadores lógicos (las restricciones son en sí condiciones conformadas con operadores de comparación). Una condición con el operador lógico AND se utiliza para realizar una consulta que muestra aquellas filas que cumplen de forma simultánea todas las condiciones especificadas en la consulta. El aspecto que debe tener tal condición es el que se muestra a continuación.

```
columna condición 1 AND columna condición 2;
```

Una condición con el operador lógico OR se utiliza para realizar una consulta que muestra aquellas filas que cumplen con cualquiera de las condiciones especificadas en la consulta. Ya sea que solo cumplan una de ellas o cumplan con ambas. El aspecto que debe tener tal condición es el que se muestra a continuación.

```
columna condición 1 OR condición 2;
```

Una condición con el operador lógico NOT se utiliza para realizar una consulta que muestra aquellas filas que según la sentencia presentan un valor falso al especificado en la condición. El aspecto que debe tener tal condición es el que se muestra a continuación.

```
NOT columna condición;
```

Una condición con la palabra clave BETWEEN se utiliza para realizar una consulta que muestra las filas cuyos valores (correspondientes a la columna especificada en la condición) sean mayor o igual al especificado como límite 1 y menor o igual al especificado como límite 2. El aspecto que debe tener tal condición es el que se muestra a continuación.

```
columna BETWEEN límite1 AND límite2;
```

Si por el contrario se requiere que los valores de selección estén fuera del rango establecido, se utiliza la misma condición, pero en su lugar se utiliza la frase NOT BETWEEN en lugar de BETWEEN.

Una condición con la instrucción IS NULL se utiliza para realizar una consulta que muestra las filas que tienen un valor nulo (en la columna especificada en la condición). El aspecto que debe tener tal condición es el que se muestra a continuación.

*columna IS NULL;*

Si por el contrario se requiere que la consulta omita las filas con valores nulos, se utiliza la misma condición, pero en su lugar se utiliza la frase IS NOT NULL en lugar de IS NULL.

Una condición con la palabra clave LIKE se utiliza para realizar una consulta que muestra las filas cuyos valores (correspondientes a la columna especificada en la condición) coincidan con un patrón específico. El formato de dicho patrón se realiza con elementos de tipo carácter, los símbolos (\_) y (%) o una combinación entre estos. El aspecto que debe tener tal condición es el que se muestra a continuación.

*columna LIKE patrón;*

A continuación, se presenta una serie de ejemplos que muestran las diferentes formas en las que se puede realizar el patrón de comparación, en base al criterio de búsqueda requerido.

- Se utiliza el patrón de búsqueda ('\_\_\_\_\_') para especificar que los datos seleccionados tienen que ser de seis caracteres. Ejemplo: puerta, espejo, llanta, etc.
- Se utiliza el patrón de búsqueda ('Lu\_\_') para especificar que los datos seleccionados tienen que ser de cuatro caracteres y que tienen que empezar con "Lu". Ejemplo: Luis, Luna, Lulú, etc.
- Se utiliza el patrón de búsqueda ('\_\_\_ pa \_\_\_\_') para especificar que los datos seleccionados tienen que ser de nueve caracteres. Para lo cual el cuarto y quinto carácter tienen que estar ocupados por las letras "p" y "a" respectivamente. Ejemplo: compactar, despachar, respaldar, etc.
- Se utiliza el patrón de búsqueda ('\_\_\_\_\_ or') para especificar que los datos seleccionados tienen que ser de ocho caracteres y que tienen que terminar con "or". Ejemplo: vendedor, operador, profesor, etc.
- Se utiliza el patrón de búsqueda ('pa%') para especificar que los datos seleccionados tienen que empezar con "pa". Ejemplo: paella, panucho, pambazo, etc.

- Se utiliza el patrón de búsqueda ('%ar%') para especificar que los datos seleccionados tienen que contener "ar" como subcadena. Ejemplo: Lara, Barrera, Camarena, etc.
- Se utiliza el patrón de búsqueda ('%lia') para especificar que los datos seleccionados tienen que terminar con "lia". Ejemplo: Argelia, Australia, Italia, etc.
- Se utiliza el patrón de búsqueda ('\_\_\_\_\_') para especificar que los datos seleccionados tienen que ser de al menos siete caracteres. Ejemplo: lechuga, jitomate, zanahoria, etc.
- Se utiliza el patrón de búsqueda ('\_ ni%') para especificar que los datos seleccionados tienen que empezar con un caracter, continuar con "ni" como subcadena, y terminar con un patrón cualquiera de caracteres. Ejemplo: animado, enigmático, universal, etc.

Una condición con la palabra clave IN se utiliza para realizar una consulta que compare el valor de una fila (correspondiente a la columna especificada en la condición) con una lista de valores, y como resultado se despliegan aquellas filas cuyos valores coinciden con la lista. El aspecto que debe tener tal condición es el que se muestra a continuación.

***columna IN (conjunto de valores);***

Si por el contrario se requiere que la consulta omita aquellas filas cuyos valores coinciden con los mostrados en la lista, se utiliza la misma condición, pero en su lugar se utiliza la frase NOT IN en lugar de IN.

En una consulta se utiliza la instrucción ORDER BY para mostrar las filas correspondientes a la consulta ordenadas en base al orden ascendente (alfabético o numérico) de los valores de las columnas que acompañan a la instrucción ORDER BY. El orden ascendente se obtiene escribiendo la palabra ASC después del nombre de la columna (aunque también se obtiene el mismo resultado aun cuando se omita la palabra ASC). Por otro lado, por medio de la palabra DESC se especifica que el ordenamiento de los datos debe realizarse de forma descendente. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**SELECT *columna(s)* FROM *tabla* ORDER BY *columna(s)*;**

En caso de que se requiera ordenar las filas en una consulta compuesta, la sentencia se escribe de la misma forma en que se realiza una consulta compuesta acompañada de la instrucción ORDER BY junto con las columnas de ordenamiento. También es posible escribir dos columnas de ordenamiento, así cuando la primera columna de ordenamiento presenta valores repetidos, en tal caso el ordenamiento de las filas con valores repetidos entre si se realiza en base a la segunda columna.

Una sentencia de selección también puede ser especificada como una combinación de columnas de diferentes tablas. Para esto, en la sentencia se debe especificar el nombre de las columnas que se desean consultar, y el nombre de las tablas a las cuales pertenecen dichas columnas. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
SELECT columna 1, columna 2, columna 3 FROM tabla-a, tabla-b WHERE condición;
```

Para poder efectuar este tipo de consultas es necesario que las tablas presenten una relación padre/hijo, en cuyo caso en la condición de selección debe estar especificada esta relación. Esto es mediante la igualación de la clave primaria de la tabla padre con su correspondiente clave foránea de la tabla hijo. Por lo cual, este tipo de consulta recibe el nombre de consulta padre/hijo.

Es posible incluir más condiciones de selección después de especificar la relación padre/hijo, para lo cual se utiliza el operador AND seguido de la siguiente condición de búsqueda.

En una consulta los nombres de las columnas suelen ser siempre los mismos que los nombres de las columnas de la tabla original. Sin embargo, en una consulta es posible desplegar diferentes columnas con un nombre diferente a su nombre original. Para esto, en la sentencia de selección, en la sección donde están las columnas se debe especificar el nombre de la columna original, la palabra clave AS seguido del nuevo nombre que se le quiere asignar. En cambio, las columnas que no estén especificadas de esta forma conservan su nombre original. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
columna 1 AS c1, columna 2 AS c2, columna 3 AS c3, ...
```

Al realizar una sentencia de selección con más de una tabla puede ocurrir que dos tablas tengan una columna con el mismo nombre, por lo que se puede crear una falla en la consulta. Debido a esto, en la sentencia cada columna debe llevar primero el nombre de la tabla a la cual pertenece, seguido del nombre de la columna y un punto (.) entre ambos nombres.

Si además de esto se desea asignar un nombre diferente para cada tabla utilizada en la consulta, en la sentencia el nombre original de cada tabla debe ir acompañado de un espacio en blanco seguido del nuevo nombre que se le quiera asignar. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

```
SELECT U.columna, V.columna ... FROM tabla-a U, tabla-b V WHERE condición;
```

También se requiere especificar el nombre de la tabla a la cual pertenece una columna en la condición de selección, si es que se requiere comparar dos columnas que tengan el mismo nombre.

### ***Sentencia para la inserción de datos***

La sentencia de inserción se efectúa por medio de la instrucción INSERT INTO y la palabra VALUES, y se utiliza para insertar datos dentro de una tabla de la base de datos. Cabe mencionar que la ejecución de este tipo de instrucción genera una nueva fila dentro de una tabla. Esto porque para agregar más datos se requiere crear nuevas filas que contengan dichos datos. Para esto, en la sentencia se debe especificar el nombre de la tabla, pero para especificar los datos se deben tomar en consideración los siguientes puntos:

- Los datos tienen que ser especificados de forma que se tenga uno por cada columna (los cuales deben estar separados por comas).
- Los datos de la lista deben ser ordenados tomando en cuenta que estos son acomodados uno por uno sobre la tabla en un orden de izquierda a derecha.
- La lista de los datos se debe realizar prestando atención en que el tipo de dato de cada uno de los datos coincida con el dominio de sus correspondientes columnas.
- La palabra reservada NULL se utiliza para poder asignar un valor en aquellas columnas que no tienen un valor definido. Esto da como resultado la inserción de datos permitiendo la omisión de algunos de estos.

El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**INSERT INTO *tabla* VALUES (*valores*);**

Si bien, no es necesario especificar el nombre de las columnas para efectuar la inserción de los datos, la sentencia si puede ser escrita de esta forma (en este caso deben estar separados por comas). La sentencia escrita de esta forma es de utilidad en el aspecto de que ya no es necesario acomodar los datos en el mismo orden al presentado en la tabla, sino que se acomodan siguiendo el orden presentado en la lista con los nombres de las columnas. En caso de no disponer de todos los datos de las columnas para realizar la inserción, se puede omitir el nombre y el valor de dichas columnas en la sentencia, y en automático se le asigna un valor nulo o un valor por defecto. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**INSERT INTO *tabla* (*columnas*) VALUES (*valores*);**

### ***Sentencia para la eliminación de datos***

La sentencia de eliminación se efectúa por medio de la instrucción DELETE FROM y la palabra WHERE, y se utiliza para eliminar los datos de una tabla de la base de datos. Para esto, en la sentencia se debe especificar el nombre de la tabla y una condición de selección. El resultado es

la eliminación de aquellas filas que cumplan con la condición. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**DELETE FROM *tabla* WHERE *condición*;**

Es importante mencionar que la sentencia solamente puede borrar filas de una tabla, por lo que no es posible utilizar esta sentencia para eliminar datos de ciertas columnas en concreto.

En caso de que se requiera eliminar todas las filas de una tabla, se utiliza la misma sentencia, pero en este caso se omite la palabra clave WHERE junto con la condición de selección.

### ***Sentencia para la actualización de datos***

La sentencia de actualización se efectúa por medio de la instrucción UPDATE, y las palabras SET y WHERE, y se utiliza para actualizar los datos de una tabla de la base de datos. Para esto, en la sentencia se debe especificar el nombre de la tabla, el nombre de las columnas que requieren ser actualizadas asignando un nuevo valor a cada una de estas (en este caso las sentencias de asignación deben estar separadas por comas), y una condición de selección. El resultado es que los valores de las columnas son modificados por los especificados, aunque solamente cambian aquellos valores pertenecientes a las filas que cumplen con la condición de selección.

El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**UPDATE *tabla* SET *columna* = *valor* WHERE *condiciones*;**

En caso de que se quiera actualizar una columna cambiando todos sus valores a un solo valor, se utiliza la misma sentencia, pero en este caso se omite la palabra clave WHERE junto con la condición de selección.

### ▪ **Sentencias DCL**

El lenguaje de control de datos o DCL es un lenguaje orientado a gestionar el acceso, las transacciones y los permisos de los usuarios que interactúan con la base de datos. Para esta sección se proporciona una descripción de las sentencias que están asociadas a la creación y eliminación de sesiones de usuario, las sentencias para realizar transacciones, así como las sentencias asociadas a la autorización y remoción de privilegios dirigidos a los usuarios.

### ***Sentencia para la creación de usuarios***

Por norma, un SGBD permite a los usuarios ingresar a la base de datos no sin antes pedirle iniciar sesión, ingresando un nombre de usuario y una contraseña. En base a esto, los nombres y contraseñas permitidos para ingresar a la base de datos son definidos por medio de una sentencia, la cual se efectúa por medio de la instrucción CREATE USER y la frase WITH PASSWORD. Para esto, en la sentencia se debe especificar un nombre correspondiente al

usuario y una contraseña. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**CREATE USER *nombre de usuario* WITH PASSWORD *contraseña*;**

#### ***Sentencia para la eliminación de usuarios***

La sentencia para la eliminación de un usuario se efectúa por medio de la instrucción DROP USER. Para esto, en la sentencia se debe especificar el nombre de usuario que se desea eliminar. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**DROP USER *nombre de usuario*;**

#### ***Sentencia para formular transacciones***

Un usuario para interactuar con la base de datos debe ejecutar sentencias SQL, para lo cual se dice que establece una transacción con la base de datos cada vez que ejecuta una instrucción. Estas transacciones se realizan de forma implícita. Sin embargo, el usuario para efectuar una transacción de forma explícita lo puede hacer por medio de una sentencia con la instrucción SET TRANSACTION, seguido de esto debe especificar si la transacción es para efectuar consultas con la frase READ ONLY, o si desea efectuar actualizaciones con la frase READ WRITE. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**SET TRANSACTION [READ ONLY | READ WRITE];**

Una vez concluida la transacción, se debe ejecutar alguna de las sentencias mostradas a continuación.

**[COMMIT | ROLLBACK];**

Esto se debe a que, si al concluir la transacción el usuario realizó cambios en la base de datos, al ejecutar la sentencia COMMIT tales cambios serán confirmados, haciendo que se guarden los cambios. Por otro lado, al ejecutar la sentencia ROLLBACK tales cambios son descartados, restaurando la base de datos al mismo estado que tenía antes de que se iniciara la transacción.

#### ***Sentencia para la asignación de privilegios***

Tomando en cuenta que múltiples usuarios pueden acceder al sistema de base de datos, la interacción de cada uno de ellos con la base de datos depende de los privilegios que le hayan sido otorgados como usuario para ingresar a la base de datos. La sentencia para otorgar privilegios a los usuarios se efectúa por medio de la instrucción GRANT, y las palabras ON y TO. Para esto, en la sentencia se debe especificar los privilegios, el objeto al cual corresponde los privilegios y el nombre de usuario o usuarios (separados por comas) a los que se le están otorgando los privilegios. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**GRANT privilegio(s) ON objeto TO usuario(s) [WITH GRANT OPTION];**

De forma opcional la sentencia puede ir acompañada por la frase WITH GRANT OPTION para permitir que el usuario a su vez pueda otorgar estos privilegios a otros usuarios.

Un privilegio debe formar parte de los privilegios de un usuario para que ese usuario pueda otorgar ese privilegio a otro usuario ejecutando la sentencia antes mencionada.

Es importante conocer estos privilegios para garantizar una adecuada administración de los recursos de la base de datos. A continuación, se muestran algunos de los privilegios que pueden tener los usuarios en un sistema de base de datos.

Privilegio	Descripción
SELECT	Permite consultar la tabla o la vista especificada
INSERT	Permite insertar filas en la tabla o la vista especificada
DELETE	Permite eliminar la tabla o la vista especificada
UPDATE	Permite actualizar la columna de la tabla o la vista especificada

**Tabla 2.9** Privilegios para interactuar con la base de datos.

Fuente: Elaboración propia <sup>34</sup>

Se debe tomar en cuenta que cuando se utiliza UPDATE en una sentencia para otorgar privilegios, esta palabra debe ir acompañada de una lista entre paréntesis, y separada por comas de los nombres de las columnas a las cuales el usuario tiene derecho de actualizar. Por otro lado, si después de la instrucción UPDATE no se incluye una lista, el usuario podrá realizar actualizaciones en cualquiera de las columnas de la tabla especificada.

Otro privilegio que también puede ser especificado con una lista de nombres de columnas es INSERT. Lo cual implica que el usuario solamente tiene derecho a ingresar datos en aquellas columnas que hayan sido especificadas en la lista, mientras que los datos para las columnas restantes serán valores predeterminados o valores nulos.

En caso de que sea necesario otorgar todos los privilegios a algún usuario, la sentencia debe tener la frase ALL PRIVILEGES como privilegio. Por otro lado, si se desea otorgar uno de los

---

<sup>34</sup> Tabla ejemplificativa de los privilegios designados a los usuarios.

privilegios a todos los usuarios del sistema de base de datos (incluidos los futuros usuarios), se utiliza la palabra clave PUBLIC como el usuario al que se quiere otorgar el privilegio.

### ***Sentencia para la eliminación de privilegios***

En caso de que se requiera retirar algún privilegio a un usuario, esto se efectúa por medio de una sentencia con la instrucción REVOKE, y las palabras ON y FROM. Para esto, en la sentencia se debe especificar los privilegios que se requieren eliminar, el nombre de la tabla o la vista que esté asociada con tales privilegios, y el nombre del usuario al cual se le tiene que retirar el privilegio. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**REVOKE *privilegio(s)* ON *objeto* FROM *usuarios* [RESTRICT | CASCADE];**

La sentencia puede estar acompañada de la palabra clave CASCADE, lo cual implica que el privilegio declarado en la sentencia no solamente será eliminado para ese usuario, sino que también otros usuarios que tengan ese privilegio lo perderán (siempre y cuando los usuarios hayan adquirido ese privilegio por medio de ese usuario). Aunque la sentencia de forma predeterminada elimina los privilegios a todos los usuarios, por lo cual se puede omitir la palabra clave CASCADE para este efecto. Por otro lado, la palabra clave RESTRICT evita eliminar el privilegio a un usuario si es que ese usuario a otorgado ese privilegio a otros usuarios.

En caso de que el usuario deba mantener ese privilegio y solo se quisiese retirar su capacidad para conceder ese privilegio a otros usuarios, en la sección de la sentencia donde está especificado el privilegio se escribe la frase GRANT OPTION FOR acompañado del privilegio.

Por otro lado, para remover todos los privilegios que tiene un usuario sobre la base de datos, la sentencia debe tener la frase ALL PRIVILEGES como privilegio, y DATABASE junto con el nombre de la base de datos como objeto.

### ***Sentencia para la creación de roles***

También existe un concepto al que se le denomina rol, el cual es considerado como una agrupación de privilegios, de forma que los privilegios que pertenezcan al rol serán asignados de forma automática a los usuarios que tengan asignado dicho rol. La sentencia para crear un rol se efectúa por medio de la instrucción CREATE ROLE. Para esto, en la sentencia se debe especificar un nombre correspondiente al rol. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**CREATE ROLE *nombre de rol*;**

Para proporcionar privilegios a un rol se utiliza la misma sentencia con la cual se proporcionan los privilegios a un usuario, siendo el nombre de usuario reemplazado por el nombre del rol dentro de la sentencia. Si por el contrario lo que se quiere es eliminar un privilegio asignado a un rol, se utiliza la misma sentencia con la cual se elimina los privilegios a un usuario.

### ***Sentencia para la asignación de roles***

La sentencia para asignar un role a un usuario se efectúa por medio de la instrucción GRANT y la palabra TO. Para esto, en la sentencia se debe especificar el nombre del role para asignar y el nombre del usuario al que se le debe designar el rol. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**GRANT** *nombre del role* **TO** *nombre del usuario*;

### ***Sentencia para la eliminación de roles***

La sentencia para la eliminación de un role se efectúa por medio de la instrucción DROP ROLE. Para esto, en la sentencia se debe especificar el nombre del role que se desea eliminar. El aspecto que debe tener tal sentencia es el que se muestra a continuación.

**DROP ROLE** *nombre de role*;

## 2.6 APLICACIONES

Las bases de datos facilitan el manejo de la información estando presentes en infinidad de aplicaciones. Sus aplicaciones (por mencionar algunas) son:

- **Escuelas**

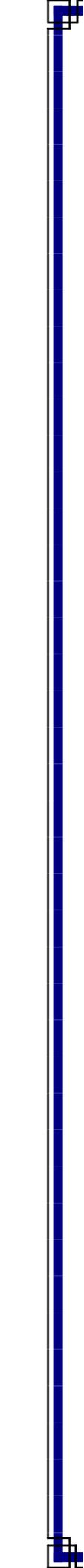
Se almacena información de los estudiantes como nombre, materias, grupos, calificaciones e información adicional. También se puede llevar un registro de la información del personal, mobiliario y toda aquella información utilizada para la gestión de la institución.

- **Bibliotecas**

Se almacena información referente a los libros como título, autor, tema, editorial, entre otros. Además, si la biblioteca ofrece el préstamo de los libros, se tendrá un registro de la información de los usuarios y de los préstamos que realicen.

- **Bancos**

Se almacena información relacionada como información personal del usuario, cuentas bancarias, montos, transacciones, préstamos, entre otros.



# **CAPÍTULO 3**

## **IMPLEMENTACIÓN DE LA BASE DE DATOS**

# Capítulo 3 IMPLEMENTACIÓN DE LA BASE DE DATOS

## 3.1 JUSTIFICACIÓN DE LAS HERRAMIENTAS

### 3.1.1 ¿Por qué GNU/Linux?

Entre los sistemas operativos más usados hoy en día tenemos al sistema operativo de Windows y al sistema operativo GNU/Linux, de los cuales debido a sus características para el desarrollo de este trabajo se ha optado por utilizar GNU/Linux. Entre las características por las cuales destaca este sistema operativo tenemos las siguientes:

- **Inversión**

El nivel de inversión en las aplicaciones de software de código abierto es mayor que la que se hace en las aplicaciones de software propietario. Los desarrolladores de código abierto realizan esta inversión buscando motivar a más personas a hacer uso del sistema operativo GNU/Linux. Ellos a su vez adquieren reconocimiento y beneficios profesionales entre mayor popularidad adquiera el software desarrollado.

- **Preferencia**

En el desarrollo de aplicaciones la comunidad de código abierto toma en consideración que estas sean similares a las aplicaciones de Windows, esto para facilitar a los usuarios el cambio de su sistema operativo a un menor costo. Por ejemplo, algunas aplicaciones de código abierto tales como Open Office, Mozilla y Firefox siendo aplicaciones compatibles con Windows pueden reducir el costo de cambiar el sistema operativo a GNU/Linux.

- **Seguridad**

GNU/Linux es un software más seguro teniendo 0.17 fallas de seguridad por cada 1,000 líneas de código en su kernel en comparación con el software patentado, el cual tiene de 10 a 20 fallas.

Además de las características antes descritas, la mayor parte de las distribuciones de GNU/Linux no se requiere de realizar la compra de una licencia para su uso. El repertorio de distribuciones es amplio para satisfacer los requerimientos que el usuario necesite y las aplicaciones para cada distribución son de código abierto.

Dadas sus características el sistema operativo GNU/Linux está adquiriendo cada vez más aceptación por parte de los usuarios y empresas. En el caso de las empresas, varias de estas suelen dar preferencia al uso de GNU/Linux (o alguna variante originada de la unión entre Windows y GNU/Linux), y a las aplicaciones de código abierto.

### 3.1.2 ¿Por qué PostgreSQL?

El Sistema de Gestión de Base de Datos PostgreSQL demuestra ser un software de calidad dadas sus características descritas a continuación.

- **Instalación**

Es un programa de código libre, por lo cual el usuario tiene la libertad de descargar el programa de forma gratuita y de poder instalarlo a gran escala. Además, PostgreSQL es multiplataforma, estando disponible para diferentes sistemas operativos.

- **Eficiencia**

Su eficiencia radica en cuanto al manejo de grandes cantidades de información, y en la implementación de tareas más complejas, tales como la ejecución de consultas complejas y frecuentes.

- **Alta disponibilidad**

El sistema puede ser configurado de forma que garantice la protección y disponibilidad de los datos por medio de diferentes métodos de reserva que mantengan una copia de la base de datos.

- **Recursos SQL**

Implementa diferentes tipos de elementos para implementar una base de datos de calidad. Para lo cual este cuenta con diferentes tipos de datos, soporte de vistas, integridad referencial, procedimientos almacenados y demás elementos incorporados por el lenguaje SQL.

- **Soporte**

El soporte con el que dispone está integrado por la documentación ofrecida en la página oficial de PostgreSQL, foros en línea, entre otros medios. Por otro lado, también se proporciona un soporte comercial en caso de que se requiera de un soporte más especializado.

Es importante mencionar que es necesario contar con una base de datos con gran cantidad de datos para que el SGBD tenga un óptimo desempeño. Aun así, PostgreSQL demuestra ser un SGBD estable, potente y eficiente. Además de que es un software de código abierto, el cual permite ser configurado en base a las necesidades del usuario.

### 3.1.3 ¿Por qué CentOS?

Como se mencionó anteriormente el sistema operativo GNU/Linux tiene ciertas características a favor para ser usado como sistema operativo, pero dado que este cuenta con diversas distribuciones, la versión que se ha elegido para el desarrollo de este trabajo es la de CentOS, el cual hace referencia a que es un sistema operativo para la comunidad empresarial, esto por su nombre en inglés (Community Enterprise Operating System).

CentOS es un sistema operativo de código abierto para servidores, el cual puede ser adquirido y usado libremente por los usuarios sin ningún costo. A continuación, se presenta una descripción de sus cualidades comparado con otros sistemas operativos.

- **Seguridad**

CentOS comparado con otros sistemas operativos tiene una menor cantidad de errores o agujeros en su código haciendo que sea menos vulnerable a ataques, además de contar con actualizaciones de seguridad.

- **Estabilidad**

CentOS es estable gracias al soporte que se le da por parte de la comunidad y por parte de Red Hat. Los usuarios hacen uso de foros para compartir soluciones en el mantenimiento y actualización del sistema operativo, además de que puede permanecer en funcionamiento por más tiempo sin requerir de una nueva actualización.

El código utilizado en su desarrollo proviene del código fuente liberado de Red Hat de la distribución RHEL (por sus siglas en inglés Red Hat Enterprise Linux), siendo equivalente a una distribución empresarial, haciendo que CentOS y RHEL sean compatibles. Es así que, de entre las distribuciones antes mencionadas, CentOS demuestra ser uno de los sistemas operativos mejores calificados. Por tal motivo se optó por utilizar esta distribución para el desarrollo de este trabajo.

Además de esto, una nueva versión de CentOS es lanzada aproximadamente cada 2 años. Las diversas versiones son actualizadas de manera regular cada 6 meses, y se les da soporte durante un periodo de 10 años. Considerando por ejemplo la versión de CentOS 7, esta versión fue lanzada el 7 de julio de 2014, por lo cual tendrá soporte hasta junio del 2024.

Entre las distribuciones de Linux más utilizadas está CentOS, Ubuntu y Debian. Debido a sus características, las empresas y organizaciones hacen uso de estos sistemas operativos en sus servidores web.

### 3.2 REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DE LA BASE DE DATOS

En esta sección se muestran las características del equipo utilizado para implementar la base de datos. Adicional a esto se muestra otro equipo, el cual es una recomendación de un servidor de base de datos con características más adecuadas para la aplicación. A continuación, se muestra una imagen de cómo es cada equipo, así como de sus características.



(a) Equipo utilizado



(b) Equipo sugerido

**Figura 3.1** Equipos de cómputo contemplados para implementar la base de datos.

Fuente: <https://www.pinterest.com.mx/><sup>35</sup>

	Equipo utilizado	Equipo sugerido
<b>Elemento</b>	Laptop	Servidor de torre
<b>Marca</b>	Lenovo	DELL
<b>Procesador</b>	AMD	Intel
<b>Disco duro</b>	512 GB	2 TB
<b>Memoria RAM</b>	8 GB	4 ranuras de 128 GB c/u
<b>Sistema operativo</b>	Windows	Opcional
<b>Precio</b>	\$15,000.00 MXN	\$21,499.00 MXN

**Tabla 3.1** Características de las computadoras contempladas para el proyecto.

Fuente: Elaboración propia<sup>36</sup>

<sup>35</sup> Recursos contemplados para implementar la base de datos.

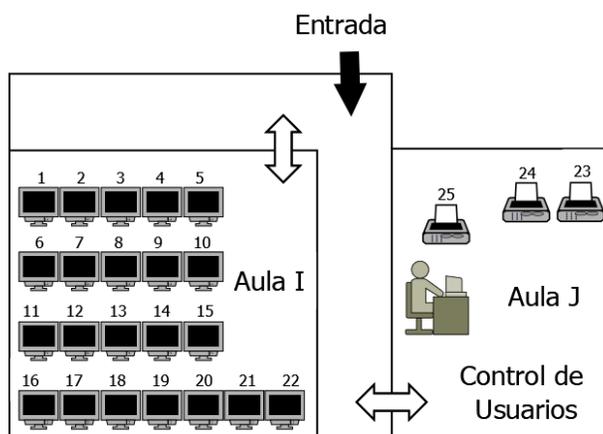
<sup>36</sup> Características de cada uno de los equipos para implementar la base de datos.

### 3.3 DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

#### 3.3.1 Antecedentes para el análisis y diseño

- **Distribución de equipos 2020-2021 en salas UNICA**

*Distribución de equipos en sala 1*



**Figura 3.2 Sala 1 UNICA.**

Fuente: Elaboración propia <sup>37</sup>

Servicio			Software	
Tipo	Cantidad	Ubicación	Tipo	Nombre
Cómputo	22	Aula I	Sistemas operativos	Windows 10
Impresión	3	Aula J		GNU / Linux
			Ofimática	Microsoft Office 2016
			Navegadores	Mozilla
				Chrome
			Aplicaciones básicas	WinRAR
				Acrobat
				Reader
				VLC
			CAD	Microsoft Essentials
				AutoCAD 2016
			IDE	Matlab
				Dev C++
				Visual C++

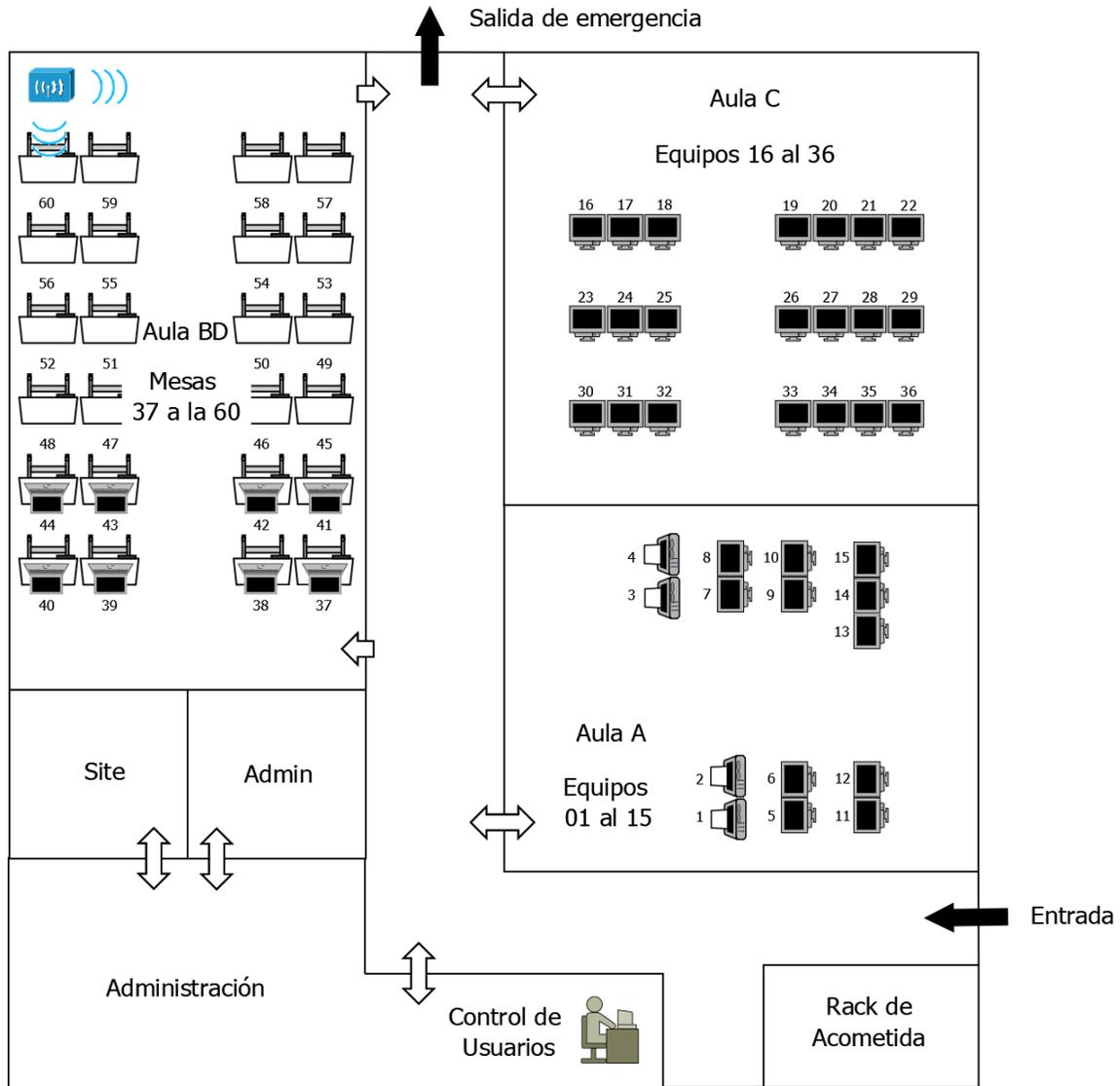
**Tabla 3.2 Recursos Sala 1 UNICA.**

Fuente: Elaboración propia <sup>38</sup>

<sup>37</sup> Distribución de equipos en sala 1.

<sup>38</sup> Recursos disponibles en Sala 1 para uso de los alumnos.

## Distribución de equipos en sala 2



**Figura 3.3 Sala 2 UNICA.**

Fuente: Elaboración propia <sup>39</sup>

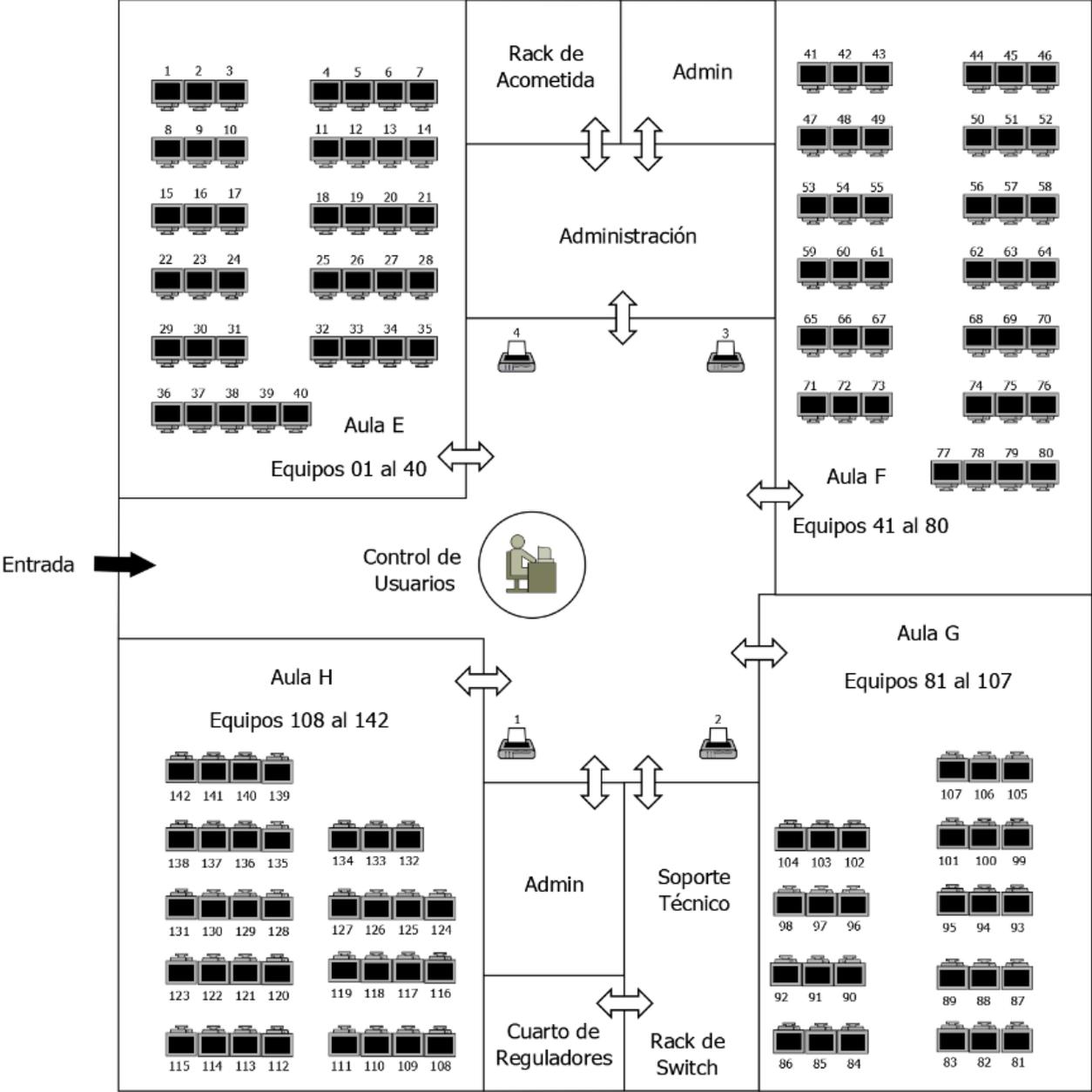
Servicio			Software	
Tipo	Cantidad	Ubicación	Tipo	Nombre
Cómputo	11	Aula A	Sistemas operativos	Windows 10
Impresión	4			GNU / Linux
Usos múltiples	24	Aula BD	Ofimática	Microsoft Office 2016
Cómputo	21	Aula C	Navegadores	Mozilla
				Chrome

<sup>39</sup> Distribución de equipos en sala 2.

IDE	Dev C++
-----	---------

**Tabla 3.3 Recursos Sala 2 UNICA.**  
Fuente: Elaboración propia <sup>40</sup>

*Distribución de equipos en sala 3*



**Figura 3.4 Sala 3 UNICA.**  
Fuente: Elaboración propia <sup>41</sup>

<sup>40</sup> Recursos disponibles en Sala 2 para uso de los alumnos.

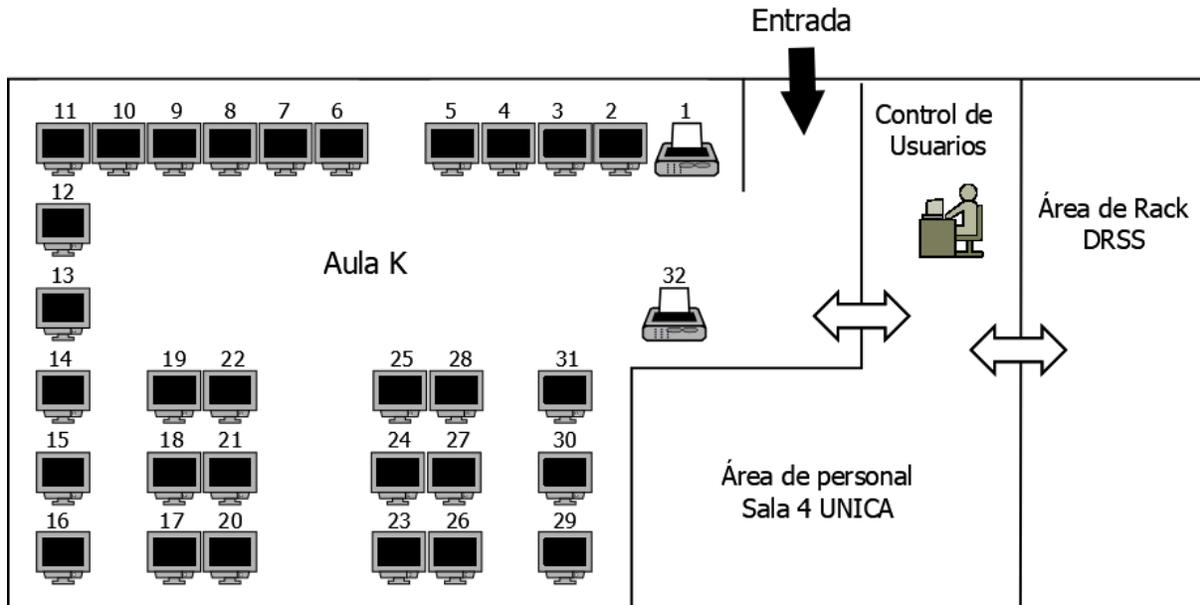
<sup>41</sup> Distribución de equipos en sala 3.

Servicio			Software	
Tipo	Cantidad	Ubicación	Tipo	Nombre
Cómputo	40	Aula E	Sistemas operativos	Windows 10
Cómputo	40	Aula F		GNU / Linux
Cómputo	27	Aula G	Ofimática	Microsoft Office 2016
Cómputo	35	Aula H	Navegadores	Mozilla
				Chrome
Impresión	4	Zona central	IDE	Dev C++

**Tabla 3.4** Recursos Sala 3 UNICA.

Fuente: Elaboración propia <sup>42</sup>

*Distribución de equipos en sala 4*



**Figura 3.5** Sala 4 UNICA.

Fuente: Elaboración propia <sup>43</sup>

Servicio			Software	
Tipo	Cantidad	Ubicación	Tipo	Nombre
Cómputo	30	Aula K	Sistemas Operativos	Windows 10
Impresión	2		GNU / Linux	
			Ofimática	Microsoft Office 2016
			Navegadores	Mozilla

<sup>42</sup> Recursos disponibles en Sala 3 para uso de los alumnos.

<sup>43</sup> Distribución de equipos en sala 4.

	Chrome
CAD	Maple
	AutoCAD 2016
	Matlab
IDE	Dev C++

**Tabla 3.5 Recursos Sala 4 UNICA.**  
Fuente: Elaboración propia <sup>44</sup>

▪ **Elementos de salas UNICA**

Elemento	Detalle	Descripción
Curso	Cursos Intersemestrales	Inscripciones Sala 3.
	De fines de semana	
	Especiales	
	Internos	
	Certificaciones	
Equipo	Computadoras	Son utilizadas para efectuar labores que requieren de algún software o acceso a internet.
	Impresoras	Son utilizadas para realizar impresiones.
	Puertos con acceso a internet	Son utilizados para proporcionar internet a dispositivos externos.
	Escáneres	Son utilizados para realizar escaneos.
Software	Software de sistema	Sistemas operativos.
	Software de desarrollo	Entornos IDE y editores de texto.
	Software de aplicación	Ofimática, simuladores y CAD.

<sup>44</sup> Recursos disponibles en Sala 4 para uso de los alumnos.

Personal	Auxiliar	Apoyo y servicio social.
	Autorizado	Administrador, jefe de sala y jefe de aula.
	Especializado	Soporte técnico e instructores.
Información	Aviso	Se proporciona información en relación a los servicios y actividades de UNICA.
	Reporte	Se proporciona información en relación a eventualidades relacionadas al equipo.
	Información Sala	Se proporciona información sobre ciertos aspectos y elementos de las salas.
	Información aula	Se proporciona información sobre ciertos aspectos y elementos de las aulas.

**Tabla 3.6** Subsistemas relevantes del sistema principal.

Fuente: Elaboración propia <sup>45</sup>

### 3.3.2 Diseño

Para dar inicio al diseño de una base de datos primero se requiere identificar aquella información que es relevante registrar para llevar a cabo la gestión ya sea de una empresa, una organización, o de cualquier otra cosa en la que se tenga que manejar un conjunto de datos.

Por tanto, para realizar el diseño de la base de datos de las salas de cómputo de la Facultad de Ingeniería se requiere identificar los elementos que son relevantes para la gestión de las salas. A continuación, se presenta una descripción de las entidades más relevantes de las salas de cómputo.

Entidad	Descripción
Aula	Es el espacio donde los usuarios acceden para hacer uso del equipo de cómputo.
Aviso	Es un comunicado relacionado con los servicios de las salas de cómputo.
Curso	Es otro servicio que se brinda en las salas de cómputo. Su objetivo es contribuir con la formación académica de los usuarios.

<sup>45</sup> Sistema principal de Salas de Cómputo.

Equipo	Es el equipo de cómputo que se le proporciona a los usuarios para desempeñar sus labores académicas.
Reporte	Es un informe relacionado con cualquier anomalía que afecte el desempeño de un equipo de cómputo.
Sala	Forma parte de las instalaciones de la Facultad, donde cada sala está conformada por un aula como mínimo.
Software	Es toda aplicación que está instalada en el equipo de cómputo, la cual contribuye en el desarrollo de las actividades de los usuarios.
Usuario	Es la persona que forma parte del personal de una sala. Su función es la de realizar ciertas labores que garanticen el buen funcionamiento del equipo y servicios de las salas de cómputo.

**Tabla 3.7** Planteamiento de las entidades de las salas de cómputo.

Fuente: Elaboración propia <sup>46</sup>

Para cada entidad se tiene un conjunto de atributos, cada uno de los cuales describe algo en particular de la entidad a la que pertenece. Es así que a continuación se presentan los atributos más relevantes de cada una de las entidades antes mencionadas.

Entidad	Atributos
Aula	nom_aula (A, BD, C, E, F, G, H, I, J, K), resp_aula, num_sala (1 - 4) y num_equipos.
Aviso	id_aviso, titulo, autor, num_cuenta, fecha_inicio, fecha_fin y descripción.
Curso	id_curso, cartel, nom_curso, modalidad (Presencial o En Línea), fecha_inicio, fecha_fin, dias, horario, duracion, cupo, num_sala y temario.
Equipo	id_equipo, id_inventario, num_equipo, modelo, marca, sistema_op(Windows o Linux), num_sala (1 - 4), nom_aula (A, BD, C, E, F, G, H, I, J, K) y estado (Activo, Revisado o Inactivo).

<sup>46</sup> Entidades establecidas para efectuar un primer diseño de la base de datos.

Reporte	id_reporte, id_maquina, foto, autor, num_cuenta, situacion (Solucionado o Sin solucionar), condicion (Nuevos, Activos o Históricos), num_sala (1 - 4), num_equipo, id_equipo, id_inventario, fecha_reporte, fecha_entrega, motivo y nota.
Sala	num_sala (1 - 4), resp_sala, telefono, control y ubicacion.
Software	id_software, foto, nom_software, clasificacion (Aplicación, CAD, IDE, Navegador, Ofimatica, SO), descripcion e instalacion.
Usuario	num_cuenta, curp, foto, nombre, apellido_pat, apellido_mat, sexo (H o M), telefono, correo, facultad_escuela, carrera, clave_carrera, semestre (1 - 10), num_sala (1 - 4), fecha_registro, clase (Administrador, Jefe de sala, Jefe de aula o Apoyo), condicion (Becario, Servicio social) y estado (Activo o Histórico).

**Tabla 3.8** Atributos asociados a las entidades de las salas de cómputo.

Fuente: Elaboración propia <sup>47</sup>

Dadas las entidades, estas requieren de relaciones de correspondencia que demuestren que dichas entidades tienen un parentesco. A continuación, se presentan las relaciones entre las entidades antes planteadas.

- Las entidades “*usuario*” y “*sala*” están relacionadas por medio de la relación “*encarga*”. Para esto, la cardinalidad de la relación es de uno a uno (1: 1), donde un usuario solamente puede estar a cargo de una de las salas, y una sala solamente puede tener un usuario como encargado de la sala.
- Las entidades “*usuario*” y “*aula*” están relacionadas por medio de la relación “*encarga*”. Para esto, la cardinalidad de la relación es de uno a uno (1: 1), donde un usuario solamente puede estar a cargo de un aula, y un aula solamente puede tener un usuario como encargado del aula.
- Las entidades “*sala*” y “*usuario*” están relacionadas por medio de la relación “*participa*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde un usuario solamente puede dar servicio en una sala en específico, pero en una sala pueden estar múltiples usuarios, no importando que todos ellos tengan asignada la misma sala para dar servicio.

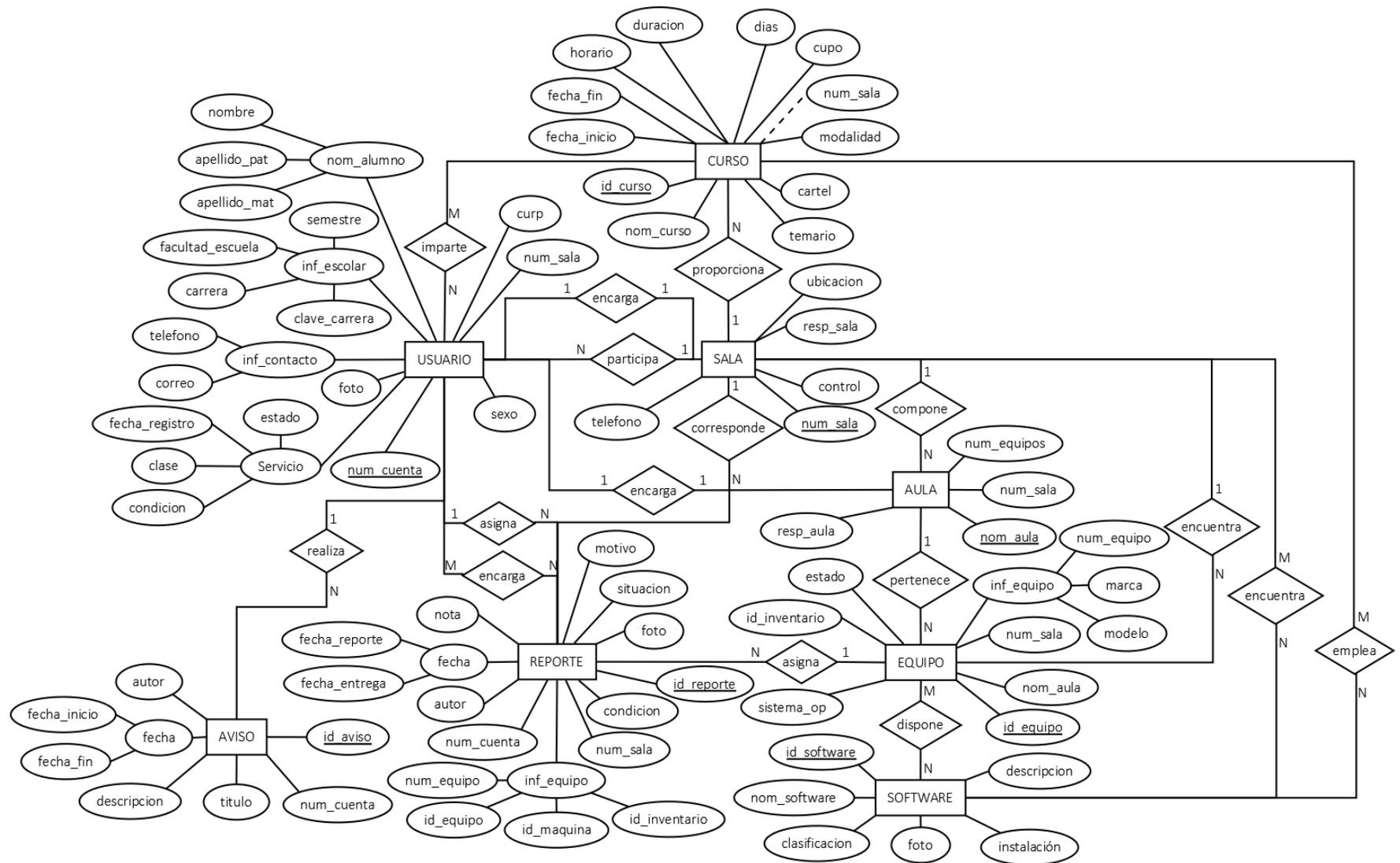
<sup>47</sup> Atributos que proporcionan información relevante sobre las entidades de las salas de cómputo.

- Las entidades “*usuario*” y “*aviso*” están relacionadas por medio de la relación “*realiza*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde un usuario puede ser el autor de múltiples avisos, pero un aviso puede ser escrito solamente por un usuario.
- Las entidades “*usuario*” y “*reporte*” están relacionadas por medio de la relación “*asigna*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde un usuario puede ser el autor de múltiples reportes, pero un reporte puede ser escrito solamente por un usuario.
- Las entidades “*sala*” y “*curso*” están relacionadas por medio de la relación “*proporciona*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde cada curso será proporcionado dentro de una sola sala (si es que el curso es presencial), pero una sala puede ser empleada para impartir más de un curso, ya sea que esta disponga de múltiples aulas para impartir los cursos, o que los cursos sean impartidos en diferentes horarios.
- Las entidades “*sala*” y “*reporte*” están relacionadas por medio de la relación “*corresponde*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde una sala puede tener asignado más de un reporte, pero un reporte solo puede tener registrada una sala, haciendo referencia a la sala en la cual fue creado el reporte.
- Las entidades “*equipo*” y “*reporte*” están relacionadas por medio de la relación “*asigna*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde un equipo puede tener asignado más de un reporte, pero un reporte solamente es asignado para un solo equipo.
- Las entidades “*sala*” y “*aula*” están relacionadas por medio de la relación “*compone*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde una sala puede estar compuesta por más de un aula, pero un aula forma parte de una sola sala.
- Las entidades “*sala*” y “*equipo*” están relacionadas por medio de la relación “*encuentra*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde una sala dispone de múltiples equipos, pero un equipo pertenece a una sola sala.
- Las entidades “*aula*” y “*equipo*” están relacionadas por medio de la relación “*pertenece*”. Para esto, la cardinalidad de la relación es de uno a muchos (1: N), donde un aula dispone de múltiples equipos, pero un equipo pertenece a una sola aula.

- Las entidades “*software*” y “*curso*” están relacionadas por medio de la relación “*emplea*”. Para esto, la cardinalidad de la relación es de muchos a muchos (M: N), donde un software puede formar parte de más de un curso, a su vez para que un curso sea impartido este puede requerir más de un software.
- Las entidades “*software*” y “*equipo*” están relacionadas por medio de la relación “*dispone*”. Para esto, la cardinalidad de la relación es de muchos a muchos (M: N), donde un software puede estar presente en más de un equipo, a su vez un equipo dispone de múltiples softwares.
- Las entidades “*software*” y “*sala*” están relacionadas por medio de la relación “*encuentra*”. Para esto, la cardinalidad de la relación es de muchos a muchos (M: N), donde un software puede estar presente en más de una sala, a su vez en una sala se encuentran múltiples softwares.
- Las entidades “*usuario*” y “*curso*” están relacionadas por medio de la relación “*imparte*”. Para esto, la cardinalidad de la relación es de muchos a muchos (M: N), donde un curso puede ser impartido por múltiples usuarios, a su vez un usuario puede impartir más de un curso.
- Las entidades “*usuario*” y “*reporte*” están relacionadas por medio de la relación “*encarga*”. Para esto, la cardinalidad de la relación es de muchos a muchos (M: N), donde un usuario puede estar encargado de atender múltiples reportes, a su vez un reporte puede ser atendido por más de un usuario.

Por medio de las entidades, los atributos y las relaciones es posible realizar el diseño conceptual de la base de datos por medio de un modelo entidad relación, tal como se muestra en la Figura 3.6.

De igual forma, el diseño de la base de datos puede ser representado por medio de un diagrama de esquema, el cual es otra forma de expresar la estructura de la base de datos. En la Figura 3.7 se muestra el diagrama de esquema de la base de datos de las salas de cómputo, en el cual cada entidad y sus atributos se representan por medio de una tabla. De igual forma las relaciones con cardinalidad muchos a muchos son representadas por medio de tablas.



**Figura 3.6** Modelo ER de las salas de cómputo de UNICA.

Fuente: Elaboración propia <sup>48</sup>

<sup>48</sup> Modelo ER en el que se relaciona información sobre el personal y los recursos de las salas de cómputo de UNICA.

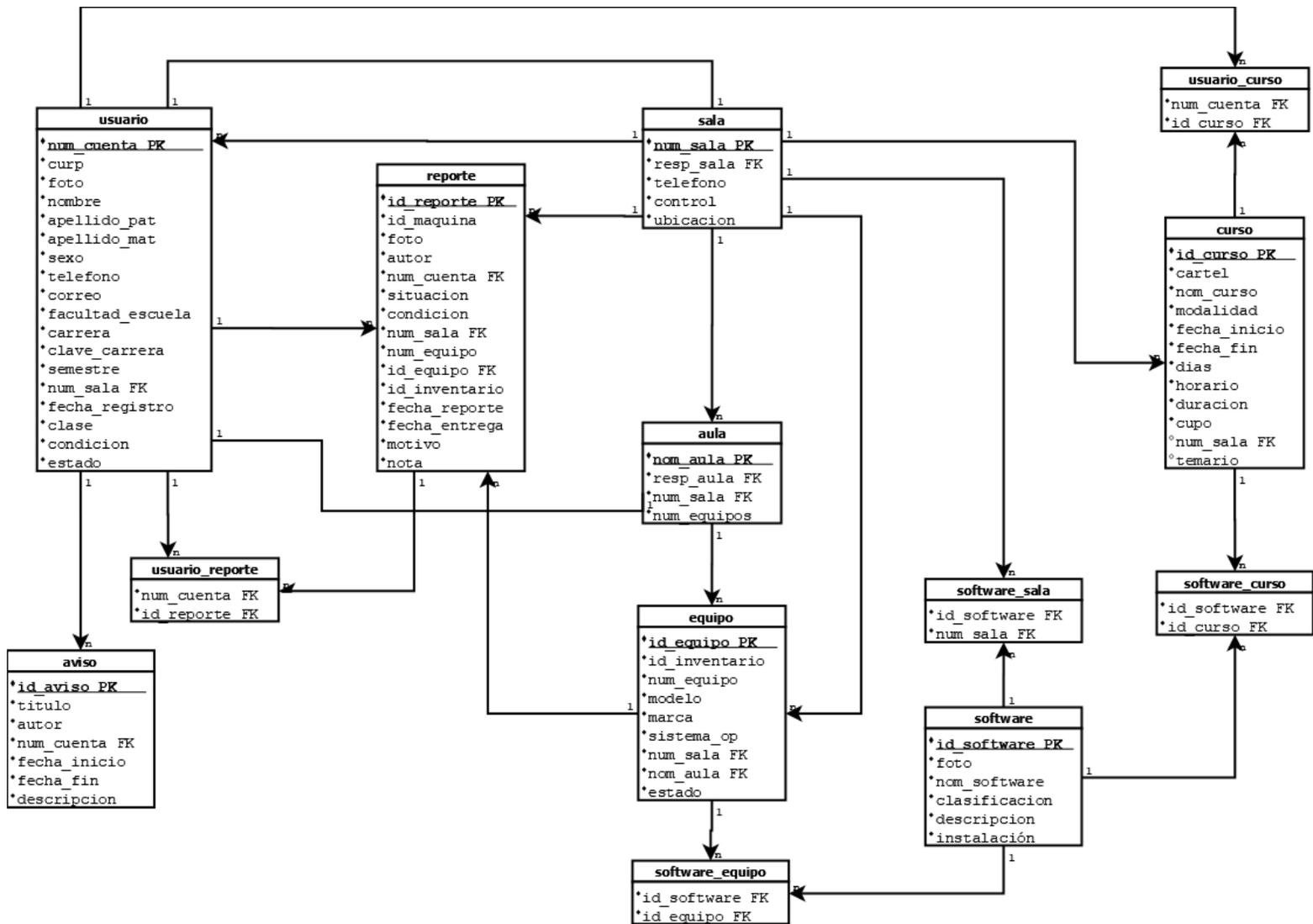


Figura 3.7 Diagrama de esquema de la base de datos UNICA.

Fuente: Elaboración propia <sup>49</sup>

<sup>49</sup> Diagrama de la base de datos UNICA.

Cabe mencionar que, en el diagrama de la figura anterior, las entidades son representadas por medio de tablas, por lo que las tablas que se tienen en el diagrama son aula, aviso, curso, equipo, reporte, sala, software y usuario. En este caso, el nombre de cada tabla es el mismo de la entidad que representa.

En el caso de las relaciones con cardinalidad muchos a muchos, el nombre de cada una de las tablas es: software\_curso, software\_equipo, software\_sala, usuario\_curso y usuario\_reporte. En este caso el nombre de cada tabla está dado por el nombre de las dos entidades unidas por la relación.

- **Diccionario de datos de las tablas de la base de datos**

En esta sección se define un diccionario de datos para cada una de las tablas de la base de datos. Para esto, los datos considerados en el diccionario de datos están los nombres de cada atributo, el dominio que tendrá el atributo según el tipo de dato a almacenar, la restricción del atributo (pudiendo tener más de una restricción), y una descripción sobre la información que almacena el atributo. Los diccionarios de datos que describen cada una de las tablas de la base de datos son los que se muestran a continuación.

*Tabla aula*

Columna	Tipo	Restricción	Descripción
nom_aula	varchar (2)	primary key	Clave primaria del aula
resp_aula	numeric (9)	Unique not null	Número de cuenta del usuario responsable del aula
num_sala	numeric (1)	not null	Sala donde se encuentra ubicada el aula
numEquipos	numeric (2)	not null	Número de computadoras dentro del aula

**Tabla 3.9** Diccionario de datos de la tabla aula.

Fuente: Elaboración propia <sup>50</sup>

<sup>50</sup> Diccionario de datos que describe la tabla aula.

*Tabla aviso*

Columna	Tipo	Restricción	Descripción
id_aviso	numeric (3)	primary key	Clave primaria del aviso
titulo	varchar (50)	not null	Título del aviso
autor	varchar (75)	default current_user	Nombre del usuario que es autor del aviso
num_cuenta	numeric (9)	not null	Número de cuenta del usuario que es autor del aviso
fecha_inicio	date	default current_date	Fecha de emisión del aviso
fecha_fin	date	not null	Fecha de expiración del aviso
descripción	text	not null	Texto correspondiente al aviso

**Tabla 3.10** Diccionario de datos de la tabla aviso.

Fuente: Elaboración propia <sup>51</sup>

*Tabla curso*

Columna	Tipo	Restricción	Descripción
id_curso	numeric (2)	primary key	Clave primaria del curso
cartel	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al curso
nom_curso	varchar(100)	not null	Nombre del curso
modalidad	varchar (10)	not null	Modo de impartición
fecha_inicio	date	not null	Fecha en la que inicia el curso
fecha_fin	date	not null	Fecha en la que acaba el curso
días	varchar (30)	not null	Días en los que se imparte el curso
horario	varchar (17)	not null	Horario de las clases

<sup>51</sup> Diccionario de datos que describe la tabla aviso.

duracion	numeric (3)	not null	Número de horas
cupo	numeric (3)	not null	Número de lugares disponibles
num_sala	numeric (1)	default null	Sala designada para el curso
temario	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al temario del curso

**Tabla 3.11** Diccionario de datos de la tabla curso.

Fuente: Elaboración propia <sup>52</sup>

### Tabla equipo

Columna	Tipo	Restricción	Descripción
id_equipo	numeric (3)	primary key	Clave primaria del equipo
Id_inventario	numeric (3)	not null unique	Identificador del equipo como inventario
num_equipo	numeric (3)	not null	Número de equipo
modelo	varchar (10)	not null	Modelo del equipo
marca	varchar (25)	not null	Marca del equipo
sistema_op	varchar (7)	not null	Sistema operativo del equipo
num_sala	numeric (1)	not null	Sala donde se encuentra ubicado el equipo
nom_aula	varchar (2)	not null	Aula donde se encuentra ubicado el equipo
estado	varchar (8)	not null	Especifica si el equipo está activo, si requiere revisión (pero sigue activo), o si queda inactivo

**Tabla 3.12** Diccionario de datos de la tabla equipo.

Fuente: Elaboración propia <sup>53</sup>

<sup>52</sup> Diccionario de datos que describe la tabla CURSO.

<sup>53</sup> Diccionario de datos que describe la tabla equipo.

*Tabla reporte*

Columna	Tipo	Restricción	Descripción
id_reporte	numeric (3)	primary key	Clave primaria del reporte
id_maquina	varchar(120)	not null unique	Identificador del equipo como máquina
foto	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al reporte
autor	varchar (75)	default current_user	Nombre del usuario que ha generado el reporte
num_cuenta	numeric (9)	not null	Número de cuenta del usuario que ha generado el reporte
situacion	varchar (14)	not null	Especifica si el reporte ha sido solucionado
condicion	varchar (9)	not null	Especifica la antigüedad del reporte
num_sala	numeric (1)	not null	Sala donde se encuentra ubicado el equipo
num_equipo	numeric (3)	not null	Número de equipo
id_equipo	numeric (3)	not null	Clave primaria del equipo
id_inventario	numeric (3)	not null	Identificador del equipo como inventario
fecha_reporte	Date	default current_date	Fecha de emisión del reporte
fecha_entrega	Date	default null	Fecha límite para atender el reporte
motivo	Text	not null	Motivo del reporte
nota	Text	not null	Descripción detallada del reporte

**Tabla 3.13** *Diccionario de datos de la tabla reporte.*

Fuente: Elaboración propia <sup>54</sup>

<sup>54</sup> Diccionario de datos que describe la tabla reporte.

### Tabla sala

Columna	Tipo	Restricción	Descripción
num_sala	numeric (1)	primary key	Clave primaria de la sala
resp_sala	numeric (9)	not null unique	Número de cuenta del usuario responsable de la sala
Telefono	numeric (10)	not null	Número telefónico de la sala
Control	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al horario de los usuarios que atienden la sala
Ubicación	Text	not null	Información sobre la ubicación de la sala

**Tabla 3.14** Diccionario de datos de la tabla sala.

Fuente: Elaboración propia <sup>55</sup>

### Tabla software

Columna	Tipo	Restricción	Descripción
id_software	numeric (2)	primary key	Clave primaria del software
foto	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al software
nom_software	varchar (25)	not null	Nombre del software
clasificacion	varchar(17)	not null	Clasificación del software
descripcion	Text	not null	Descripción del software
instalacion	Text	not null	Proceso de instalación

**Tabla 3.15** Diccionario de datos de la tabla software.

Fuente: Elaboración propia <sup>56</sup>

### Tabla software\_curso

Columna	Tipo	Restricción	Descripción
id_software	numeric (2)	not null	Identificador del software
id_curso	numeric (2)	not null	Identificador del curso

**Tabla 3.16** Diccionario de datos de la tabla software\_curso.

Fuente: Elaboración propia <sup>57</sup>

<sup>55</sup> Diccionario de datos que describe la tabla sala.

<sup>56</sup> Diccionario de datos que describe la tabla software.

<sup>57</sup> Diccionario de datos que describe la tabla software\_curso.

### Tabla software\_equipo

Columna	Tipo	Restricción	Descripción
id_software	numeric (2)	not null	Identificador del software
id_equipo	numeric (3)	not null	Identificador del equipo

**Tabla 3.17** Diccionario de datos de la tabla software\_equipo.

Fuente: Elaboración propia <sup>58</sup>

### Tabla software\_sala

Columna	Tipo	Restricción	Descripción
id_software	numeric (2)	not null	Identificador del software
num_sala	numeric (1)	not null	Número de sala

**Tabla 3.18** Diccionario de datos de la tabla software\_sala.

Fuente: Elaboración propia <sup>59</sup>

### Tabla usuario

Columna	Tipo	Restricción	Descripción
num_cuenta	numeric (9)	primary key	Número de cuenta del usuario
curp	char (18)	not null	CURP del usuario
foto	varchar(120)	default null	Dirección donde se aloja la foto correspondiente al usuario
nombre	varchar (25)	not null	Nombre del usuario
apellido_pat	varchar (25)	not null	Apellido paterno del usuario
apellido_mat	varchar (25)	not null	Apellido materno del usuario
sexo	Char	not null	Género del usuario
telefono	numeric (10)	not null	Número telefónico del usuario
correo	varchar (50)	not null	Correo electrónico del usuario
facultad_escuela	varchar (50)	not null	Facultad o escuela de procedencia

<sup>58</sup> Diccionario de datos que describe la tabla software\_equipo.

<sup>59</sup> Diccionario de datos que describe la tabla software\_sala.

carrera	varchar (35)	not null	Carrera correspondiente al usuario
clave_carrera	numeric (3)	not null	Clave correspondiente a la carrera
semestre	numeric (2)	not null	Semestre que cursa el usuario
num_sala	numeric (1)	default null	Sala donde el usuario labora
fecha_registro	Date	not null	Fecha de registro del usuario
clase	varchar (15)	not null	Especifica el role del usuario
condicion	varchar (15)	not null	Especifica el tipo de relación de los usuarios dentro de la sala
estado	varchar (9)	not null	Especifica si el usuario está vigente en el sistema

**Tabla 3.19** Diccionario de datos de la tabla usuario.

Fuente: Elaboración propia <sup>60</sup>

#### Tabla usuario\_curso

Columna	Tipo	Restricción	Descripción
num_cuenta	numeric (9)	not null	Número de cuenta del usuario designado instructor del curso
id_curso	numeric (2)	not null	Identificador del curso

**Tabla 3.20** Diccionario de datos de la tabla usuario\_curso.

Fuente: Elaboración propia <sup>61</sup>

#### Tabla usuario\_reporte

Columna	Tipo	Restricción	Descripción
num_cuenta	numeric (9)	not null	Número de cuenta del usuario responsable del reporte
id_reporte	numeric (3)	not null	Identificador del reporte

**Tabla 3.21** Diccionario de datos de la tabla usuario\_reporte.

Fuente: Elaboración propia <sup>62</sup>

<sup>60</sup> Diccionario de datos que describe la tabla usuario.

<sup>61</sup> Diccionario de datos que describe la tabla usuario\_curso.

<sup>62</sup> Diccionario de datos que describe la tabla usuario\_reporte.

### 3.3.3 Instalación y configuración

Entre los requisitos planteados para poder implementar la base de datos están el uso de un servidor que tenga instalado PostgreSQL como manejador de base de datos, y cuyo sistema operativo sea CentOS versión 7.

Para cumplir con estos requisitos es necesario el uso de una máquina virtual, ya que por medio de esta será posible disponer del sistema operativo CentOS en una computadora con sistema operativo Windows. Para esto se utilizará VirtualBox, programa disponible en internet para la creación de máquinas virtuales, de modo que el equipo emulará ser otra computadora con su propio sistema operativo independiente del sistema operativo original.

La descripción de cómo crear una máquina virtual utilizando VirtualBox se encuentra en el apéndice A, en la sección *“Creación de la máquina virtual”*.

Una vez creada la máquina virtual, esta se debe poner en marcha para dar lugar a la instalación de CentOS 7 sobre la máquina creada. La descripción de cómo instalar el sistema operativo CentOS 7 se encuentra en el apéndice A, en la sección *“Instalación de CentOS 7”*.

Después de que el sistema operativo CentOS 7 ha sido instalado, para implementar la base de datos es necesaria la instalación de un manejador de base de datos. Es así que el manejador que se utilizará para la base de datos será PostgreSQL. La descripción de cómo instalar PostgreSQL se encuentra en el apéndice A, en la sección *“Instalación de PostgreSQL”*.

Tomando en cuenta que la interfaz de usuario del sistema de base de datos es una página web, la máquina virtual para poder establecer una conexión de red habrá que configurar los parámetros de red destinados para su conexión. La descripción de cómo realizar la configuración de red se encuentra en el apéndice A, en la sección *“Configuración de red”*.

Se requiere además de un servidor web cuya función sea la de actuar como un mediador que le proporcione al usuario el sitio web junto con la información contenida en el servidor de base de datos. Para tal objetivo se realiza la instalación del servidor Apache, la descripción de su instalación se encuentra en el apéndice A, en la sección *“Instalación de Apache”*.

Entre los recursos para implementar una página web está el uso de lenguaje PHP, formando parte del código junto con el lenguaje HTML. Entre las utilidades sobre el uso de PHP en una página web está la conexión y consultas con la base de datos, y el poder guardar secciones de código de la página en archivos separados, y pudiendo establecer un vínculo entre los mismos. La descripción de cómo realizar la instalación de PHP se encuentra en el apéndice A, en la sección *“Instalación de PHP”*.

### 3.3.4 Implementación

- **Inicio de sesión dentro del sistema operativo**

Inicialmente se ejecuta la máquina virtual creada en VirtualBox, esto hace que el programa abra una nueva ventana correspondiente a la interfaz para que el usuario pueda interactuar con el sistema. Una vez que la máquina virtual ha cargado, se muestra una pantalla de inicio, tal como se muestra a continuación.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.71.1.el7.x86_64 on an x86_64

localhost login: admin
Password:
Last login: Mon Jul 18 14:16:18 on tty1
[admin@localhost ~]# su -
Contraseña:
```

**Figura 3.8** Comandos iniciales para iniciar sesión en el sistema operativo.

Fuente: Elaboración propia <sup>63</sup>

Inicialmente, el sistema solicita ingresar el nombre de usuario y la contraseña, para lo cual se ingresa el nombre de usuario y contraseña creados por el usuario durante la instalación de CentOS 7. Las líneas de la figura anterior en las que se inicia sesión son:

```
localhost login: admin
Password:
```

Para continuar, se inicia sesión como super usuario y se ingresa la contraseña correspondiente, este es un requisito por parte del sistema para acceder a PostgreSQL. Las líneas de la figura anterior en las que se inicia sesión como super usuario son:

```
[admin@localhost ~]# su -
Contraseña:
```

- **Inicio de sesión en PostgreSQL**

El comando para acceder a la terminal de PostgreSQL como usuario postgres es (para lo cual se utilizan los comandos -U para el nombre de usuario y -i para inicializar psql):

```
[root@localhost ~]# sudo -u postgres -i psql
```

- **Creación de la base de datos**

Se utiliza la sentencia SQL para crear la base de datos de las salas de cómputo con el nombre de “bdunica” por medio de la siguiente instrucción.

---

<sup>63</sup> Comandos previos a la creación de la base de datos.

**CREATE DATABASE *bdunica*;**

- **Acceso a la base de datos**

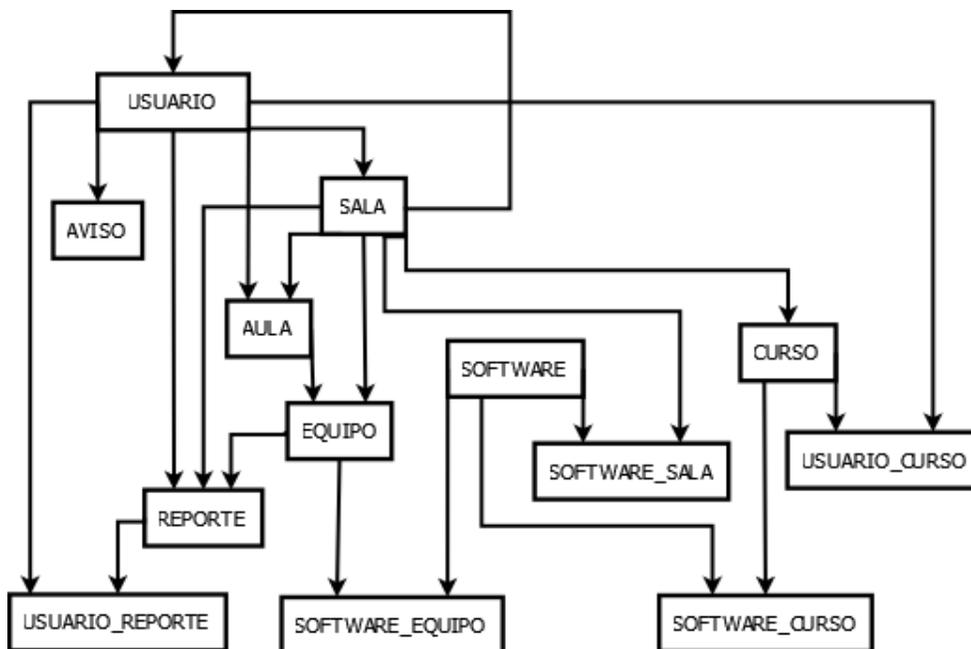
Para acceder a la base de datos “*bdunica*” se utiliza un comando propio de PostgreSQL. El comando es el siguiente:

`\c bdunica`

- **Creación de las tablas de la base de datos**

La creación de las tablas de la base de datos se realiza haciendo uso de sentencias SQL por medio de la instrucción CREATE TABLE. El código de cada una de las tablas se encuentra en el apéndice B, en la sección “*Creación de las tablas de la base de datos*”.

Sin embargo, al revisar el desarrollo de la base de datos se puede observar que las tablas en su mayoría cuentan con una restricción de clave foránea en su código. Debido a esto, para poder ejecutar el código, es necesario tomar en cuenta la jerarquía que existe entre las tablas, esto es, la jerarquía en base a su relación padre hijo debido a la restricción de clave foránea (tal como se muestra en la siguiente figura), esto para evitar que el sistema genere un error impidiendo la ejecución del código.



**Figura 3.9** Relación padre hijo entre las tablas de la base de datos.

Fuente: Elaboración propia <sup>64</sup>

<sup>64</sup> Representación de la relación padre hijo de cada una de las tablas de la base de datos UNICA.

Es así que, el orden de ejecución del código de cada tabla se realiza empezando con aquellas tablas que son el origen de la relación padre hijo como los primeros padres, seguido de esto se continúa con las tablas correspondientes al siguiente nivel de inferioridad con relación de hijos, y así sucesivamente hasta llegar a aquellas tablas que no tengan hijos. Por tanto, la creación de las tablas se realizó en orden de usuario, sala, aviso, aula, equipo, reporte, usuario\_reporte, software, software\_equipo, software\_sala, curso, software\_curso, usuario\_curso.

La siguiente línea es un ejemplo de una restricción de clave foránea, la cual establece una relación padre hijo entre las tablas sala y usuario (siendo sala la tabla padre y usuario la tabla hijo).

```
FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL;
```

Dado que la primera tabla en ser creada es la tabla usuario, esta restricción se debe omitir del código de la tabla usuario, esto para evitar hacer referencia a una tabla que no existe generando error en la ejecución. Debido a esto, esta restricción fue establecida por medio de la siguiente línea (siendo necesaria la creación de la tabla sala antes de su ejecución).

```
ALTER TABLE usuario ADD FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL;
```

Esta sentencia no requiere de tener que asignar un nombre a la restricción, sino que el sistema asigna un nombre por defecto (el nombre de la restricción es “*usuario\_num\_sala\_fkey*”). Es importante tener en cuenta el nombre de la restricción en caso de que esta requiera ser eliminada o modificada.

#### ▪ **Creación de roles y usuarios**

La forma de acceso a la base de datos antes planteada implica un inicio de sesión como super usuario (que a su vez permite acceder a la base de datos mediante la cuenta de usuario “*postgres*” sin necesidad de ingresar una contraseña). Esto implica que tal usuario cuenta con todos los permisos disponibles de PostgreSQL reduciendo la seguridad del sistema. Debido a esto, se opta por remover el acceso al sistema al usuario “*postgres*” y quitarle el privilegio de super usuario, no sin antes conceder el permiso de super usuario a un nuevo usuario, el cual será el nuevo administrador global de la base de datos. Las sentencias SQL para esto son las que se muestran a continuación.

La sentencia SQL para la creación de un usuario con password y privilegio de super usuario es:

```
CREATE USER admin WITH SUPERUSER PASSWORD ‘3175’;
```

La sentencia SQL para retirar el privilegio de super usuario postgres es la que se muestra a continuación. (ya que tal privilegio permite al usuario postgres conectarse a la base de datos aun cuando se le hayan retirado los privilegios de conexión).

**ALTER USER *postgres* WITH NOTSUPERUSER;**

Se ejecuta la siguiente sentencia que da como resultado impedir que el usuario “*postgres*” pueda acceder a la base de datos “*bdunica*”.

**REVOKE CONNECT ON DATABASE *bdunica* FROM *postgres*;**

Aun así, el ingreso a la base de datos “*postgres*” al usuario “*postgres*” le permite ingresar a la base de datos “*bdunica*”. Debido a esto se elimina el privilegio que le permite tal acceso al usuario “*postgres*”. La sentencia para eliminar este privilegio es la que se muestra a continuación.

**REVOKE CONNECT ON DATABASE *postgres* FROM PUBLIC;**

En este caso la sentencia anterior incluye la palabra PUBLIC, por lo que ningún usuario podrá acceder a la base de datos “*postgres*” (a excepción de los usuarios con privilegio de superusuario. No es posible retirarle el acceso exclusivamente al usuario “*postgres*” debido a este privilegio). Por otro lado, los usuarios que ya no tengan acceso a la base de datos “*postgres*” aun así podrán acceder a la base de datos “*bdunica*”.

Los usuarios que requieran acceder a la base de datos desde la terminal de postgres lo pueden hacer por medio de la sentencia que se muestra a continuación.

**[admin@localhost ~]\$ psql -h localhost -p 5432 -U rafael -W -d bdunica;**  
**Contraseña:**

Como se puede observar, esta forma de acceder a la base de datos se compone de un conjunto de comandos. Por lo general estos comandos requieren estar acompañados de un valor que en conjunto permiten establecer la conexión con la base de datos.

El significado de cada uno de los comandos es:

- h**: Comando para identificar el nombre del host de la máquina en la que se ejecuta el servidor.
- p**: Comando para identificar el puerto a utilizar para establecer la conexión.
- U**: Comando para identificar el nombre de usuario a utilizar para acceder a la base de datos.
- W**: Comando que solicita al usuario de una contraseña para poder iniciar sesión.
- d**: Comando para identificar el nombre de la base de datos a la cual se desea acceder.

Dado que los usuarios están clasificados en base a las funciones que desempeñan dentro de las salas, las operaciones que puedan realizar dentro de la base de datos están definidas en base al tipo de role que les haya sido asignado a los usuarios.

La creación de los roles de usuario se realiza por medio de sentencias SQL con la instrucción CREATE ROLE seguido del nombre del role. Una vez creados los roles, las sentencias para proporcionar privilegios a estos roles se realizan mediante la instrucción GRANT, seguido de la acción permitida para su uso y del nombre de los roles a los que se les desea proporcionar los permisos. En el apéndice B, en la sección “*Creación de roles y usuarios*” se muestran las sentencias SQL para la creación de roles y la asignación de privilegios.

Después de esto, la creación de usuarios permite que cada usuario disponga de una cuenta personal para acceder a la base de datos, esto se realiza por medio de la instrucción CREATE USER seguido del nombre de usuario. Además, se utilizan las palabras clave WITH PASSWORD para establecer una contraseña que será necesaria para poder iniciar sesión.

Los roles disponen de privilegios por lo que, al asignar un role a un usuario el usuario adquiere en automático los privilegios del role asignado. Esto se realiza mediante la instrucción GRANT, para lo cual se debe indicar primero el nombre del role y después el nombre del usuario.

En el apéndice B, en la sección “*Creación de roles y usuarios*” se muestran las sentencias SQL para la creación de usuarios y la asignación de los roles a los usuarios.

### **3.3.5 Población de la base de datos**

Para ingresar los datos a la base de datos fue necesario definir los datos para cada tabla. Una vez teniendo los datos estos fueron almacenados por medio de tablas de Excel. Para lo cual, las tablas debían estar almacenadas de forma individual, por lo que se crearon múltiples documentos (uno por cada tabla). Adicional a esto, fue necesario que cada documento de Excel fuese guardado como un documento CSV (esto mediante la opción “*guardar como*” y seleccionando el formato “*CSV (delimitado por comas) (\*.csv)*”).

La población de la base de datos se realiza mediante PgAdmin (programa que dispone de múltiples recursos permitiendo la interacción con la base de datos de forma gráfica). Para este proceso se utilizó la interfaz de línea de comandos la cual permite la ejecución de sentencias desde PgAdmin. La sentencia utilizada copia los datos de cada uno de los archivos CSV y los importa sobre cada una de las tablas de la base de datos. El aspecto de este tipo de sentencias es el que se muestra a continuación.

```
\COPY usuario FROM 'C:\CSV_Postgres\usuario_1.csv' DELIMITER ',' CSV HEADER;
```

Los elementos que constituyen esta sentencia son:

**\COPY** - Metacomando PostgreSQL para importar o exportar datos seguido del nombre de la tabla de la base de datos implicada para tal acción. Las columnas de la tabla del archivo deberán coincidir con las columnas de la tabla de la base de datos (en número y orden). De lo contrario en la sentencia se deberá especificar el nombre de las columnas de la tabla de la base de datos con las que se hará la operación (entre paréntesis y separadas por comas) a un lado del nombre de la tabla.

**FROM** – Palabra clave para indicar la dirección donde se aloja el archivo, así como el nombre del archivo. La palabra FROM hace referencia a que el archivo es utilizado para la importación de datos.

**DELIMITER** – Se utiliza para especificar el caracter con el cual serán delimitados los datos para poder ser exportados. Al ser un archivo con formato CSV el caracter que se debe especificar es una coma.

**CSV HEADER** – Palabras clave para indicar que el formato del archivo es CSV y que el archivo contiene un encabezado respectivamente.

Debido a la relación padre hijo entre las tablas el orden de inserción de los datos para cada tabla es el mismo en el que fueron creadas. Esto fue en un orden de usuario, sala, aviso, aula, equipo, reporte, usuario\_reporte, software, software\_equipo, software\_sala, curso, software\_curso, usuario\_curso.

Cabe mencionar que la tabla sala es padre de la tabla usuario (una sala puede tener múltiples usuarios asignados para laborar solamente en esa sala). A su vez la tabla usuario es padre de la tabla sala (algunos usuarios tienen asignada una sala de la cual son responsables). Por tanto, primero se ingresan los datos de los usuarios responsables de cada sala sin un valor asignado en la posición correspondiente a la sala designada para laborar (hijos sin un padre). Para lo cual se ejecuta una sentencia para cargar los datos del archivo CSV de nombre *“usuario\_1”*.

Las entidades de reporte y aviso de caracterizan por emplear en algunas de sus columnas como valor por defecto el usuario actual o la fecha actual mediante la restricción de columna. De esta forma la inserción de los datos podría realizarse asignando valores nulos para tales columnas, y dar como resultado asignar por valor la fecha o el usuario por defecto.

Una vez que fueron registrados los usuarios responsables de cada sala fue posible ejecutar una nueva sentencia para registrar los datos de la tabla sala junto con el número de cuenta del usuario responsable de cada sala.

Por último, para poder asignar una sala a los usuarios responsables de sala se utilizó una sentencia de actualización, tal como se muestra a continuación.

```
UPDATE usuario SET num_sala = 1 WHERE num_cuenta = 362442788;
```

Seguido de esto, se procedió a cargar el resto de los datos de las otras tablas en el orden mencionado anteriormente. Puesto que ya se encontraban definidos los datos de las salas, se ejecuta otra sentencia con la cual puedan registrarse los demás usuarios mediante un archivo CSV de nombre “*usuario\_2*”, en el cual los usuarios ya cuentan con una sala definida donde laborar.

### **3.3.6 Diseño e implementación de la interfaz de usuario**

Los usuarios requieren de una interfaz que les permita visualizar las actividades, servicios y recursos de las salas de forma sencilla.

Debido a que la creación de páginas web no es el objetivo del presente proyecto, se mencionarán algunos aspectos relevantes del diseño de la página web del presente proyecto.

Para iniciar con el diseño de la página web es necesario definir la información que contendrá la página web. Sin embargo, también es importante mencionar que el grupo al que está dirigida la página es para estudiantes de la facultad y para el personal de las salas de cómputo. Con esto en mente puede considerarse que cierta información puede ser de interés para estudiantes, mientras que habrá otro tipo de información que deba mantenerse para uso exclusivo que solamente pueda ser consultado por el personal de las salas.

La vista inicial de la página será con la información de avisos de salas (estando disponible para el público). De igual forma, la página tendrá otros apartados de acceso libre a los usuarios como son salas, software y cursos, los cuales tendrán información de salas, software y cursos respectivamente.

Para los usuarios que entran en la categoría de estudiantes se crea al usuario persona, para el cual se le otorga el privilegio de consulta sobre cada una de las tablas correspondientes a los diferentes apartados a los cuales tiene acceso.

Adicional a esto, la página tendrá una interfaz privada, la cual será de uso exclusivo para usuarios que formen parte del personal de las salas de cómputo. Se podrá acceder a dicha interfaz por medio del apartado sesión, donde aquellos usuarios que estén dados de alta en el sistema PostgreSQL podrán iniciar sesión ingresando un nombre de usuario y una contraseña.

La interfaz privada tendrá un acceso global a la información de la base de datos. Es decir que la información a la que tendrá acceso será referente a los equipos, reportes y usuarios (exclusivo de la interfaz privada), e información de sala, software y cursos (información pública).

El acceso a la información en la interfaz privada estará delimitado aún para los usuarios que forman parte del personal de las salas, pues dependiendo del cargo designado al usuario, el acceso de cada uno de los apartados podrá estar permitido o restringido en base al role designado.

A continuación, se enlistan los apartados a los cuales tienen permitido acceder el personal de salas según el role asignado.

*“administrador”*: Avisos, cursos, equipos, reportes, salas, aulas, software, usuarios.

*“jefe\_de\_sala”*: Avisos, cursos, equipos, reportes, salas, aulas, software.

*“jefe\_de\_aula”*: Avisos, cursos, equipos, salas, aulas, software.

*“apoyo”*: Avisos, cursos, salas, aulas, software.

La consulta del nombre del instructor en el apartado cursos es propia de la interfaz privada. Sin embargo, no es posible visualizar el nombre del instructor sin tener acceso a la tabla usuario.

Por medio de las siguientes líneas de código, los roles sin la autorización de consulta de la tabla usuario podrán consultar nombre y apellido de la tabla usuario, siendo la única información que puedan consultar de esta tabla.

```
GRANT SELECT (num_cuenta, nombre, apellido_pat) ON usuario TO jefe_de_sala;  
GRANT SELECT (num_cuenta, nombre, apellido_pat) ON usuario TO jefe_de_aula;  
GRANT SELECT (num_cuenta, nombre, apellido_pat) ON usuario TO apoyo;
```

El aspecto de la página web está dado por líneas de código escritas bajo el lenguaje de programación HTML, y estas han sido agrupadas y almacenadas en archivos de texto con extensión .php.

Al disponer de Apache instalados sobre la máquina virtual, el directorio destinado para almacenar los archivos que conforman la página es *var/www/html*.

Además, se crearon archivos adicionales para almacenar otro tipo de lenguaje complementario a la página web. Se tienen a los archivos con extensión .css los cuales contienen código que sirve para dar formato a los diferentes elementos dentro de la página. Por otro lado, se encuentran los archivos con extensión .js que son otro tipo de archivos. El lenguaje empleado para este tipo de archivos es javascript, y por medio de este lenguaje es posible implementar funciones que al ser ejecutadas estas generen un resultado dentro de la misma página (característica de las páginas web dinámica).

### 3.4 PRUEBAS Y MANTENIMIENTO (AJUSTES)

#### ▪ Consultas de datos

Dentro del código de la página se encuentran consultas ya preestablecidas mediante sentencias SQL. Por medio de recursos tales como menú de opciones, botones o recuadros para ingresar dígitos, el usuario podrá proporcionar parámetros para procesar las consultas y así obtener un resultado (requiriendo una intervención mínima por parte del usuario). El resultado obtenido para esta forma de consulta se encuentra en las secciones del apéndice C “*REPRESENTACIÓN GRÁFICA DE LA INTERFAZ DE USUARIO*”.

Algunas de las consultas utilizadas en la interfaz para la consulta de datos son las que se muestran a continuación, siendo las consultas multi tabla las utilizadas para este ejemplo (esto para demostrar la capacidad que tiene el sistema de procesar consultas complejas).

Para mostrar los datos de aula tales como número de sala, nombre de aula, número de equipos, y nombre, apellido y estado de los responsables de aula de sala 3, la sentencia de consulta SQL para tal objetivo es la que se muestra a continuación.

```
SELECT num_sala, nom_aula, nombre, apellido_pat, num_equipos, estado
FROM usuario s1, aula s2
WHERE s1.num_cuenta = s2.resp_aula
AND s2.num_sala = 3;
```

El resultado de la consulta antes descrita es el que se muestra a continuación.

num_sala	nom_aula	nombre	apellido_pat	num_equipos	estado
3	E	Alberto	Palacios	40	Activo
3	F	Raúl	Torres	40	Activo
3	G	Gabriel	Zúñiga	27	Activo
3	H	Óscar	Gámez	35	Activo

(4 filas)

**Figura 3.10** Consulta de datos de las aulas de sala 3 UNICA.

Fuente: Captura de pantalla <sup>65</sup>

Para mostrar los datos de curso tales como modalidad del curso, número de sala asignada, nombre del curso, nombre del software empleado, y clasificación del software de los cursos con modalidad presencial, la sentencia de consulta SQL para tal objetivo es la que se muestra a continuación.

```
SELECT s1.modalidad, num_sala, nom_curso, nom_software, clasificación
FROM curso s1, software s2, software_curso s3
WHERE s1.id_curso = s3.id_curso
AND s2.id_software = s3.id_software
AND s1.modalidad = 'Presencial';
```

<sup>65</sup> Sentencia SQL para realizar una multi tabla entre las tablas aula y usuario.

El resultado de la consulta antes descrita es el que se muestra a continuación.

modalidad	num_sala	nom_curso	nom_software	clasificacion
Presencial	3	Aplicaciones con AutoCAD	AutoCAD 2016	CAD
Presencial	3	Programación en C	Dev C++	IDE
Presencial	3	Aprendiendo Linux	GNU / Linux	SO

(3 filas)

**Figura 3.11** Consulta de datos de los cursos presenciales de UNICA.

Fuente: Captura de pantalla <sup>66</sup>

Para mostrar los datos de software tales como número de sala, nombre de software y clasificación de todos aquellos programas disponibles en sala 1, la sentencia de consulta SQL para tal objetivo es la que se muestra a continuación.

```
SELECT s2.num_sala, nom_software, clasificacion
FROM software s1, sala s2, software_sala s3
WHERE s1.id_software = s3.id_software
AND s2.num_sala = s3.num_sala
AND clasificacion = 'Aplicacion'
AND s2.num_sala = 1;
```

El resultado de la consulta antes descrita es el que se muestra a continuación.

num_sala	nom_software	clasificacion
1	Acrobat	Aplicacion
1	Essentials	Aplicacion
1	Reader	Aplicacion
1	VLC	Aplicacion
1	WinRAR	Aplicacion

(5 filas)

**Figura 3.12** Consulta del software disponible en sala 1.

Fuente: Captura de pantalla <sup>67</sup>

### ▪ Control de acceso y permisos de usuarios

La capacidad de consulta dependerá del tipo de usuario con el cual se accede a la base de datos, ya sea por medio de la interfaz o mediante la terminal de comandos PostgreSQL.

Tal como se ha mencionado, los usuarios que no formen parte del personal de las salas solamente tendrán acceso a los apartados de avisos, cursos, salas y software. Por otro lado, hay información que se considera sensible que solamente podrá ser consultada por el personal de sala. La siguiente figura muestra el formulario de inicio de sesión para el acceso a la interfaz con este tipo de información.

<sup>66</sup> Sentencia SQL para realizar una consulta multi tabla entre las tablas curso y software.

<sup>67</sup> Sentencia SQL para realizar una consulta multi tabla entre las tablas software y sala.



**Figura 3.13** *Formulario de inicio de sesión.*  
**Fuente:** Captura de pantalla <sup>68</sup>

Para un mayor control de la información el personal de sala está autorizado para acceder a información sensible hasta cierto punto. Por ejemplo, un usuario que forme parte de personal de sala cuyo role sea apoyo, tiene autorización para consultar información de avisos, cursos, salas, aulas y software. Cualquier intento de acceso a información para la cual no esté autorizado, en el caso del role apoyo esta información corresponde a equipos, reportes y usuarios, el resultado de la consulta aparecerá en blanco.

La siguiente figura es un ejemplo del resultado obtenido al realizar una consulta para la cual no se dispone de la autorización (iniciando sesión como usuario con role de apoyo y realizando la consulta de datos de algún reporte en específico – En el apartado reporte, sección ID reporte).



**Figura 3.14** *Consulta realizada en el apartado reporte.*  
**Fuente:** Captura de pantalla <sup>69</sup>

<sup>68</sup> Apartado de inicio de sesión para el personal de salas de cómputo.

URL de las imágenes: <https://www.pinterest.com.mx/>

<sup>69</sup> Resultado obtenido de procesar una consulta en el apartado reporte teniendo apoyo como cargo de usuario.

URL de las imágenes: <https://www.pinterest.com.mx/>

- **Estrés de la base de datos**

Un sistema de base de datos experimenta estrés cuando se presenta una carga en el número de transacciones solicitadas al sistema. En base a esto, se puede configurar la base de datos para mejorar el rendimiento del sistema.

Las pruebas de estrés en una base de datos PostgreSQL se efectúa con la herramienta pgbench. La siguiente sentencia forma parte de pgbench, con esto la base de datos “*bdunica*” es adecuada por el programa (esta sentencia se ejecuta en la línea de comandos del sistema operativo. Es decir que no es necesario ingresar en el entorno de línea de comandos de PostgreSQL).

```
[root@localhost ~]# pgbench -h localhost -p 5432 -U postgres -i -s 10 -d bdunica
```

Como se puede observar, la mayoría de los comandos de esta sentencia son comunes a los utilizados por la sentencia anterior para acceder a la base de datos por medio de un usuario. Esto es el nombre del host (-h), el número de puerto (-p), el nombre de usuario (-U), y el nombre de la base de datos (-d). Adicional a esto, se incorpora el comando -i que da inicio a la base de datos, y el comando -s que es un factor de escala.

La ejecución de esta sentencia da lugar a la creación de cuatro tablas adicionales dentro de la base de datos, las cuales son utilizadas por el programa pgbench para realizar las transacciones sobre la base de datos “*bdunica*”. En este caso, un factor de escala de 50 significa que se aumentará cincuenta veces el número de filas de cada tabla generada por pgbench.

Tabla	Número filas (s = 1)	Número de filas (s = 50)
pgbench_branches	1	50
pgbench_tellers	10	500
pgbench_accounts	100,000	5,000,000
pgbench_history	0	0

**Tabla 3.22** Tablas generadas por la sentencia pgbench.

Fuente: Elaboración propia <sup>70</sup>

---

<sup>70</sup> Por medio de las tablas se busca generar estrés sobre la base de datos.

Las tablas creadas por pgbench son las mencionadas en la Tabla 3.22, en la cual se muestra el número de filas que cada tabla tiene por defecto ( $s = 1$ ), y el número de filas resultantes de cada tabla generadas por el factor de escala especificado ( $s = 50$ ).

Las tablas al ser creadas también son llenadas con datos muestra, de esta forma se genera una mayor cantidad de datos, y por consiguiente se aumenta la carga para poder llevar a cabo la prueba de estrés sobre la base de datos.

Después de esto, se procede a ejecutar las transacciones para medir el rendimiento del sistema. La sentencia para efectuar tal acción es la que se muestra a continuación (la cual contempla los comandos empleados para acceder a la base de datos, más los comandos para efectuar las transacciones).

```
[root@localhost ~]# pgbench -h localhost -p 5432 -U postgres -c 50 -j 2 -t 100 -d bdunica
```

Como se puede observar, dentro de la sentencia se incluye el comando -c, el cual representa el número de clientes que acceden a la base de datos de forma simultánea, el comando -j, el cual representa el número de procesos de trabajo para efectuar las transacciones, y el comando -t, el cual representa el número de transacciones que ejecuta cada cliente.

La ejecución de esta sentencia implica que pgbench abrió 50 sesiones de usuario de forma simultánea. A su vez los 50 usuarios ejecutaron 100 transacciones cada uno, y este proceso fue simulado mediante el uso de 2 procesos de trabajo. El resultado obtenido de ejecutar tal sentencia es el que se muestra a continuación.

```
transaction type: TPC-B (sort of)
scaling factor: 10
query mode: simple
number of clients: 50
number of threads: 2
number of transactions per client: 100
number of transactions actually processed: 5000/5000
tps = 111.817532 (including connections establishing)
tps = 112.780311 (excluding connections establishing)
```

**Figura 3.15** Resultados iniciales de la prueba de estrés.

Fuente: Elaboración propia <sup>71</sup>

El resultado muestra que el número de transacciones totales solicitadas fue de 5000, de las cuales las 5000 fueron procesadas. Además de que el número de transacciones por segundo o TPS fue de 111 (incluyendo el tiempo de conexión de los usuarios).

---

<sup>71</sup> Despliegue de los resultados de la prueba de estrés utilizando la herramienta pgbench (con caché de 128 MB).

Para medir el rendimiento de la base de datos se utiliza el número de transacciones por segundo, para lo cual 111 transacciones por segundo se consideran como el promedio de ese factor.

Es posible aumentar el rendimiento de las TPS al modificar la cantidad de memoria caché, la cual utiliza PostgreSQL para almacenar el contenido de tablas e índices. Para esto, PostgreSQL utiliza el parámetro `shared_buffers`, al cual puede ser accedido y modificado ejecutando la sentencia que se muestra a continuación.

```
[root@localhost ~]# vim /var/lib/pgsql/14/data/postgresql.conf
```

Esta sentencia accede al archivo `postgresql.conf` localizado en la dirección especificada, el cual contiene el parámetro `shared_buffers`. Dentro del archivo se observa la cantidad de memoria caché de esta forma:

```
# - Memory -  
shared_buffers = 128MB    # min 128kB
```

Se cambia el valor de 128 a 256 MB (esto es, el doble de memoria). Una vez hecho esto se reinicia PostgreSQL con el siguiente comando:

```
sudo systemctl restart postgresql-14
```

Se ejecuta nuevamente la sentencia `pgbench` para efectuar las transacciones, comparando los resultados con los obtenidos anteriormente. Los nuevos resultados obtenidos se muestran a continuación.

```
transaction type: TPC-B (sort of)  
scaling factor: 10  
query mode: simple  
number of clients: 50  
number of threads: 2  
number of transactions per client: 100  
number of transactions actually processed: 5000/5000  
tps = 114.044519 (including connections establishing)  
tps = 115.008798 (excluding connections establishing)
```

**Figura 3.16** Resultados finales de la prueba de estrés.

Fuente: Elaboración propia <sup>72</sup>

El aumento de las TPS no fue muy significativo, ya que solo se alcanzó un valor de 114 (con un incremento de 3 transacciones, en comparación con el resultado anterior), Por lo cual, el rendimiento máximo que puede llegar a alcanzar el sistema es en promedio de 111 transacciones por segundo.

---

<sup>72</sup> Despliegue de los resultados de la prueba de estrés utilizando la herramienta `pgbench` (con caché de 256 MB).

- **Copia de seguridad**

Una base de datos puede contar con una copia de seguridad de sus datos, esto es de gran utilidad debido a que toda base de datos es susceptible de experimentar fallas, como la eliminación de alguno de sus elementos, o una modificación que altere la integridad en sus datos.

Una copia de seguridad en una base de datos PostgreSQL se efectúa con la herramienta `pg_dump`. La sentencia para implementar una copia de seguridad de todos los elementos de la base de datos “*bdunica*” es la que se muestra a continuación.

```
[root@localhost ~]# pg_dump -h localhost -p 5432 -U postgres bdunica >
/var/lib/pgsql/14/backups/respaldo_22.sql
```

En la sentencia se incluye el nombre del host, el número de puerto y el nombre de usuario (esto es para realizar una copia de seguridad de la base de datos de manera remota). El resto de la sentencia se compone por el nombre de la base de datos a respaldar (“*bdunica*”), y el nombre con el que se quiera realizar la copia de seguridad (`respaldo_22.sql`). En este caso se incluye un directorio, el cual especifica el lugar en el que se desea que se guarde dicha copia de seguridad.

Con esto la base de datos ahora cuenta con una copia de seguridad, por lo que se procede a eliminar la base de datos con el fin de garantizar que la copia de seguridad creada funciona. La sentencia para la eliminación de la base de datos es la que se muestra a continuación.

```
DROP DATABASE bdunica;
```

Después de esto se crea una nueva base de datos, se le asigna el mismo nombre que la base de datos eliminada, pero con la diferencia de que ahora se encuentra vacía. La sentencia para esto es la que se muestra a continuación.

```
CREATE DATABASE bdunica;
```

Por último, para reestablecer la base de datos se ejecuta la sentencia que se muestra a continuación.

```
[root@localhost ~]# psql -h localhost -p 5432 -U postgres bdunica <
/var/lib/pgsql/14/backups/respaldo_22.sql
```

La sentencia es casi la misma que la utilizada para crear la copia de seguridad de la base de datos, utilizando los mismos parámetros para acceder a la base de datos. A esta se incluye el nombre de la base de datos donde se efectuará la restauración (“*bdunica*”), seguido del directorio y nombre del archivo generado como copia de seguridad.

## **RESULTADOS**

El servidor de base de datos está implementado para favorecer el funcionamiento de las actividades que se llevan a cabo en las salas de cómputo de la Facultad de Ingeniería. Para lo cual, el personal podrá realizar comunicados en relación a algún cambio en las actividades de la sala, programación de juntas, desempeñar alguna función en particular, entre otros.

Se espera que el servidor de base de datos favorezca en la formación de los alumnos de la Facultad de Ingeniería, gracias a que proporcionará información sobre los servicios impartidos por UNICA. Su finalidad es que los alumnos sean más conscientes del préstamo de equipo de cómputo, el software del que se dispone en las salas de cómputo, y de la información sobre los cursos impartidos por el personal de UNICA.

Se podrá realizar un registro sobre las irregularidades que comprometan el funcionamiento del equipo de cómputo, esto para dar atención a los equipos que lo requieran. La finalidad es mantener un buen funcionamiento en todos los equipos.

Además de esto, se tomaron en cuenta algunos elementos en el diseño del sistema de base de datos para mejorar su funcionamiento. Estos elementos son:

- El uso de una cuenta y contraseña de usuario para acceder a la base de datos.
- Proporcionar privilegios a los usuarios en base a como son catalogados dentro del sistema.
- El uso de un respaldo de la base de datos.

## CONCLUSIONES

Considero el presente proyecto como un material didáctico que proporciona los fundamentos en materia de las TIC que permite a todo estudiante adquirir los conocimientos para implementar su propio servidor de base de datos, con la finalidad de poder aplicarlo en algún entorno en el que pueda resultar de utilidad. Un servidor de base de datos independientemente del entorno, este es de utilidad para gestionar grandes cantidades de información de forma simple y ordenada para su consulta y manipulación, además de que es de utilidad para la toma de decisiones.

El proyecto de tesis desarrollado es un claro ejemplo del uso que se le puede dar a un sistema de base de datos aplicado a una situación del mundo real. Este proyecto además de ser un medio que brinde información de los servicios, actividades y recursos de las salas, también es un medio para definir un método de trabajo con el que se pueda sistematizar las actividades de las salas de cómputo, favoreciendo en una mejor organización y cumplimiento de tales actividades, así como una forma de demostrar si las actividades se están cumpliendo en tiempo y forma.

Por medio del presente proyecto se demuestra lo eficaz que resulta el uso del software libre en el desarrollo de aplicaciones para empresas de pequeña y mediana escala. En cambio, el uso de software comercial es viable para grandes empresas en las que se haga uso de una gran cantidad de información, o empresas que requieran de una mayor calidad y cantidad de recursos.

El uso de los sistemas de base de datos supone una serie de beneficios en el almacenamiento de datos. Algunas aplicaciones como procesadores de texto y hojas de cálculo son de utilidad para almacenar datos. Sin embargo, los beneficios de un sistema de base de datos son: realizar un almacenamiento de datos en formato simple (evitando un mayor consumo de memoria y recursos del equipo), control de acceso de usuarios, acceso concurrente de los usuarios, la integridad de los datos, copias de seguridad de la base de datos, entre otros.

El trabajo de tesis presentado proporciona los conceptos básicos para implementar un servidor de base de datos. Como resultado se obtuvo la base de datos descrita en este trabajo, la cual demuestra tener un desempeño óptimo. Sin embargo, para que la base de datos sea de utilidad se deberá de dar mantenimiento a la base de datos, el cual consiste en realizar la inserción, actualización y eliminación de los datos, así como realizar cambios en la estructura de la base de datos (esto en base a los requerimientos cambiantes del mundo real, o para cambiar ciertos aspectos que mejoren la calidad del sistema).

El motivo por el cual decidí participar en el desarrollo de este proyecto fue para adquirir conocimientos relacionados al manejo y desarrollo de los sistemas de base de datos. Considero que estos conocimientos son de gran importancia, ya que hoy en día las empresas en su mayoría hacen uso de las bases de datos para poder gestionar sus recursos.

Debido a esto algunas empresas solicitan a sus futuros candidatos conocimientos de manejadores de bases de datos o lenguajes de consultas.

La importancia del proceso de diseño de una base de datos es identificar los datos del mundo real que son relevantes almacenar (entidades y atributos), así como aprender a organizar los datos en estructuras eficientes para su búsqueda e interpretación de los mismos. Esto junto con el hecho de que algunos procesadores de texto almacenan datos en forma de tablas (de una forma muy similar a como lo hacen las bases de datos relacionales), se demuestra que el trabajo de tesis descrito también supone un beneficio en la elaboración de documentos digitales para el almacenamiento de datos.

## Apéndice A. INSTALACIÓN DEL SOFTWARE DE LA BASE DE DATOS

### ■ Creación de la máquina virtual

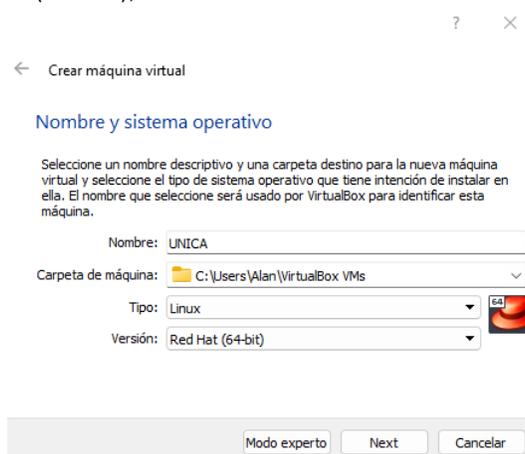
Al abrir el programa VirtualBox aparece una pantalla con múltiples opciones, para lo cual se selecciona la opción Nueva para crear una nueva máquina virtual, tal como se muestra a continuación.



**Figura A.1** Pantalla de inicio de VirtualBox.

Fuente: Captura de pantalla <sup>73</sup>

El siguiente paso es designar un nombre a la máquina virtual, seleccionar el directorio donde estará alojada la máquina virtual, seleccionar el tipo de sistema operativo y la versión del mismo. Es así que, el nombre de la máquina virtual es UNICA, el tipo de sistema operativo es Linux, y la versión de este es Red Hat (64-bit), tal como se muestra a continuación.



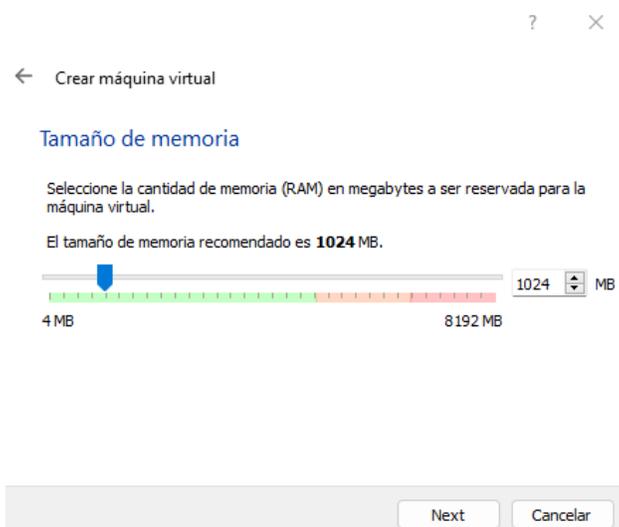
**Figura A.2** Datos básicos de la máquina virtual.

Fuente: Captura de pantalla <sup>74</sup>

<sup>73</sup> Pantalla principal del programa Oracle VM VirtualBox.

<sup>74</sup> Pantalla para la selección del nombre, directorio y sistema operativo de la máquina virtual.

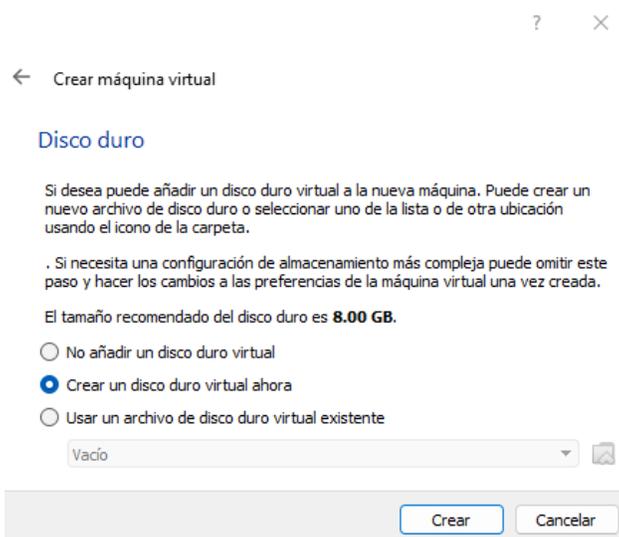
El siguiente paso es asignar la cantidad de memoria RAM que tendrá la máquina virtual, para lo cual se selecciona la cantidad mínima recomendada, tal como se muestra a continuación.



**Figura A.3** Cantidad de memoria RAM reservada.

Fuente: Captura de pantalla <sup>75</sup>

El siguiente paso es especificar si se requiere la creación de un disco duro virtual, o si ya existe un disco a partir del cual se desea crear la máquina virtual. Para esto, se escoge la opción para crear un nuevo disco duro para la máquina virtual, tal como se muestra a continuación.



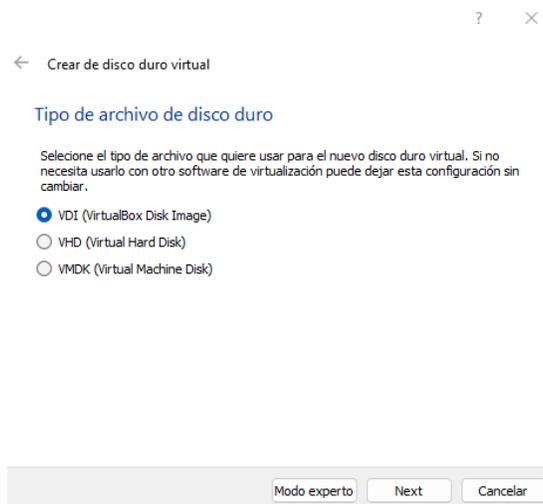
**Figura A.4** Forma de implementación del disco duro.

Fuente: Captura de pantalla <sup>76</sup>

<sup>75</sup> Pantalla para la selección de la capacidad de la memoria RAM que ocupará la máquina virtual.

<sup>76</sup> Pantalla sobre la forma de implementación del disco duro virtual.

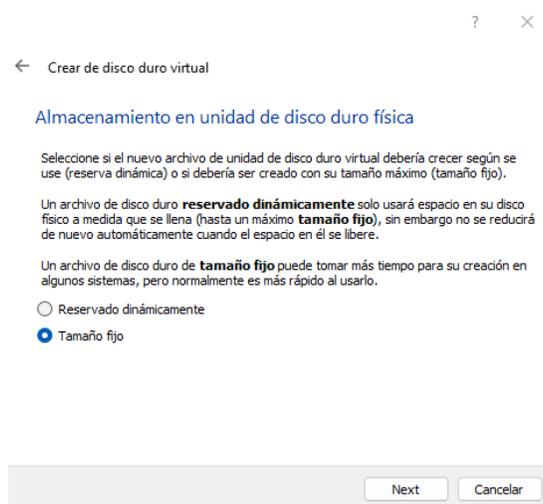
El siguiente paso es especificar el tipo de archivo del cual se requiere que sea el disco duro, para lo cual se selecciona el tipo de archivo recomendado, tal como se muestra a continuación.



**Figura A.5** Tipo de archivo del disco duro.

Fuente: Captura de pantalla <sup>77</sup>

El siguiente paso es especificar el comportamiento del tamaño del disco para almacenar información, para lo cual se selecciona que el disco sea de un tamaño estático (es decir, que no puede cambiar de tamaño a medida que se llena el disco), tal como se muestra a continuación. En caso contrario, el tamaño del disco puede aumentar conforme este se empieza a llenar (tamaño dinámico).



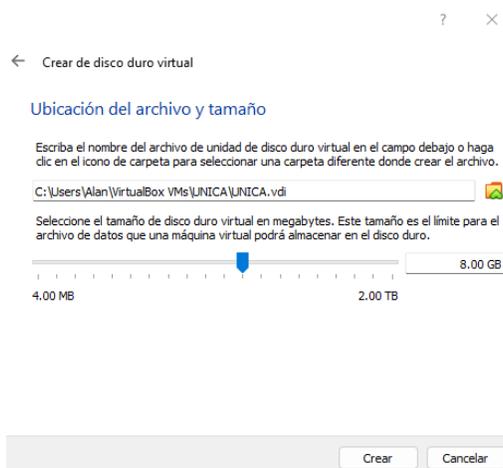
**Figura A.6** Configuración del comportamiento del disco duro.

Fuente: Captura de pantalla <sup>78</sup>

<sup>77</sup> Pantalla para la selección del tipo de archivo bajo el cual está implementado el disco duro.

<sup>78</sup> Pantalla para especificar el comportamiento del tamaño del disco duro de la máquina virtual.

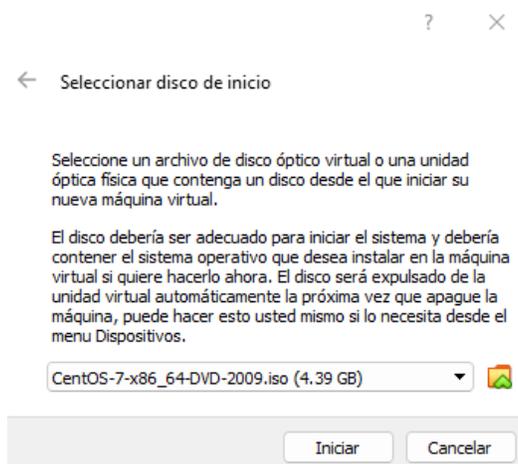
El siguiente paso es seleccionar el directorio donde estará alojado el disco duro de la máquina virtual y su capacidad en memoria, para lo cual se seleccionan los valores por defecto, tal como se muestra a continuación.



**Figura A.7** Directorio y capacidad del disco duro.  
Fuente: Captura de pantalla <sup>79</sup>

#### ▪ Instalación de CentOS 7

La instalación inicialmente se realiza seleccionando el archivo ISO que contiene el sistema operativo CentOS 7, tal como se muestra a continuación.



**Figura A.8** Selección del archivo ISO para instalar el sistema operativo.  
Fuente: Captura de pantalla <sup>80</sup>

<sup>79</sup> Pantalla para la selección del directorio y capacidad del disco duro que ocupará la máquina virtual.

<sup>80</sup> La página para efectuar la descarga del archivo de CentOS 7 es: <https://www.centos.org/>.

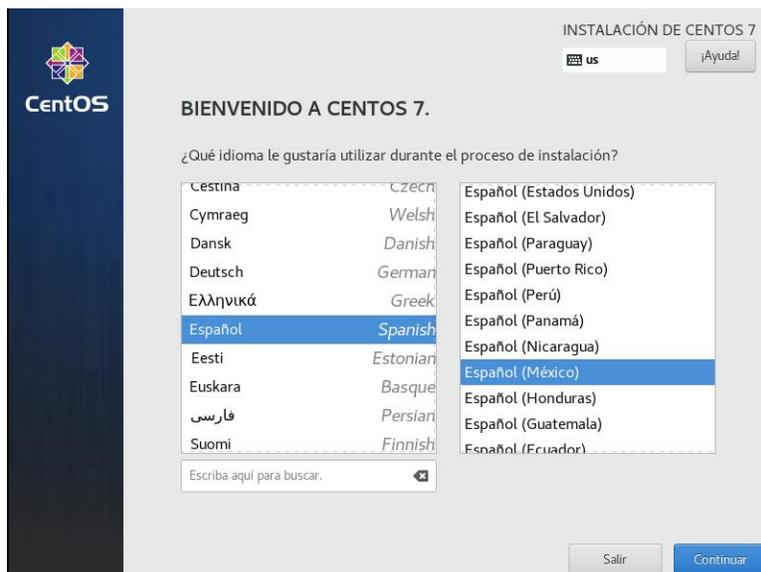
Una vez hecho esto, se despliega un menú con las opciones mostradas en la siguiente figura, para lo cual se selecciona la opción de instalar CentOS 7, tal como se muestra a continuación.



**Figura A.9** Pantalla de inicio para la instalación de CentOS 7.

Fuente: Captura de pantalla <sup>81</sup>

Para proceder con la instalación se debe seleccionar el idioma que debe ocupar CentOS 7 para interactuar con el usuario, tal como se muestra a continuación.



**Figura A.10** Selección del idioma de CentOS 7.

Fuente: Captura de pantalla <sup>82</sup>

<sup>81</sup> Pantalla de inicio para la instalación de CentOS 7.

<sup>82</sup> Pantalla para la selección del idioma del sistema.

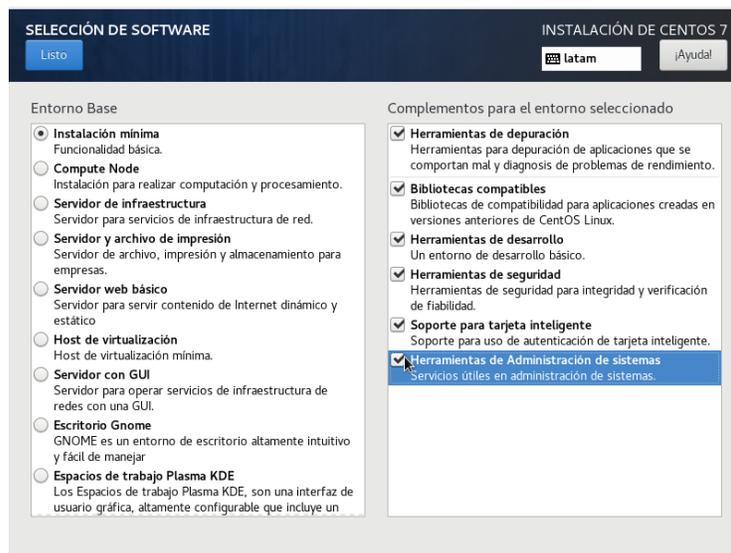
La configuración del sistema se efectúa por medio del menú mostrado a continuación, para lo cual en cada sección se selecciona la opción recomendada por defecto.



**Figura A.11** Menú principal de configuración del sistema.

Fuente: Captura de pantalla <sup>83</sup>

Para efectuar la instalación del tipo de entorno del sistema, en la sección de selección de software se selecciona la opción instalación mínima junto con todas las herramientas y opciones que contiene, tal como se muestra a continuación.



**Figura A.12** Selección de software.

Fuente: Captura de pantalla <sup>84</sup>

<sup>83</sup> Pantalla de configuración para la definición ciertos aspectos del sistema.

<sup>84</sup> Pantalla para la selección de las herramientas y complementos que tendrá instalado el sistema.

Para concluir con la instalación se solicita al usuario ingresar un nombre de usuario, una contraseña, y una contraseña para el super usuario, la cual será solicitada cada que se requiera efectuar cambios en el sistema. Esta configuración se realiza por medio de la sección de ajustes de usuario, tal como se muestra a continuación.



**Figura A.13** Asignación de usuario y contraseña.

Fuente: Captura de pantalla <sup>85</sup>

#### ▪ Instalación de PostgreSQL

La instalación de PostgreSQL se realiza por medio de comandos, los cuales son ejecutados desde la consola del sistema (esto debido a que el sistema operativo fue instalado sin interfaz gráfica).

Para iniciar con la instalación de PostgreSQL se ejecuta el siguiente comando, el cual configura el repositorio para poder instalar el paquete que contiene la última versión de PostgreSQL desde la URL especificada.

```
sudo yum -y install https://download.postgresql.org/pub/repos/yum/repopms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm
```

Para actualizar los nuevos paquetes en el repositorio correspondientes al repositorio oficial de PostgreSQL se ejecuta el siguiente comando.

```
sudo yum update -y
```

<sup>85</sup> Pantalla para la definición de un usuario, una contraseña de usuario, y una contraseña root.

Para efectuar la instalación de los paquetes disponibles en el repositorio se ejecuta el siguiente comando.

```
sudo yum -y install postgresql14-server
```

Esto crea un servicio llamado postgresql-14, aunque para activar este servicio se inicializan los archivos y bases de datos de PostgreSQL por medio de la ejecución del siguiente comando.

```
sudo /usr/pgsql-14/bin/postgresql-14-setup initdb
```

Posteriormente, el servicio postgresql-14 es habilitado junto con el sistema operativo CentOS 7 por medio del siguiente comando.

```
sudo systemctl enable postgresql-14
```

Finalmente se activa dicho servicio por medio del siguiente comando.

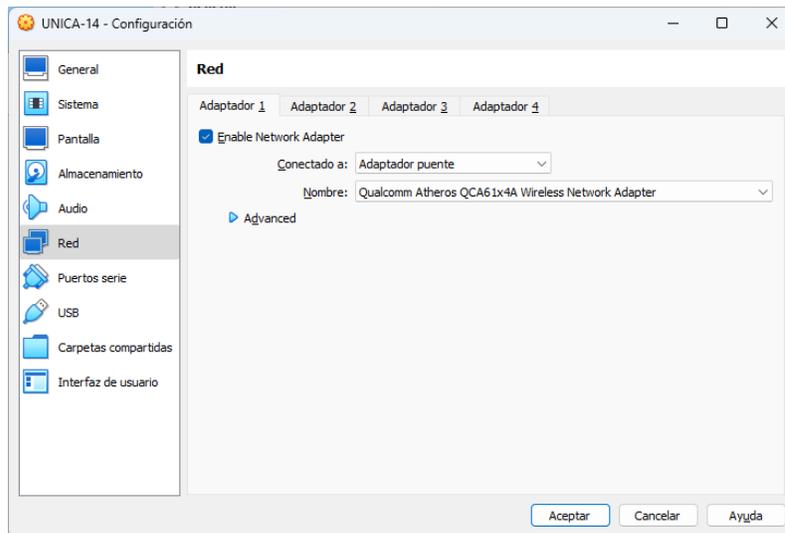
```
sudo systemctl start postgresql-14
```

#### ▪ **Configuración de red**

Para que la máquina virtual pueda establecer una conexión a internet, el valor por defecto configurado para la conexión de red de la máquina virtual es del tipo NAT. Es así que para conectarse a internet mediante la máquina virtual a esta se le asignará por dirección IP la dirección IP privada que tiene el equipo físico. El inconveniente de este tipo de configuración es que aquellos equipos conectados a la misma red (red local), no podrán conectarse de forma directa con la máquina virtual.

La configuración de red adecuada para la máquina virtual es del tipo adaptador de puente. Para configurar el tipo de red se debe seleccionar la opción de configuración al dar clic derecho sobre la máquina virtual, o mediante la opción de configuración que se encuentra en la parte superior. De esta forma la máquina virtual formará parte de la red local como un equipo independiente del equipo físico que la aloja.

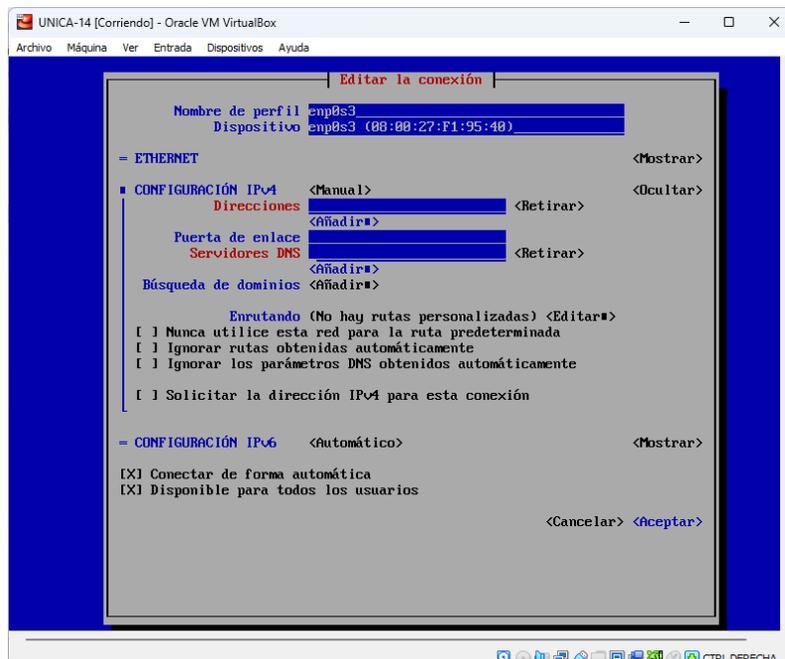
El apartado de configuración de red de la máquina virtual con la opción adaptador de puente seleccionada se muestra a continuación.



**Figura A.14** Configuración de red.

Fuente: Captura de pantalla <sup>86</sup>

Para continuar con la configuración es necesario iniciar la máquina, y una vez en la línea de comandos del sistema CentOS (como usuario root) se ingresa el comando **nmtui** (Network Manager Terminal User Interface).



**Figura A.15** Edición de conexión.

Fuente: Captura de pantalla <sup>87</sup>

<sup>86</sup> Sección de configuración de red de la máquina virtual en Virtual Box.

<sup>87</sup> Sección para editar la conexión de la máquina virtual en Virtual Box.

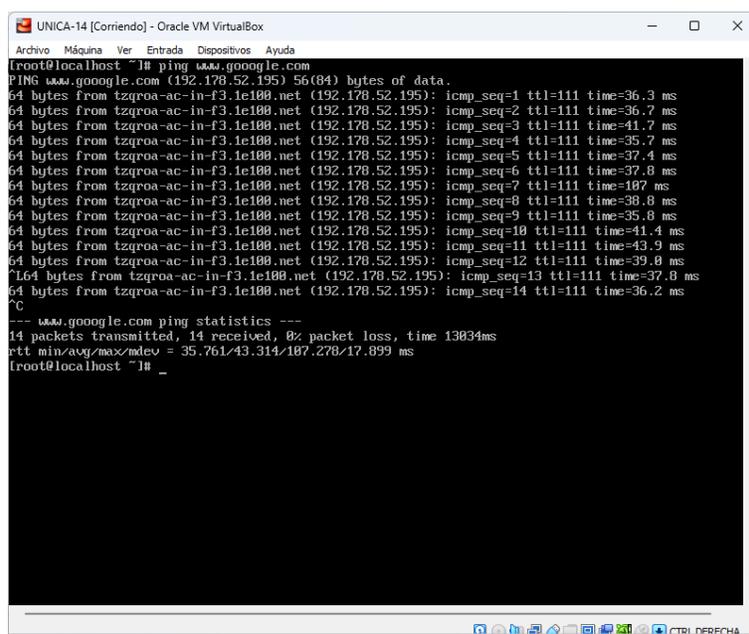
En pantalla aparecerá el gestor de red del sistema, para dirigirse a la sección de configuración de red se seleccionan las opciones “*Modificar una conexión -> enp0s3*” (siendo enp0s3 la tarjeta de red empleada para el host). Se procede a editar la conexión, para lo cual la opción “*CONFIGURACIÓN IPv4*” deberá cambiarse de automático a manual. Seguido se procede a configurar los parámetros de red (dirección IP, puerta de enlace y servidor DNS). El apartado para realizar tal configuración es el mostrado en la Figura A.14 *Edición de conexión*.

Para guardar la configuración se seleccionan las opciones de aceptar y salir. Dentro del mismo apartado aparece la opción “*Activar conexión*”, al seleccionar esta opción aparecerá nuevamente el nombre de la tarjeta de red, y para confirmar que esta se encuentra activa deberá tener un asterisco.

Una vez completada la configuración se procede a salir del menú y regresar nuevamente a la consola. Para confirmar tales cambios se reinicia la red por medio de la siguiente línea de comando:

```
systemctl restart network.service
```

Por último, para confirmar que el sistema no presente fallas de conexión se efectúa una transmisión de paquetes por medio de la herramienta ping. La transmisión se finaliza con las teclas Ctl + C en cuyo caso la pérdida de los paquetes deberá de ser de 0%. La transmisión efectuada por el ping deberá ser como la mostrada a continuación.



```
UNICA-14 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@localhost ~]# ping www.google.com
PING www.google.com (192.178.52.195) 56(84) bytes of data:
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=1 ttl=111 time=36.3 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=2 ttl=111 time=36.7 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=3 ttl=111 time=41.7 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=4 ttl=111 time=35.7 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=5 ttl=111 time=37.4 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=6 ttl=111 time=37.8 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=7 ttl=111 time=107 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=8 ttl=111 time=38.8 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=9 ttl=111 time=35.8 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=10 ttl=111 time=41.4 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=11 ttl=111 time=43.9 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=12 ttl=111 time=39.8 ms
^L64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=13 ttl=111 time=37.8 ms
64 bytes from tzqroa-ac-in-f3.1e100.net (192.178.52.195): icmp_seq=14 ttl=111 time=36.2 ms
^C
--- www.google.com ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13834ms
rtt min/avg/max/mdev = 35.761/43.314/107.278/17.899 ms
root@localhost ~]#
```

**Figura A.16 Prueba de conexión PING.**

Fuente: Captura de pantalla <sup>88</sup>

<sup>88</sup> Transmisión de paquetes empleando la herramienta PING.

- **Instalación de Apache**

Ahora para que el servidor sea funcional se hará uso de un servidor Apache cuya función será mostrar la página web a los usuarios. En este caso Apache forma parte de los repositorios de software predeterminados de CentOS, siendo antes necesaria su actualización. Esto se hace por medio de la siguiente línea de comando:

```
sudo yum update httpd
```

Apache ahora puede ser instalado por medio de la siguiente línea de comando:

```
sudo yum install httpd
```

Para regular el tráfico del servidor se recomienda la instalación de firewalls, los cuales habilitarán puertos para atender las solicitudes de los usuarios. La siguiente línea de comando habilita el puerto 80 el cual atiende las solicitudes de los usuarios empleando el protocolo http.

```
sudo firewall-cmd --permanent --add-service=http
```

Un puerto adicional que también debe ser habilitado es el número 443, el cual hace uso del protocolo https para atender las solicitudes de los usuarios. La habilitación de dicho puerto se realiza mediante la siguiente línea de comando:

```
sudo firewall-cmd --permanent --add-service=https
```

Para procesar los cambios efectuados se deberá de recargar el firewall mediante la siguiente línea de comando:

```
sudo firewall-cmd --reload
```

Se procede a inicializar de forma manual el servicio Apache mediante la siguiente línea de comando:

```
sudo systemctl start httpd
```

Como resultado el servicio Apache ya se encuentra instalado y funcionando sobre el equipo. Aunque para confirmar esto se puede ejecutar la siguiente línea de comando:

```
sudo systemctl status httpd
```

Por consiguiente, en pantalla deberá mostrarse una descripción en la cual se describa el estado del servidor Apache como active (running).

#### ▪ **Instalación de PHP**

Para dar lugar a la página web se requiere de un lenguaje de programación complementario a HTML. El lenguaje que se optó utilizar para desarrollar este proyecto ha sido PHP.

Para obtener el repositorio de PHP este debe ser descargado desde un sitio web de uso exclusivo para su descarga. El comando para tal acción es el que se muestra a continuación.

```
sudo yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

Para configurar el nuevo repositorio se realiza la instalación de la herramienta yum mediante la siguiente línea de comando:

```
sudo yum install -y yum-utils
```

Para configurar la versión del repositorio descargado, esto se realiza mediante la siguiente línea de comando, siendo la versión 7.4 la seleccionada para el presente proyecto.

```
sudo yum-config-manager --enable remi-php74
```

Para actualizar los paquetes del repositorio de php se ejecuta el siguiente comando:

```
sudo yum update -y
```

Una vez configurado el repositorio se procede con su instalación mediante la siguiente línea de comando:

```
sudo yum -y install php
```

Para implementar los cambios efectuados se reinicia el servidor mediante la siguiente línea de comando:

```
sudo systemctl restart httpd
```

Se debe tomar en cuenta que por motivos de seguridad al intentar acceder a la base de datos por medio de un archivo PHP el sistema impide establecer una conexión. Debido a esto se necesario configurar la conexión ejecutando las líneas de código mostradas a continuación.

```
Service httpd stop  
Service postgresql stop  
Setsebool -P httpd_can_network_connect 1  
Service httpd start  
Service postgresql start
```

## Apéndice B. LÓGICA DE DESARROLLO DE LA BASE DE DATOS

- Creación de las tablas de la base de datos

*Tabla aula*

```
CREATE TABLE aula (  
  
    nom_aula            VARCHAR(2),  
    resp_aula          NUMERIC(9)      NOT NULL UNIQUE,  
    num_sala           NUMERIC(1)      NOT NULL,  
    numEquipos         NUMERIC(2)      NOT NULL,  
  
    PRIMARY KEY (nom_aula),  
    FOREIGN KEY (resp_aula) REFERENCES usuario ON DELETE SET NULL,  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL,  
    CHECK (nom_aula IN ('A','BD','C','E','F','G','H','I','J','K')),  
    CHECK (num_sala <= 4));
```

**Figura B.1** Definición de la tabla aula.

Fuente: Elaboración propia <sup>89</sup>

*Tabla aviso*

```
CREATE TABLE aviso(  
  
    id_aviso           NUMERIC(3),  
    titulo             VARCHAR(50)     NOT NULL,  
    autor              VARCHAR(75)     DEFAULT CURRENT_USER,  
    num_cuenta         NUMERIC(9)      NOT NULL,  
    fecha_inicio       DATE            DEFAULT CURRENT_DATE,  
    fecha_fin          DATE            NOT NULL,  
    descripcion        TEXT            NOT NULL,  
  
    PRIMARY KEY (id_aviso),  
    FOREIGN KEY (num_cuenta) REFERENCES usuario ON DELETE CASCADE,  
    CHECK (fecha_inicio < fecha_fin));
```

**Figura B.2** Definición de la tabla aviso.

Fuente: Elaboración propia <sup>90</sup>

---

<sup>89</sup> Sentencia SQL para la creación de la tabla aula.

<sup>90</sup> Sentencia SQL para la creación de la tabla aviso.

### Tabla curso

```
CREATE TABLE curso(  
  
    id_curso          NUMERIC(2),  
    cartel            VARCHAR(120)      DEFAULT NULL,  
    nom_curso         VARCHAR(100)      NOT NULL,  
    modalidad        VARCHAR(10)       NOT NULL,  
    fecha_inicio     DATE              NOT NULL,  
    fecha_fin        DATE              NOT NULL,  
    dias             VARCHAR(30)       NOT NULL,  
    horario          VARCHAR(17)       NOT NULL,  
    duración         NUMERIC(3)        NOT NULL,  
    cupo             NUMERIC(3)        NOT NULL,  
    num_sala         NUMERIC(1)        DEFAULT NULL,  
    temario          VARCHAR(120)      DEFAULT NULL,  
  
    PRIMARY KEY (id_curso),  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL,  
    CHECK (modalidad IN ('Presencial','En linea')));
```

**Figura B.3** Definición de la tabla curso.

Fuente: Elaboración propia <sup>91</sup>

### Tabla equipo

```
CREATE TABLE equipo(  
  
    id_equipo        NUMERIC(3),  
    id_inventario   NUMERIC(3)        NOT NULL UNIQUE,  
    num_equipo       NUMERIC(3)        NOT NULL,  
    modelo          VARCHAR(10)       NOT NULL,  
    marca           VARCHAR(25)       NOT NULL,  
    sistema_op      VARCHAR(7)        NOT NULL,  
    num_sala        NUMERIC(1)        NOT NULL,  
    nom_aula        VARCHAR(2)        NOT NULL,  
    estado          VARCHAR(8)        NOT NULL,  
  
    PRIMARY KEY (id_equipo),  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL,  
    FOREIGN KEY (nom_aula) REFERENCES aula ON DELETE SET NULL,  
    CHECK (sistema_op IN ('windows','Linux')),  
    CHECK (num_sala <= 4),  
    CHECK (nom_aula IN ('A','BD','C','E','F','G','H','I','J','K')),  
    CHECK (estado IN ('Activo','Revisar','Inactivo')));
```

**Figura B.4** Definición de la tabla equipo.

Fuente: Elaboración propia <sup>92</sup>

<sup>91</sup> Sentencia SQL para la creación de la tabla curso.

<sup>92</sup> Sentencia SQL para la creación de la tabla equipo.

### Tabla reporte

```
CREATE TABLE reporte(  
  
    id_reporte          NUMERIC(3),  
    id_maquina         VARCHAR(120)    NOT NULL,  
    foto               VARCHAR(120)    DEFAULT NULL,  
    autor              VARCHAR(75)     DEFAULT CURRENT_USER,  
    num_cuenta         NUMERIC(9)      NOT NULL,  
    situacion          VARCHAR(14)    NOT NULL,  
    condicion          VARCHAR(9)     NOT NULL,  
    num_sala           NUMERIC(1)     NOT NULL,  
    num_equipo         NUMERIC(3)     NOT NULL,  
    id_equipo          NUMERIC(3)     NOT NULL,  
    id_inventario     NUMERIC(3)     NOT NULL,  
    fecha_reporte     DATE           DEFAULT CURRENT_DATE,  
    fecha_entrega     DATE           DEFAULT NULL,  
    motivo            TEXT           NOT NULL,  
    nota              TEXT           NOT NULL,  
  
    PRIMARY KEY (id_reporte),  
    FOREIGN KEY (num_cuenta) REFERENCES usuario ON DELETE RESTRICT,  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL,  
    FOREIGN KEY (id_equipo) REFERENCES equipo ON DELETE CASCADE,  
    CHECK (situacion IN ('Solucionado','Sin solucionar')),  
    CHECK (condicion IN ('Nuevo','Activo','Historico')),  
    CHECK (num_sala <= 4),  
    CHECK (fecha_reporte < fecha_entrega));
```

Figura B.5 Definición de la tabla reporte.

Fuente: Elaboración propia <sup>93</sup>

### Tabla sala

```
CREATE TABLE sala(  
  
    num_sala           NUMERIC(1),  
    resp_sala         NUMERIC(9)     NOT NULL UNIQUE,  
    telefono           NUMERIC(10)   NOT NULL,  
    control            VARCHAR(120)  DEFAULT NULL,  
    ubicacion         TEXT           NOT NULL,  
  
    PRIMARY KEY (num_sala),  
    FOREIGN KEY (resp_sala) REFERENCES usuario ON DELETE SET NULL,  
    CHECK (num_sala <= 4));
```

Figura B.6 Definición de la tabla sala.

Fuente: Elaboración propia <sup>94</sup>

<sup>93</sup> Sentencia SQL para la creación de la tabla reporte.

<sup>94</sup> Sentencia SQL para la creación de la tabla sala.

### Tabla software

```
CREATE TABLE software(  
  
    id_software      NUMERIC(2),  
    foto             VARCHAR(120)      DEFAULT NULL,  
    nom_software     VARCHAR(25)       NOT NULL,  
    clasificacion    VARCHAR(17)       NOT NULL,  
    descripcion      TEXT              NOT NULL,  
    instalacion      TEXT              NOT NULL,  
  
    PRIMARY KEY (id_software),  
    CHECK (clasificacion IN ('Aplicacion', 'CAD', 'IDE', 'Navegador',  
    'Ofimatica', 'SO')));
```

Figura B.7 Definición de la tabla software.

Fuente: Elaboración propia <sup>95</sup>

### Tabla software\_curso

```
CREATE TABLE software_curso(  
  
    id_software      NUMERIC(2)       NOT NULL,  
    id_curso         NUMERIC(2)       NOT NULL,  
  
    FOREIGN KEY (id_software) REFERENCES software ON DELETE CASCADE,  
    FOREIGN KEY (id_curso) REFERENCES curso ON DELETE CASCADE);
```

Figura B.8 Definición de la tabla software\_curso.

Fuente: Elaboración propia <sup>96</sup>

### Tabla software\_equipo

```
CREATE TABLE software_equipo(  
  
    id_software      NUMERIC(2)       NOT NULL,  
    id_equipo        NUMERIC(3)       NOT NULL,  
  
    FOREIGN KEY (id_software) REFERENCES software ON DELETE CASCADE,  
    FOREIGN KEY (id_equipo) REFERENCES equipo ON DELETE CASCADE);
```

Figura B.9 Definición de la tabla software\_equipo.

Fuente: Elaboración propia <sup>97</sup>

---

<sup>95</sup> Sentencia SQL para la creación de la tabla software.

<sup>96</sup> Sentencia SQL para la creación de la tabla software\_curso.

<sup>97</sup> Sentencia SQL para la creación de la tabla software\_equipo.

### Tabla software\_sala

```
CREATE TABLE software_sala(  
  
    id_software          NUMERIC(2)          NOT NULL,  
    num_sala             NUMERIC(1)          NOT NULL,  
  
    FOREIGN KEY (id_software) REFERENCES software ON DELETE CASCADE,  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE CASCADE,  
    CHECK (num_sala <= 4));
```

Figura B.10 Definición de la tabla software\_sala.

Fuente: Elaboración propia <sup>98</sup>

### Tabla usuario

```
CREATE TABLE usuario(  
  
    num_cuenta          NUMERIC(9),  
    curp                CHAR(18)          NOT NULL,  
    foto                VARCHAR(120)      DEFAULT NULL,  
    nombre              VARCHAR(25)       NOT NULL,  
    apellido_pat        VARCHAR(25)       NOT NULL,  
    apellido_mat        VARCHAR(25)       NOT NULL,  
    sexo                CHAR              NOT NULL,  
    telefono            NUMERIC(10)       NOT NULL,  
    correo              VARCHAR(50)       NOT NULL,  
    facultad_escuela    VARCHAR(50)       NOT NULL,  
    carrera             VARCHAR(35)       NOT NULL,  
    clave_carrera       NUMERIC(3)        NOT NULL,  
    semestre            NUMERIC(2)        NOT NULL,  
    num_sala            NUMERIC(1)        DEFAULT NULL,  
    fecha_registro      DATE              NOT NULL,  
    clase               VARCHAR(15)       NOT NULL,  
    condicion           VARCHAR(15)       NOT NULL,  
    estado              VARCHAR(9)        NOT NULL,  
  
    PRIMARY KEY (num_cuenta),  
    FOREIGN KEY (num_sala) REFERENCES sala ON DELETE SET NULL,  
    CHECK (sexo IN ('H', 'M')),  
    CHECK (semestre <= 10),  
    CHECK (num_sala <= 4),  
    CHECK(clase IN ('Administrador', 'Jefe de sala', 'Jefe de aula', 'Apoyo')),  
    CHECK (condicion IN ('Becario', 'Servicio social')),  
    CHECK (estado IN ('Activo', 'Historico')));
```

Figura B.11 Definición de la tabla usuario.

Fuente: Elaboración propia <sup>99</sup>

<sup>98</sup> Sentencia SQL para la creación de la tabla software\_sala.

<sup>99</sup> Sentencia SQL para la creación de la tabla usuario.

### Tabla usuario\_curso

```
CREATE TABLE usuario_curso(  
  
    num_cuenta        NUMERIC(9)        NOT NULL,  
    id_curso          NUMERIC(2)        NOT NULL,  
  
    FOREIGN KEY (num_cuenta) REFERENCES usuario ON DELETE CASCADE,  
    FOREIGN KEY (id_curso) REFERENCES curso ON DELETE CASCADE);
```

Figura B.12 Definición de la tabla usuario\_curso.

Fuente: Elaboración propia <sup>100</sup>

### Tabla usuario\_reporte

```
CREATE TABLE usuario_reporte(  
  
    num_cuenta        NUMERIC(9)        NOT NULL,  
    id_reporte        NUMERIC(3)        NOT NULL,  
  
    FOREIGN KEY (num_cuenta) REFERENCES usuario ON DELETE CASCADE,  
    FOREIGN KEY (id_reporte) REFERENCES reporte ON DELETE CASCADE);
```

Figura B.13 Definición de la tabla usuario\_reporte.

Fuente: Elaboración propia <sup>101</sup>

## ▪ Creación de roles y usuarios

### Roles de la base de datos

```
CREATE ROLE administrador;  
CREATE ROLE jefe_de_sala;  
CREATE ROLE jefe_de_aula;  
CREATE ROLE apoyo;
```

Figura B.14 Creación de los roles de la base de datos.

Fuente: Elaboración propia <sup>102</sup>

## ▪ Asignación de privilegios a los roles de usuario de la base de datos

```
GRANT SELECT ON aula, aviso, curso, equipo, reporte, sala, software, usuario,  
software_curso, software_equipo, software_sala, usuario_curso, usuario_reporte  
TO administrador;
```

```
GRANT SELECT ON aula, aviso, curso, equipo, reporte, sala, software,  
software_curso, software_equipo, software_sala, usuario_curso, usuario_reporte  
TO jefe_de_sala;
```

<sup>100</sup> Sentencia SQL para la creación de la tabla usuario\_curso.

<sup>101</sup> Sentencia SQL para la creación de la tabla usuario\_reporte.

<sup>102</sup> Sentencias SQL para la creación de los roles: administrador, jefe de sala, jefe de aula y apoyo.

```
GRANT SELECT ON aula, aviso, curso, equipo, sala, software, software_curso,  
software_equipo, software_sala, usuario_curso TO jefe_de_aula;
```

```
GRANT SELECT ON aula, aviso, curso, sala, software, software_curso,  
software_sala, usuario_curso TO apoyo;
```

```
GRANT SELECT ON aula, aviso, curso, sala, software, software_curso,  
software_sala TO persona;
```

**Figura B.15** Asignación de privilegios a los roles de la base de datos.

Fuente: Elaboración propia <sup>103</sup>

#### Usuarios de la base de datos

```
CREATE USER rafael WITH PASSWORD '3587';  
CREATE USER alejandra WITH PASSWORD '3624';  
CREATE USER jaime WITH PASSWORD '3203';  
CREATE USER yoselin WITH PASSWORD '3627';  
CREATE USER persona WITH PASSWORD '3254';
```

**Figura B.16** Creación de cuentas de usuario de la base de datos.

Fuente: Elaboración propia <sup>104</sup>

#### Vínculo entre roles y usuarios

```
GRANT administrador TO rafael;  
GRANT jefe_de_sala TO alejandra;  
GRANT jefe_de_aula TO jaime;  
GRANT apoyo TO yoselin;
```

**Figura B.17** Asignación de roles a los usuarios de la base de datos.

Fuente: Elaboración propia <sup>105</sup>

---

<sup>103</sup> Sentencias SQL para la asignación de privilegios en base a las funciones designadas a cada role.

<sup>104</sup> Sentencias SQL para la creación de cuentas de usuario para el ingreso y autenticación sobre la base de datos.

<sup>105</sup> Sentencias SQL para la asignación de roles en base a la categoría a la que pertenece cada usuario.

# Apéndice C. REPRESENTACIÓN GRÁFICA DE LA INTERFAZ DE USUARIO

## Interfaz pública de usuario

### AVISOS

**UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO**

INICIO CURSOS SALAS SOFTWARE SESIÓN

**Avisos**

TÍTULO	AUTOR	FECHA IN	FECHA FIN	DESCRIPCIÓN
Actualizacion del antivirus	Alejandra	2022-06-24	2022-06-31	Favor de actualizar el antivirus de las computadoras de sala 4
Revision de las impresoras	Alejandra	2022-06-24	2022-06-31	Al personal de las salas se les solicita a que esta semana realicen una revision del estado de las impresoras
Cierre anticipado de las salas	Sergio	2022-06-24	2022-06-25	Se les informa que las salas cerrarán hoy a las 5:00 pm
Día feriado	Javier	2022-08-24	2022-08-31	Se les informa que el día jueves 1 de septiembre no habrá labores
Reunion de personal	Rafael	2022-06-24	2022-06-29	Aviso. El día 29 de agosto a las 12 pm se llevará a cabo una reunión entre el personal de UNICA (Sala 1)

### CURSOS

**UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO**

INICIO CURSOS SALAS SOFTWARE SESIÓN

**CURSOS PRESENCIALES**

A

Aplicaciones con AutoCAD

FECHA	2022-06-13 2022-06-17
DÍAS	Lunes - Viernes
HORARIO	9:00 a 13:00 hrs
DURACIÓN	20 hrs
CUPO	30
LUGAR	Sala 3 Edificio I (Anexo)

C

Programación en C

FECHA	2022-06-13 2022-06-17
DÍAS	Lunes - Viernes
HORARIO	9:00 a 13:00 hrs
DURACIÓN	20 hrs
CUPO	30
LUGAR	Sala 3 Edificio I (Anexo)

Aprendiendo Linux

FECHA	2022-06-13 2022-06-17
DÍAS	Lunes - Viernes
HORARIO	12:00 a 16:00 hrs
DURACIÓN	20 hrs
CUPO	30
LUGAR	Sala 3 Edificio I (Anexo)

### SALAS

**UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO**

INICIO CURSOS SALAS SOFTWARE SESIÓN

**SALAS UNICA**

SALAS

1

2

3

4

**SALA 2**

**UBICACIÓN**

Ubicación: Edificio M (Anexo)

**AULAS**

**SERVICIOS**

Computadoras	32
Impresoras	4
Usos múltiples	24
Escáneres	1

**SOFTWARE**

APLICACION	
CAD	
IDE	Dev C++,
NAVEGADOR	Chrome, Mozilla,
OFIMÁTICA	Microsoft Office 2016,
SISTEMA OP	GNU / Linux, Windows 10,

SOFTWARE



Figura C.1 Apartados de la interfaz pública de usuario.

Fuente: Captura de pantalla <sup>106</sup>

Interfaz privada de usuario

AVISOS



CURSOS



<sup>106</sup> Vistas de los apartados que son acceso público disponibles para personal y usuarios de salas de cómputo.

URL de las imágenes: <https://www.pinterest.com.mx/>

SALAS

DATOS DE SALA	
ENCARGADO	Alejandra Castañeda
TELÉFONO	557902862

SOFTWARE	
APLICACION	Acrobat, Essentials, Reader, VLC, WinRAR,
CAD	AutoCAD 2016, Matlab,
IDE	Dev C++, Visual C++,
NAVEGADOR	Chrome, Mozilla,
OFIMÁTICA	Microsoft Office 2016,
SISTEMA OP	GNU / Linux, Windows 10,

DATOS DE AULA I	
RESPONSABLE	Jaime Méndez
N° EQUIPOS	22

DATOS DE AULA J	
RESPONSABLE	Felipe Ayala
N° EQUIPOS	3

SOFTWARE

CLASIFICACIÓN
Navegador
SALA
1 2 3 4
CURSOS

DESCRIPCIÓN

Navegador de internet

USUARIOS

USUARIO	
Nombre	Rafael Valle Zepeda

ESCOLAR	
Carrera	Ingeniería Civil (107)
Escuela	Facultad de Ingeniería
Semestre	7

PERSONAL	
CURP	VAZR000410HDFLFP##
Sexo	H

CONTACTO	
Teléfono	5513787733
Correos	vallezepeda@gmail.com

SERVICIO	
Clase	Administrador
Condición	Becario
Fecha de registro	2021-08-30

Figura C.2 Apartados de la interfaz privada de usuario.

Fuente: Captura de pantalla <sup>107</sup>

<sup>107</sup> Vistas de los apartados que son acceso restringido disponibles para personal de salas de cómputo.

URL de las imágenes: <https://www.pinterest.com.mx/>

▪ Consultas preestablecidas dentro del apartado equipos (Interfaz privada)

Consulta de datos de un equipo por medio de su clave de ID.



Consulta de datos de múltiples equipos cuya característica común sea su estado (Activo, Revisar, Inactivo).



Consulta de datos de los equipos pertenecientes a alguna de las salas de UNICA (1, 2, 3, 4).



Figura C.3 Vistas resultantes del apartado equipos.

Fuente: Captura de pantalla <sup>108</sup>

<sup>108</sup> Vistas del apartado equipos de cada uno de los elementos de búsqueda disponibles en el apartado.

URL de las imágenes: <https://www.pinterest.com.mx/>

▪ Consultas preestablecidas dentro del apartado reporte (Interfaz privada)

Consulta de datos de un reporte por medio de su clave de ID.



Consulta de datos de múltiples reportes cuya característica común sea su situación (Solucionado o Sin solucionar).



Consulta de datos de los reportes pertenecientes a alguna de las salas de UNICA (1, 2, 3, 4).



Figura C.4 Vistas resultantes del apartado reportes.

Fuente: Captura de pantalla <sup>109</sup>

<sup>109</sup> Vistas del apartado reportes de cada uno de los elementos de búsqueda disponibles en el apartado.

URL de las imágenes: <https://www.pinterest.com.mx/>

## REFERENCIAS BIBLIOGRÁFICAS

- DATE, C. J. (1986). *Introducción a los Sistemas de Bases de Datos*. Addison-Wesley.
- Delaney, K. (2006). *Inside Microsoft SQL Server 2005: The Storage Engine*. Microsoft Press.

## REFERENCIAS ELECTRÓNICAS

### INTRODUCCIÓN

- Dirección General de Cooperación e Internacionalización. (2018). *Sedes / DGECI*. <https://www.unaminternacional.unam.mx/es/organizacion/sedes>
- Dirección General de Orientación y Atención Educativa. (2018). *Oferta Académica / UNAM*. <http://oferta.unam.mx/>
- Fundación UNAM. (s.f). *Sabes cómo se construyó Ciudad Universitaria / Fundación UNAM*. <https://www.fundacionunam.org.mx/donde-paso/sabes-como-se-construyo-ciudad-universitaria/>
- UNAM Global Noticias. (9 de mayo de 2019). *Presencia de la UNAM en la República Mexicana 2019 / UNAM Global*. <http://www.unamglobal.unam.mx/?p=64845>
- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Misión*. Recuperado el 29 de noviembre de 2020, de <http://www.fi-a.unam.mx/~unica/presentacion/mision.php>

### CAPÍTULO 1 ANTECEDENTES

- Facultad de Ingeniería. (6 de abril de 2020). *Misión y Visión*. Recuperado el 9 de diciembre de 2021, de [https://www.ingenieria.unam.mx/nuestra\\_facultad/mision\\_vision.php](https://www.ingenieria.unam.mx/nuestra_facultad/mision_vision.php)
- Pérez Porto, J. y Merino, M. (2008). *Definición de organización – Qué es, Significado y Concepto*. Definición.DE. Recuperado el 23 de octubre de 2021, de <https://definicion.de/organizacion/>
- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Historia*. Recuperado el 10 de diciembre de 2020, de <http://www.fi-a.unam.mx/~unica/presentacion/historia.php>
- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Misión*. Recuperado el 29 de noviembre de 2020, de <http://www.fi-a.unam.mx/~unica/presentacion/mision.php>
- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Políticas de calidad*. Recuperado el 29 de noviembre de 2020, de <http://www.fi-a.unam.mx/unica/presentacion/politica.php>

- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Valores*. Recuperado el 29 de noviembre de 2020, de <http://www.fi-a.unam.mx/unica/presentacion/valores.php>
- Unidad de Servicios de Cómputo Académico. (octubre de 2018). *Visión*. Recuperado el 29 de noviembre de 2020, de <http://www.fi-a.unam.mx/unica/presentacion/vision.php>
- Universidad Nacional Autónoma de México. (9 de diciembre de 2018). *Qué es la UNAM / Portal UNAM*. Recuperado el 9 de diciembre de 2021, de <http://www.fi-a.unam.mx/unica/presentacion/vision.php>

## CAPÍTULO 2 FUNDAMENTOS TEÓRICOS

- acens. (s.f). *Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar* [Archivo PDF]. <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- Alegsa. (2016). *Definición de Sybase*. <https://www.alegsa.com.ar/Dic/sybase.php>
- Ángeles, María del Pilar., Arellano Mendoza, L. P., Hernández Hernández, L., Lizárraga Ramírez, G. B., Muñoz Villalba, N y Rodríguez Campos, J. A. (2016). *Manual de prácticas del Laboratorio de Bases de Datos* [Archivo PDF]. UNAM. <http://odin.fi-b.unam.mx/salaD/practicasyBD.php>
- App Dynamics. (s.f). *Sybase ASE Monitoring / Database / AppDynamics*. <https://www.appdynamics.com/supported-technologies/database/sybase-ase>
- Appvizer. (s.f). *Software CAD*. <https://www.appvizer.es/organizacion-planificacion/cad>
- ARIMETRICS. (s.f). *Qué es Android – Definición, significado y ejemplos*. <https://www.arimetrics.com/glosario-digital/android>
- Arsys. (31 de enero de 2018). *Cuándo, para qué y por qué utilizar MariaDB – Blog de arsys.es*. <https://www.arsys.es/blog/programacion/mariadb/>
- Avitia Hinojosa, C. (31 de enero de 2016). *Comparación de Gestores de Bases de Datos* [Archivo PDF]. Academia.edu. [https://www.academia.edu/21674886/Comparacion\\_de\\_SGBD](https://www.academia.edu/21674886/Comparacion_de_SGBD)
- ayudaley. (s.f). *Bases de datos relacional ¿Qué es y sus características? | Ayuda Ley Protección Datos*. <https://ayudaleyprotecciondatos.es/bases-de-datos/relacional/>

- Badillo Bernal, D. (2015). *ESTUDIO COMPARATIVO DE LAS DISTRIBUCIONES LINUX ORIENTADO A LA SEGURIDAD DE REDES DE COMUNICACIÓN* [Tesis de Maestría, Pontificia Universidad Católica del Ecuador].  
<http://repositorio.puce.edu.ec/bitstream/handle/22000/10002/TESISV25-26-11-2015.pdf?sequence=1&isAllowed=y>
- Buttice, C. (21 de junio de 2021). *¿Qué es Oracle Database (Oracle DB)? – Definición de Technopedia*. Techopedia. Recuperado el 15 de diciembre de 2021, de <https://www.techopedia.com/definition/8711/oracle-database>
- Camps Paré, R., Casillas Santillán, L. A., Costal Costa, D., Gibert Ginestà, M., Martín Escofet, C. y Pérez Mora, O. (2005). *Bases de datos* [Archivo PDF]. UOC.  
<https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>
- CentOS. (2021). *Sobre CentOS*.  
<https://www.centos.org/about/>
- Christensson, P. (2006). *Definición del compilador*. TechLib.  
<https://techlib.net/definition/compiler.html>
- Christensson, P. (2006). *IDE (Dispositivo integrado de electrónica o entorno de desarrollo integrado) Definición*. TechLib. <https://techlib.net/definition/ide.html>
- Coordinación de Universidad Abierta y Educación a Distancia UNAM. (2017). *Componentes de una Computadora*.  
[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/909/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/909/mod_resource/content/1/contenido/index.html)
- Debian. (27 de agosto de 2021). *Debian – Razones para escoger Debian*. Recuperado el 6 de septiembre de 2021, de [https://www.debian.org/intro/why\\_debian](https://www.debian.org/intro/why_debian)
- Delgado, A. (26 de mayo de 2009). *Linux de pago | Costumer*. Consumer.  
<https://www.consumer.es/tecnologia/software/linux-de-pago.html>
- Equipo editorial. (16 de julio de 2021). *Sistema operativo – Concepto, usos, tipos, funciones y ejemplos*. concepto. <https://concepto.de/sistema-operativo/>
- Franklin Johnson, P. y José Miguel Rubio, L. (s.f). *Base de datos* [Archivo PDF].  
<http://zeus.inf.ucv.cl/~jrubio/docs/2009-01/INF%20340/Capitulo%20IV%20-%20Parte%201.pdf>
- Gabit. (2021). *¿Qué es iOS? | Gabit*.  
<http://www.gabit.org/gabit/?q=es/que-es-ios>

- García Martínez, D. (29 de enero de 2020). *Comparativa Sistemas gestores de bases de datos relacionales* [Archivo PDF]. DocerArgentina. <http://www.gabit.org/gabit/?q=es/que-es-ios>
- García, J. (30 de abril de 2013). *Ediciones y opciones Oracle 11G – mundoDB*. <http://mundodb.es/ediciones-y-opciones-oracle-11g>
- Gargiulo, G. (2017). *¿Qué es macOS? | Significado, +9 características, versiones [2021]*. Swift Latino. <https://swiftlatino.com/macOS/>
- Graph Everywhere. (s.f). *Tipos de bases de datos | Clasificación por contenido y modelo. GRAPH EVERYWHERE*. <https://www.grapheverywhere.com/tipos-bases-de-datos-clasificacion/>
- HostingPedia. (24 de enero de 2019). *MySQL: ¿Qué es? Características, Ventajas y Desventajas*. <https://hostingpedia.net/mysql.html>
- HostingPedia. (7 de febrero de 2019). *PostgreSQL: ¿Qué es? Características, Ventajas y Desventajas*. <https://hostingpedia.net/postgresql.html>
- HostingPedia. (7 de junio de 2017). *¿Qué es Ubuntu? Historia, Características, Ventajas y Desventajas*. <https://hostingpedia.net/ubuntu-linux.html>
- Hughes, A. (abril de 2021). *¿Qué es Microsoft SQL Server? – Definición en Whatls.com. ámbito de aplicación*. ComputerWeekly.es. <https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server>
- INFORMÁTICA ACTUAL Y SENCILLA. (s.f). *EDITORES DE TEXTO – INFORMÁTICA ACTUAL Y SENCILLA*. <https://sites.google.com/site/informaticaactualysencilla/software/editores-de-texto>
- IONOS. (13 de octubre de 2020). *¿Qué es un depurador? – Definición, funcionamiento y ámbito de aplicación - IONOS*. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/depurador/>
- Javier Pérez Díaz, A. (s.f). *Definición IDE Enciclopedia Proyecto AjpdSoft. AJBD soft*. <https://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=1051>
- Julian Lugo, J. (10 de enero de 2020). *Diseño asistido por computador (CAD) para el diseño industrial*. IdeasDI. <https://ideasdi.com/recursos/disenio-asistido-por-computador/>

- Maida, E. G. y Pacienza, J. (2015). *Metodologías de desarrollo de software* [Tesis de Licenciatura, Universidad Católica Argentina].  
<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- malisper. (31 de mayo de 2017). *Postgres MVCC – malisper.me*.  
<https://malisper.me/postgres-mvcc/>
- Martínez, A. (26 de enero de 2021). *¿Qué es Software? » Su Definición y Significado [2021]*. ConceptoDefinición. Recuperado el 10 de agosto de 2021, de  
<https://conceptodefinicion.de/software/>
- Máxima Uriarte, J. (25 de junio de 2019). *10 Características del Sistema Operativo Linux*. Características. Recuperado el 2 de septiembre de 2021, de  
<https://www.caracteristicas.co/sistema-operativo-linux/>
- Máxima Uriarte, J. (20 de diciembre de 2019). *Unix: historia, usos, ventajas, desventajas y características*. Características. Recuperado el 1 de septiembre de 2021, de <https://www.caracteristicas.co/unix/>
- Máxima Uriarte, J. (21 de noviembre de 2020). *Windows: historia, evolución y características*. Características. Recuperado el 1 de septiembre de 2021, de <https://www.caracteristicas.co/sistema-operativo-windows/>
- Merchán Manzano, O. (2018). *DISEÑO DE BASE DE DATOS* [Archivo PDF]. Universidad del Azuay.  
<http://publicaciones.uazuay.edu.ec/index.php/ceazuay/catalog/download/5/5/37-2?inline=1>
- Microsoft. (4 de octubre de 2012). *Buffer Management | Microsoft Docs*.  
[https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/aa337525\(v=sql.105\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/aa337525(v=sql.105)?redirectedfrom=MSDN)
- Microsoft. (7 de marzo de 2016). *Announcing SQL Server on Linux – The Official Microsoft Blog*. <https://blogs.microsoft.com/blog/2016/03/07/announcing-sql-server-on-linux/>
- Microsoft. (27 de diciembre de 2021). *SQL Server Management Studio (SSMS) – SQL Server Management Studio (SSMS) | Microsoft Docs*.  
<https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?redirectedfrom=MSDN&view=sql-server-ver15>

- Moes, T. (s.f). *¿Qué es un antivirus? La definición y los 5 ejemplos principales*. SoftwareLab.org. <https://softwarelab.org/es/que-es-un-antivirus/>
- Ñeco Gómez, S. y Pelegrín Pérez, Y. (s.f). *INFORMIX* [Archivo PDF]. <https://iessanvicente.com/colaboraciones/Informix.pdf>
- Olarte Gervacio, L. (30 de agosto de 2017). *Clasificación de software de sistemas y aplicaciones - Conogasi*. conogasi. <http://conogasi.org/articulos/clasificacion-de-software-de-sistemas-y-aplicaciones/>
- Oracle. (2017). *Data Concurrency and Consistency*. <https://docs.oracle.com/database/121/CNCPT/consist.htm#CNCPT1333>
- Oracle. (17 de noviembre de 2021). *Lifetime-support-technology-069183.pdf* [Archivo PDF]. <https://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf>
- Oracle. (2018). *mysql-datasheet.en.pdf* [Archivo PDF]. <https://www.mysql.com/products/enterprise/mysql-datasheet.en.pdf>
- Oracle. (2020). *mysql-10-razones.pdf* [Archivo PDF]. <http://www.zak.com.mx/assets/mysql-10-razones.pdf>
- Oracle. (2022). *Oracle Configuration Manager Documentation*. [https://docs.oracle.com/cd/E49269\\_01/index.htm#:~:text=Oracle%20Configuration%20Manager%20\(OCM\)%20is,better%20service%20to%20the%20customers](https://docs.oracle.com/cd/E49269_01/index.htm#:~:text=Oracle%20Configuration%20Manager%20(OCM)%20is,better%20service%20to%20the%20customers)
- Ordenadores y Portátiles. (s.f). *¿Qué es un Software de Diagnóstico?*. <https://www.ordenadores-y-portatiles.com/software-de-diagnostico/>
- Ortega, A. (11 de septiembre de 2015). *Software de Entretenimiento by Arturo Ortega*. Prezi. <https://prezi.com/etvzeyjw6o0o/software-de-entretenimiento/>
- openSUSE wiki. (30 de agosto de 2020). *Ciclo de vida – openSUSE Wiki*. Recuperado el 19 de septiembre de 2021, de [https://es.opensuse.org/Ciclo\\_de\\_vida](https://es.opensuse.org/Ciclo_de_vida)
- parzibyte. (17 de diciembre de 2018). *Límite de bases de datos y tablas en MySQL – Parzibyte's blog*. <https://parzibyte.me/blog/2018/12/17/limite-bases-de-datos-tablas-mysql/>
- Pérez, A. (19 de marzo de 2015). *REQUERIMIENTOS DE INSTALACIÓN DE MySQL by aurora perez*. <https://prezi.com/zqymc2vdz2j2/requerimientos-de-instalacion-de-mysql/>
- Pérez Porto, J. y Gardey, A. (2009). *Definición de ofimática – Qué es, Significado y Concepto*. Definición.DE. Recuperado el 21 de agosto de 2021, de <https://definicion.de/ofimatica/>

- Pomeyrol, J. (4 de octubre de 2017). *SUSE Linux Enterprise y openSUSE permitirán migrar de una a otra* » *MuyLinux*. MuyLinux.  
<https://www.muylinux.com/2017/10/04/suse-linux-enterprise-opensuse-migracion/>
- Portillo, S. (28 de abril de 2020). *MySQL: Como obtener todas las tablas de una base de datos - TutorialesIT*. <https://tutorialesit.com/mysql-como-obtener-todas-las-tablas-de-una-base-de-datos/>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 8.0: CREATE USER*.  
<https://www.postgresql.org/docs/8.0/sql-createuser.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 9.6: Numeric Types*.  
<https://www.postgresql.org/docs/9.6/datatype-numeric.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: Appendix K. PostgreSQL Limits*.  
<https://www.postgresql.org/docs/current/limits.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: Chapter 16. Installation from Binaries*. <https://www.postgresql.org/docs/current/install-binaries.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: Chapter 17. Installation from Source Code*. <https://www.postgresql.org/docs/current/installation.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: 5.7. Privileges*.  
<https://www.postgresql.org/docs/14/ddl-priv.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: 8.3. Character Types*.  
<https://www.postgresql.org/docs/current/datatype-character.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: 22.1. Database Roles*.  
<https://www.postgresql.org/docs/14/database-roles.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 15: 21.5. Password Authentication*.  
<https://www.postgresql.org/docs/15/auth-password.html>
- PostgreSQL. (2022). *PostgreSQL: Support*.  
<https://www.postgresql.org/support/>
- Rojas Gaytán, H. (7 de agosto de 2020). *administra sistemas operativos 156 – 1.2.8 DRIVERS Ó CONTROLADORES DE DISPOSITIVOS*. *administra sistemas operativos 156*. Recuperado el 10 de agosto de 2021, de  
<https://sites.google.com/site/admonsistemasoperativos156/unidades-de-administra-sistemas-operativos/1-unidad-introduccion-a-los-sistemas-operativos/2-1-describir-el-funcionamiento-de-un-sistema-operativo/1-3-6-controladores>

- Sergio, G.B. (6 de julio de 2016). *31 distribuciones Linux*. SoloLinux. <https://www.sololinux.es/31-distribuciones-linux/>
- Silberschatz, A., Korth, H. F. y Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS* [Archivo PDF]. McGRAW-HILL/INTERAMERICANA DE ESPAÑA. [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Silberschatz.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Silberschatz.pdf)
- SISTEMAS. (s.f). *Definición de Compilador – Significado y definición de Compilador*. <https://sistemas.com/compilador.php>
- SLACKO. (26 de septiembre de 2019). *¿POR QUÉ COMPLICARSE LA VIDA USANDO SLACKWARE LINUX?* <https://slacko.home.blog/2019/07/26/por-que-complicarse-la-vida-usando-slackware-linux/>
- Software para TODO. (s.f). *Software de Simulación: Que es y Ejemplos (2021)*. <https://softwareparatodo.com/software-de-simulacion/>
- Stallings, W. (2005). *SISTEMAS OPERATIVOS* [Archivo PDF]. Pearson Education. <http://www.epet3.edu.ar/pampint/file/Tpampin3038.pdf>
- TECNOLOGÍAS INFORMACIÓN. (2018). *Bases de datos Access – Características y Archivos*. <https://www.tecnozero.com/office-365/access/>
- ThingWorx 8. (s.f). *Utilización de Microsoft SQL Server como proveedor de persistencia*. [http://support.ptc.com/help/thingworx\\_hc/thingworx\\_8\\_hc/es/index.html#page/ThingWorx/Help/Composer/DataStorage/PersistenceProviders/UsingMSSQLServerAsThePersistenceProvider.html](http://support.ptc.com/help/thingworx_hc/thingworx_8_hc/es/index.html#page/ThingWorx/Help/Composer/DataStorage/PersistenceProviders/UsingMSSQLServerAsThePersistenceProvider.html)
- Vega Gestión. (31 de octubre de 2017). *Los tres grandes tipos de software – Vega Gestión*. <https://vegagestion.es/los-tres-grandes-tipos-software/>

### **CAPÍTULO 3 IMPLEMENTACIÓN DE LA BASE DE DATOS**

- Cane, B. (5 de enero de 2017). *Turning PostgreSQL with pgbench | Cloudbees Blog*. <https://www.cloudbees.com/blog/tuning-postgresql-with-pgbench>
- DELL. (2022). *Servidor en torre PowerEdge T150 | Dell México*. [https://www.dell.com/mx/empresas/p/poweredge/t150/pd?oc=pe\\_t150\\_15330\\_mx\\_1&model\\_id=poweredge-t150](https://www.dell.com/mx/empresas/p/poweredge/t150/pd?oc=pe_t150_15330_mx_1&model_id=poweredge-t150)
- Economides, N. y Katsamakos, E. (2006). Linux vs. Windows: A Comparison of Application and Platform Innovation Incentives for Open Source and Proprietary Software Platforms [Archivo PDF]. En Bitzer, J. y Schröder, P. J. (Ed.), *The economics of open*

*source software development*. (pp. 207-218). Elsevier Science Inc..

<https://core.ac.uk/download/pdf/7068572.pdf>

- Eulises Ortiz, A. (8 de octubre de 2018). *¿Qué es CentOS? ¿Qué significa y que hace? ¿Es Linux? ¿RedHat? ¿Open Source?* | Blog HostDime Latinoamérica, servidores dedicados. HostDime Blog. <https://www.hostdime.la/blog/que-es-centos-que-significa-y-que-hace-es-linux-redhat-open-source/>
- Eulises Ortiz, A. (8 de octubre de 2018). *Ventajas de CentOS por sobre otras distros de Linux en los servidores web* | Blog HostDime Argentina, servidores dedicados. HostDime Blog. <https://www.hostdime.com.ar/blog/ventajas-de-centos-por-sobre-otras-distro-de-linux-en-los-servidores-web/>
- Hernández, J. (30 de enero de 2021). *¿Qué es Linux?» Su definición y Significado*. Concepto-Definición. Recuperado el 6 de marzo de 2021, de <https://conceptodefinicion.de/linux/>
- HostingPedia. (2 de junio de 2017). *CentOS - ¿Qué es? Características, Ventajas y Desventajas*. <https://hostingpedia.net/centos-linux.html>
- HostingPedia. (7 de febrero de 2019). *PostgreSQL: ¿Qué es? Características, Ventajas y Desventajas*. <https://hostingpedia.net/postgresql.html>
- Kili, A. (15 de octubre de 2020). *How to Backup and Restore a PostgreSQL Database*. <https://www.tecmint.com/backup-and-restore-postgresql-database/>
- Ondara, W. (25 de febrero de 2024). *How to Create a Superuser in PostgreSQL | Cherry Servers*. <https://www.cherryservers.com/blog/postgres-superuser#:~:text=A%20Postgres%20superuser%20is%20a,and%20modifying%20data%20C%20among%20others>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: ALTER DATABASE*. <https://www.postgresql.org/docs/current/sql-alterdatabase.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: GRANT*. <https://www.postgresql.org/docs/current/sql-grant.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: pgbench*. <https://www.postgresql.org/docs/current/pgbench.html>
- PostgreSQL. (2022). *PostgreSQL: Documentation: 14: psql*. <https://www.postgresql.org/docs/current/app-psql.html>

- PostgreSQL Tutorial. (2011). *Import CSV File Into PostgreSQL Table*. <https://www.postgresqltutorial.com/postgresql-tutorial/import-csv-file-into-postgresql-table/>
- Ramírez, I. (31 de enero de 2020). *Máquinas virtuales: qué son, cómo funcionan y cómo utilizarlas*. Xataka MÉXICO. Recuperado el 11 de mayo de 2022, de <https://www.xataka.com/especiales/maquinas-virtuales-que-son-como-funcionan-y-como-utilizarlas>
- Santiago. (1 de febrero de 2018). *PostgreSQL vs MySQL – ¿Cuál es el mejor? - Guiadev*. <https://guiadev.com/postgresql-vs-mysql/#:~:text=PostgreSQL%20brinda%20opciones%20m%C3%A1s%20complejas,frente%20a%20su%20cl%C3%A1sico%20rival>
- Segovia, J. (26 de septiembre de 2017). *Alta disponibilidad con PostgreSQL – TodoPostgreSQL*. <https://www.todopostgresql.com/alta-disponibilidad-postgresql/>
- StackScale. (13 de julio de 2020). *CentOS Linux: distribución, versiones y CentOS Stream*. <https://www.stackscale.com/es/blog/centos-linux/>
- Thakar, P. (21 de octubre de 2021). *Complete Guide on How to Import and Export CSV Files to PostgreSQL*. skyvia. <https://blog.skyvia.com/complete-guide-on-how-to-import-and-export-csv-files-to-postgresql/>

## **Apéndice A. INSTALACIÓN DEL SOFTWARE DE LA BASE DE DATOS**

- Cuesta Viñolo, A. (11 de mayo de 2010). *VirtualBox. Configuración de la conexión de red. - Adictos al trabajo*. <https://adictosaltrabajo.com/2010/05/11/virtualbox/>
- De Luz, S. (19 de enero de 2024). *Cómo configurar las opciones de red de VirtualBox para las VMs*. RZ redes zone. Recuperado el 17 de mayo de 2024, de <https://www.redeszone.net/tutoriales/redes-cable/configuracion-red-maquina-virtual-virtualbox/>
- El Instalador. (6 de enero de 2021). *Cómo instalar PHP en CentOS 7 [2024] PASO A PASO*. Cómo instalar.... [https://comoinstalar.me/como-instalar-php-en-centos-7/#google\\_vignette](https://comoinstalar.me/como-instalar-php-en-centos-7/#google_vignette)
- El Instalador. (6 de julio de 2021). *Cómo instalar PostgreSQL en CentOS 7 [2022] Paso a paso*. Cómo instalar.... <https://comoinstalar.me/como-instalar-postgresql-en-centos-7/>
- Mills, H. (24 de agosto de 2021). *How To Install the Apache Web Server on CentOS 7 / DigitalOcean*. <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-centos-7>

- Nettix. (s.f). *Cómo configurar nuestra de red en CentOS utilizando la herramienta nmtui* – Nettix Perú. <https://www.nettix.com.pe/documentacion/administracion/linux-administracion/como-configurar-nuestra-de-red-en-centos-7-utilizando-comandos>

## Referencias

- Badillo, D. (2015). Comparativa entre distribuciones Linux [Tabla]. Recuperado de <http://repositorio.puce.edu.ec/bitstream/handle/22000/10002/TESISV25-26-11-2015.pdf?sequence=1&isAllowed=y>
- Camps, R. (2005). Organización de un sistema de base de datos [Figura]. Recuperado de <https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>
- Merchán, O. (2018). Arquitectura de tres niveles de abstracción de datos [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). Instancia de la tabla ARTÍCULO [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). Instancia de la tabla ARTÍCULO1. [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). Instancia de la tabla FABRICANTE1 [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). (a) Relación no normalizada. (b) Relación en primera forma normal [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). Relación usuario, aplicación de bases de datos, SGBD y base de datos [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Merchán, O. (2018). Tablas en tercera forma normal [Figura]. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/download/5/5/37-2?inline=1>
- Pinterest. (s.f). Equipos de cómputo contemplados para implementar la base de datos (a) Equipo utilizado. [Figura]. <https://pin.it/2CgRC1O>
- Pinterest. (s.f). Equipos de cómputo contemplados para implementar la base de datos (b) Equipo sugerido. [Figura]. <https://www.pinterest.com.mx/pin/64246732176114729/>
- Sergio G. B. (2020). Conectar un Android o iPhone en Arch y derivados [Figura]. Solo Linux. <https://www.sololinux.es/conectar-un-android-o-iphone-con-manjaro-y-arch/>
- Silberschatz, A., Korth, H. F. y Sudarshan, S. (2002). Ejemplo de base de datos relacional [Figura]. Recuperado de [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Silberschatz.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Silberschatz.pdf)

Silberschatz, A., Korth, H. F. y Sudarshan, S. (2002). Ejemplo de diagrama E-R [Figura]. Recuperado de [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Silberschatz.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Silberschatz.pdf)

Silberschatz, A., Korth, H. F. y Sudarshan, S. (2002). Un sistema informático centralizado [Figura]. Recuperado de [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Silberschatz.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Silberschatz.pdf)

Unidad de Servicios de Cómputo Académico. (mayo de 2024). Imagen del logo de la cabecera de la página web / Interfaz de usuario. [Figura]. <https://www.ingenieria.unam.mx/unica/>

Unidad de Servicios de Cómputo Académico. (octubre de 2018). Ubicación de las Salas de Cómputo UNICA. [Figura]. <http://www.fi-a.unam.mx/unica/instalaciones/index.php>