

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

“Diseño de Bases de Datos Distribuidas con MySQL Workbench 5.2 CE”

Autor: Erick Arturo Rodríguez Valencia

**Tesis presentada para obtener el título de:
Ingeniero en Sistemas Computacionales**

**Nombre del asesor:
ALDO ISRAEL SANDOVAL MONROY**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación “Dr. Silvio Zavala” que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo “Reconocimiento-NoComercial-SinObraDerivada”, se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





**UNIVERSIDAD
VASCO DE QUIROGA**

FACULTAD DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

“Diseño de Bases de Datos Distribuidas con MySQL
Workbench 5.2 CE”

MONOGRAFÍA

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTA

ERICK ARTURO RODRIGUEZ VALENCIA

ASESOR

ALDO ISRAEL SANDOVAL MONROY

CLAVE: 16PSU0049F

ACUERDO: LIC100846

MORELIA, MICHOACÁN

AGOSTO-2015

A mis padres Pablo y Rosario, quienes me han apoyado desinteresadamente en todo momento y han sido un ejemplo de vida para lograr ser una mejor persona y padre de familia.

A mi esposa Anabel, quien estuvo conmigo y compartió incontables noches de desvelo y esfuerzo.

A mi hija Romina, quien me regalo su tiempo y me dio fortaleza durante esta etapa de mi vida.

A mi amigo y jefe Rafa Terán, quien me guió para continuar mi desarrollo personal, profesional y académico.

ÍNDICE GENERAL

RESUMEN	IV
PLANTEAMIENTO DEL PROBLEMA	V
ANTECEDENTES	VII
OBJETIVOS	VIII
ALCANCES Y LIMITACIONES	IX
JUSTIFICACIÓN	X
CAPÍTULO 1 INTRODUCCIÓN	1
CAPÍTULO 2 MARCO TEÓRICO	3
2.0 INTRODUCCIÓN.....	3
2.1 DISEÑO DE LA BASE DE DATOS DISTRIBUIDA	4
2.2 TABLAS, ENTIDADES Y RELACIONES	7
2.3 PROCEDIMIENTOS Y FUNCIONES	12
CAPÍTULO 3 REVISIÓN TÉCNICA	15
3.0 DESCRIPCIÓN Y FUNCIONALIDAD DE LA BASE DE DATOS.....	15
3.1 NORMALIZACIÓN.....	16
3.2 CALIDAD DE LA INFORMACIÓN	17
3.3 FRAGMENTACIÓN Y CONCURRENCIA	19
3.4 SEGURIDAD Y HERRAMIENTAS CON LAS QUE CONTARÁ EL SISTEMA:	21
3.5 INVERSIÓN E INFRAESTRUCTURA	25
CAPÍTULO 4 CONCLUSIONES Y TRABAJO FUTURO	31
BIBLIOGRAFÍA	34
ÍNDICE DE FIGURAS	36
GLOSARIO DE TÉRMINOS	37
APÉNDICE 1 LISTADO DEL PROGRAMA	40

RESUMEN

El objetivo del presente proyecto es presentar el análisis sobre el diseño de bases de datos distribuidas haciendo uso de MySQL Workbench 5.2 CE considerando el modelo de bases de datos relacionales, así como desarrollar una mejor estructura de las tablas (relaciones), filas (tuplas) y columnas (atributos) por medio de la normalización y calidad de la información, todo ello con el fin de lograr un mejor diseño y planeación para el desempeño y performance entre las bases de datos que estén en comunicación por medio de enlaces o conexiones virtuales a través de redes y diferentes manejadores de bases de datos que se encuentran en distintos servidores de diferentes zonas geográficas, de tal manera que permitan obtener resultados consultando o modificando datos de manera separada pero como si estuviéramos conectados de manera local a la base de datos para gestionar la información y almacenar los datos en diferentes registros, cuidando la integridad y seguridad de la información para garantizar la disponibilidad de la información y obtener el conocimiento sobre todas aquellas transacciones operativas que nos lleven a realizar una mejor toma de decisiones al momento de consultar los datos almacenados, y con ello lograr alcanzar las metas al ejecutar procesos de negocio de manera eficaz y más eficiente en cualquier organización o empresa.

PLANTEAMIENTO DEL PROBLEMA

La iniciativa y propuesta de diseño para implementar una base de datos distribuida en una Universidad es debido a la necesidad de comprender el manejo de una base de datos del modelo relacional, así como su infraestructura para tener un mejor manejo y gestión de la información que utilizan el personal de Control Escolar, Vinculación, Bibliotecas y algunas de las otras áreas administrativas como Tesorería, Becas y Seguros con el fin de dar solución a algunos conflictos en el manejo de la información de sus alumnos, docentes y empleados, así como de algunos registros en pagos que pueden no verse reflejados en la matrícula de los estudiantes.

Por mencionar algunos ejemplos, durante el trámite de constancias de estudio o solicitud de documentos del cual no llevan registro y que puedes volver a pagar a menos que cuentes con el comprobante del pago, así como el uso y manejo del archivo muerto y que se quedan como datos históricos pero que no están documentados en una base de datos dejando los documentos aislados en una habitación, o bien en bibliotecas donde se lleva una bitácora de entradas y salidas de ejemplares a través de una hoja de registro donde se llena información para generar la solicitud pero no se tiene un record en un sistema, pues al implementar una base de datos será posible llevar un mejor control en el registro de préstamos de libros para calcular la duración que lleva algún alumno con el ejemplar e identificar adeudos para enviar notificaciones de rezago a los alumnos para su entrega oportuna y evitar recargos o penalizaciones.

Por ello, es indispensable recolectar, procesar, almacenar y distribuir información de una manera ordenada y clasificada para generar una base de conocimientos actualizada y lograr tener acceso de manera sustancial a la resolución de problemas en un menor tiempo posible al compartir de manera segura la base de conocimientos para las diferentes áreas de personal que requiera dicha información, así como a las nuevas generaciones, de manera que se puedan clasificar los eventos para cada una de las áreas involucradas.

Es así que durante el desarrollo de una base de datos es esencial cuidar el diseño y su estructura, adicional de contar con la tecnología adecuada para poder almacenar los registros y consultar de manera rápida la información, ya que el desarrollar una buena base de datos, tener la estructura y diseño adecuados, así como mantener las relaciones de las tablas y conceptos, nos permitirá tener una ventaja competitiva por sobre cualquier otra organización, por lo que debemos considerar la flexibilidad, escalabilidad y velocidad con la que a futuro deberá funcionar la base de datos distribuida para poder manejar grandes cantidades de información.

Como la información es poder el gestionarla será siempre un elemento clave para el éxito de las organizaciones y de las personas, ya que el tener acceso a la información nos permite consultar, analizar, discriminar e interpretar datos, lo que se convierte en una parte fundamental para generar nuevas estrategias, proyectos, establecer objetivos, determinar necesidades de los usuarios y clientes de un sistema, con el fin de poder tomar decisiones a través de la consulta de información clara, consistente y fiable.

ANTECEDENTES

Actualmente la mayoría de las Universidades en el país ya cuentan con un sistema de bases de datos con la cual gestionan y llevan el control sobre los programas, facultades, empleados, alumnos, profesores y calificaciones por mencionar algunos, y es a través de la base de datos y el sistema que lo interpreta lo que les permite a directivos, administradores, docentes, alumnos y demás personas quienes tienen acceso a la misma el consultar los registros y datos almacenados y relacionados con el fin de obtener información que les ayude a determinar el estado actual de sus materias, sus programas, las calificaciones, el número de empleados, alumnos y profesores quienes se encuentran activos e inactivos dentro de la Universidad, así como obtener información personal y de contacto, generar estadísticas y reportes que conlleven a la toma de decisiones, por lo que es importante que dicha información almacenada en la base de datos sea segura, integra, clara, coherente, sustancial y se encuentre protegida para evitar el mal uso de la información, ya que pudiera poner en riesgo la confidencialidad e integridad de los datos e información almacenada. Por ello la propuesta del desarrollo de la base de datos distribuida es dar a conocer las ventajas de que un buen diseño deberá considerar los tipos de fragmentación horizontal (mismo sitio), vertical (diferente sitio) o mixta (ambos tipos) con el fin de dar solución sobre algunos conflictos en el manejo y protección de la información de las bases de datos para con sus alumnos, docentes, empleados y demás personas que cuenten con acceso a ellas.

OBJETIVOS

El objetivo del proyecto es analizar y proponer el diseño para desarrollar una base de datos distribuida para una Universidad, como propuesta de solución a los problemas de administración de la información, con el fin de alcanzar un mejor manejo y gestión de la información que utiliza el personal de Control Escolar, Vinculación, Bibliotecas y algunas de las otras áreas administrativas como Tesorería, Becas y Seguros.

Objetivos específicos:

- Comprender los conceptos y modelos de bases de datos relacionales utilizadas por las empresas para almacenar información.
- Obtener información y conocimiento sobre todos aquellos datos almacenados en la base de datos (como transacciones operativas), con el fin de posicionarse como líderes en el mercado.
- Comprender las tecnologías de la información requeridas por las empresas para lograr alcanzar sus metas de negocio.
- Comprender la manera en que las bases de datos permiten ejecutar los procesos de negocio de la empresa de una manera eficaz y más eficiente.

ALCANCES Y LIMITACIONES

Principales Ventajas y alcances al utilizar una base de datos distribuida:

- Poder innovar y mantenerse a la vanguardia.
- Tener un mejor manejo y gestión de la información que utiliza el personal de áreas administrativas, funcionales y operativas de la empresa u organización.
- Poder almacenar y copiar datos en múltiples sitios con el fin de mejorar el rendimiento, fiabilidad y disponibilidad de la información siendo esto transparente para cada uno de los usuarios que utilicen el sistema, tanto en la distribución de los datos así como en la conectividad de las redes de comunicación.
- Lograr una mejor distribución de los datos a través de la fragmentación al llevar a cabo una estructura de base de datos igual (horizontal) o diferente (vertical) para cada uno de los sitios.
- Contar con la tecnología adecuada para poder almacenar la información y consultar los registros de manera rápida.
- Tener una ventaja competitiva por sobre cualquier otra organización, al considerar la flexibilidad, escalabilidad y velocidad con la que a futuro deberá funcionar para poder manejar grandes cantidades de datos.

Limitaciones y desventajas al utilizar una base de datos distribuida:

- El costo de implementación es alto.
- La gestión sobre el manejo y control de los datos puede ser compleja.
- Cuidar y garantizar la replicación de datos, así como la integridad de la información.
- Mayor complejidad de interacción entre los diferentes sitios ya que las bases de datos no necesariamente están centralizadas.

JUSTIFICACIÓN

El diseño conceptual, lógico y físico es fundamental para el desarrollo de las nuevas tecnologías de bases de datos ya que permite agregar características para realizar diseños más complejos, considerando diagramas y esquemas que permitan lograr una mayor integridad relacional a través de una estructura correcta de las relaciones, tuplas y atributos.

Es a través del diseño de bases de datos que se lleva a cabo el proceso de normalización, lo que permite generar dependencias funcionales entre el diseño conceptual y los esquemas relacionales del diseño lógico y físico, ya que es aquí donde se lleva a cabo la definición de la estructura, los datos, sus conceptos, las entidades, los vínculos y demás relaciones que conllevan a la integridad de las bases de datos, así como poder identificar atributos, posibles riesgos para prevenir, posibles mejoras, unificar información, clasificar eventos, definir manejadores de bases de datos para poder estandarizar el lenguaje que se va a utilizar, así como determinar hardware y software necesario para la implementación, etc.

Al hacer uso de bases de datos distribuidas es posible acceder a los datos desde una o varias localidades para llevar a cabo la ejecución de transacciones y/o consulta de información desde cualquier zona geográfica, logrando así una mejor distribución de los datos por medio de la fragmentación con el fin de garantizar su integridad y evitar la pérdida de información al no tener los datos de manera centralizada, y con ello es posible tener una mejor utilización de la información, tener mayor eficiencia y rendimiento en los procesos de negocio, aumentar el grado de concurrencia por medio de subconsultas al estar los sistemas en paralelo, lograr mayor seguridad de los datos ya que no se almacenan en un solo sitio (nodo) evitando el acceso a usuarios no autorizados, permitiendo una mejor gestión de la información para lograr una ventaja competitiva por sobre otras empresas u organizaciones.

Capítulo 1

INTRODUCCIÓN

Gracias a las tecnologías y sistemas de información hoy en día es posible automatizar los procesos y sistematizar información de una manera más eficiente, al reunir y procesar los datos para su posterior análisis y poder establecer estrategias que nos permitan obtener una ventaja competitiva sobre otras organizaciones.

Es por ello que estas herramientas básicas, en conjunto con las telecomunicaciones, son imprescindibles durante la toma de decisiones y el diseño de estrategias para poder alcanzar el desarrollo, control y éxito en las organizaciones.

Al hacer uso de las tecnologías y sistemas de información mediante la implementación de bases de datos distribuidas es posible almacenar, recuperar, transportar, manipular o discriminar información para tener un mayor control y gestión de la información y así tomar una mejor decisión que favorezca a la empresa u organización.

El potencial del uso de bases de datos es muy grande puesto que el contar con un buen diseño de la misma, así como de su estructura y componentes físicos será parte sustancial del éxito de dichas organizaciones.

Uno de los lenguajes más importantes y utilizados por diferentes manejadores de bases de datos (DBMS), como Oracle, Postgre y Microsoft Azure, es el lenguaje de consulta estructurado SQL el cual ha sido el lenguaje estándar para el manejo de la información, ya que es compatible en cualquier dispositivo y sistemas operativos manteniendo las relaciones entre las tablas y columnas durante las consultas generadas para poder obtener información de una manera más rápida y eficiente.

Es así que durante el desarrollo de una base de datos debemos considerar un esencial cuidado en el diseño y su estructura, así como contar con la tecnología adecuada para poder almacenar los registros y consultar de manera rápida la información, ya que el desarrollar una buena base de datos, tener la estructura y diseño adecuados, así como mantener las relaciones de las tablas y conceptos, nos permitirá tener una ventaja competitiva por sobre cualquier otra organización, por lo que debemos considerar la flexibilidad, escalabilidad y velocidad con la que a futuro deberá funcionar para poder manejar grandes cantidades de datos.

Como se menciona en (Elmasri R. y., 2007) “El diseño de una aplicación nueva para una base de datos existente o el diseño de una base de datos nueva empieza con una fase denominada definición de requisitos y análisis. Estos requisitos son documentados en detalle y transformados en un diseño conceptual que se puede representar y manipular mediante algunas herramientas computarizadas, de modo que en una implementación de base de datos puedan mantenerse, modificarse y transformarse fácilmente y para el cual hacemos uso del modelo denominado Entidad-Relación. El diseño después se convierte en un diseño lógico que se puede expresar en un modelo de datos implementado en un DBMS comercial.”

Capítulo 2

Marco Teórico

2.0 Introducción

En este capítulo, se presenta la base teórica para poder comprender el diseño y estructura de una base de datos distribuida considerando el modelo relacional, y para el cual se definió como alternativa principal el uso de MySQL WorkBench 5.2 CE como DBMS y herramienta con la cual se llevará a cabo el diseño, desarrollo e implementación de la base de datos gracias a sus características y ventajas, pues es una de las plataformas más estables, ya que contiene opciones para el manejo de modelos de datos relacionales y su representación gráfica de las tablas, así como también posee herramientas para la administración de servidores y cuenta con un manejador para desarrollar las bases de datos a través de la definición y estructura relacional de las tablas (por medio de sus filas y columnas), adicional de ser una aplicación muy práctica e intuitiva ya que su interfaz gráfica es muy amigable, se muestran en la figura 2.1.

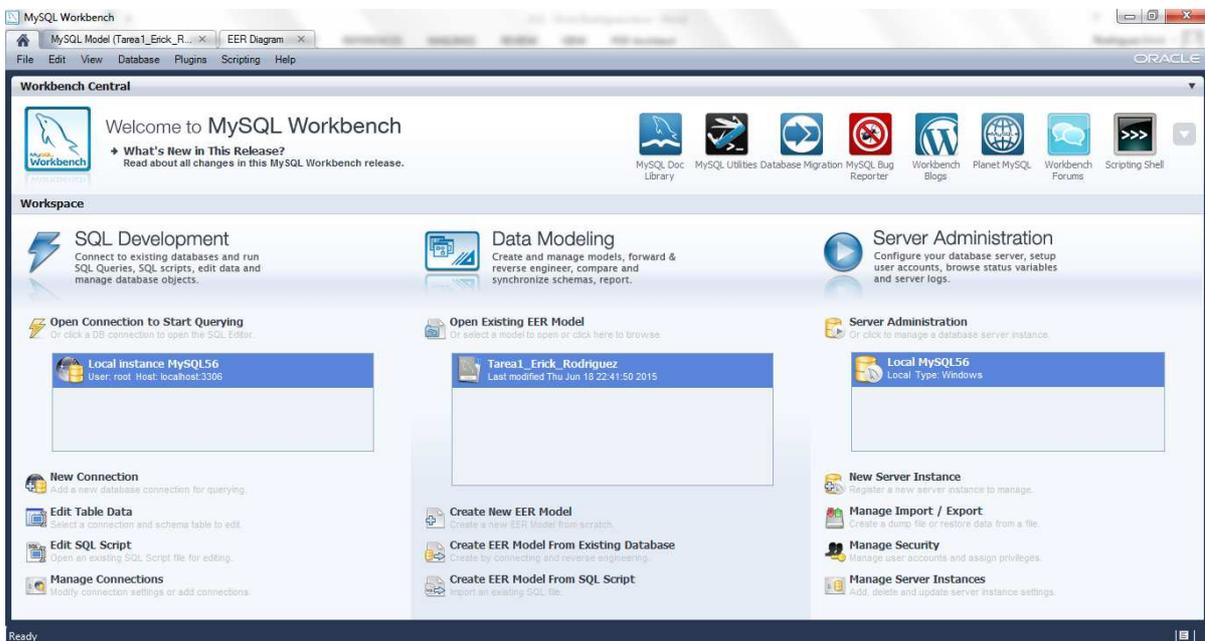


Figura 2.1 MySQL WorkBench 5.2 CE con el cual se llevó a cabo el diseño y desarrollo de la DB

2.1 Diseño de la Base de Datos Distribuida

Con la aplicación de la base de datos distribuida será posible dar solución a problemas para el procesamiento de datos y archivos, logrando una mejor coordinación, sincronización, integración, consistencia y organización de la información para su procesamiento, logrando reducir el tiempo al consultar la información requerida, lo cual permitirá su uso a cualquier persona quien cuente con un usuario con acceso y privilegios y con ello tener un mayor control y seguridad a través de la comunicación entre los diferentes equipos de cómputo que efectúan los procesos dentro de la Universidad, ya que al hacer uso de bases de datos distribuidas es posible acceder a los datos desde una o varias localidades para llevar a cabo la ejecución de transacciones y/o consulta de información desde cualquier zona geográfica, así como mantener la información almacenada de manera segura, íntegra, clara, coherente, sustancial y protegida para evitar el mal uso de los datos.

A continuación se presenta el diseño del esquema lógico mediante el modelo conceptual y físico de la base de datos distribuida con base en la figura 2.2, para poder ubicar información en sitios físicamente distintos para un ambiente cliente/servidor, donde los clientes están en localidades remotas presentando el número de sitios y servidores que se requerirán:

Modelo Conceptual

Sitio Cliente (Usuarios del Sistema)

- No conocen la ubicación física de los datos.
- Las aplicaciones y programas son transparentes.
- Son independientes a la réplica de los datos.
- Pueden tener diferente Hardware y Software.
- Opera en diferentes redes de comunicaciones y dispositivos.

Sitio Local (Sitio3 – Cualquier Campus de la Universidad)

- Tiene su propio Nodo y su propia autonomía.
- Hay cooperación entre nodos para el acceso distribuido de datos.
- Cada sitio posee su propio SMBD que controla el nodo.
- Las operaciones de acceso utilizan recursos locales.
- Evita fallo de todo el sistema en caso de caída del nodo.
- La información está fragmentada para obtener mayor rendimiento.
- Los datos y operaciones locales son almacenados para reducir el tráfico de red.
- Las aplicaciones funcionan por medio de copias locales (no de otro sitio).
- El objeto replicado está disponible siempre y cuando exista una copia.
- Pueden tener diferente Hardware y Software.
- Opera en diferentes redes de comunicaciones y dispositivos.
- La disponibilidad debe ser 24x7x365 (exceptuando mantenimiento).

Sitio Central (Sitio 4 – Matriz de la Universidad)

- Tiene su propio Nodo y DBMS que controla el nodo y es tolerante a fallos.
- Hay cooperación entre nodos para el acceso distribuido de datos.
- Las operaciones de acceso utilizan recursos locales y mantiene su atomicidad.
- Almacena y recupera información de los campus independientes.
- Mantiene backups de manera incremental.
- Si se cae este nodo los demás sistemas continúan trabajando.
- El protocolo commit se utiliza durante la ejecución de una transacción.
- Dispone de mecanismos de optimización en consultas para reducir el tráfico de red.
- Gestión y control de concurrencia y recuperación de la información.
- Pueden tener diferente Hardware y Software.
- Opera en diferentes redes de comunicaciones y dispositivos.
- La disponibilidad debe ser 24x7x365 (exceptuando mantenimiento).

Modelo Físico

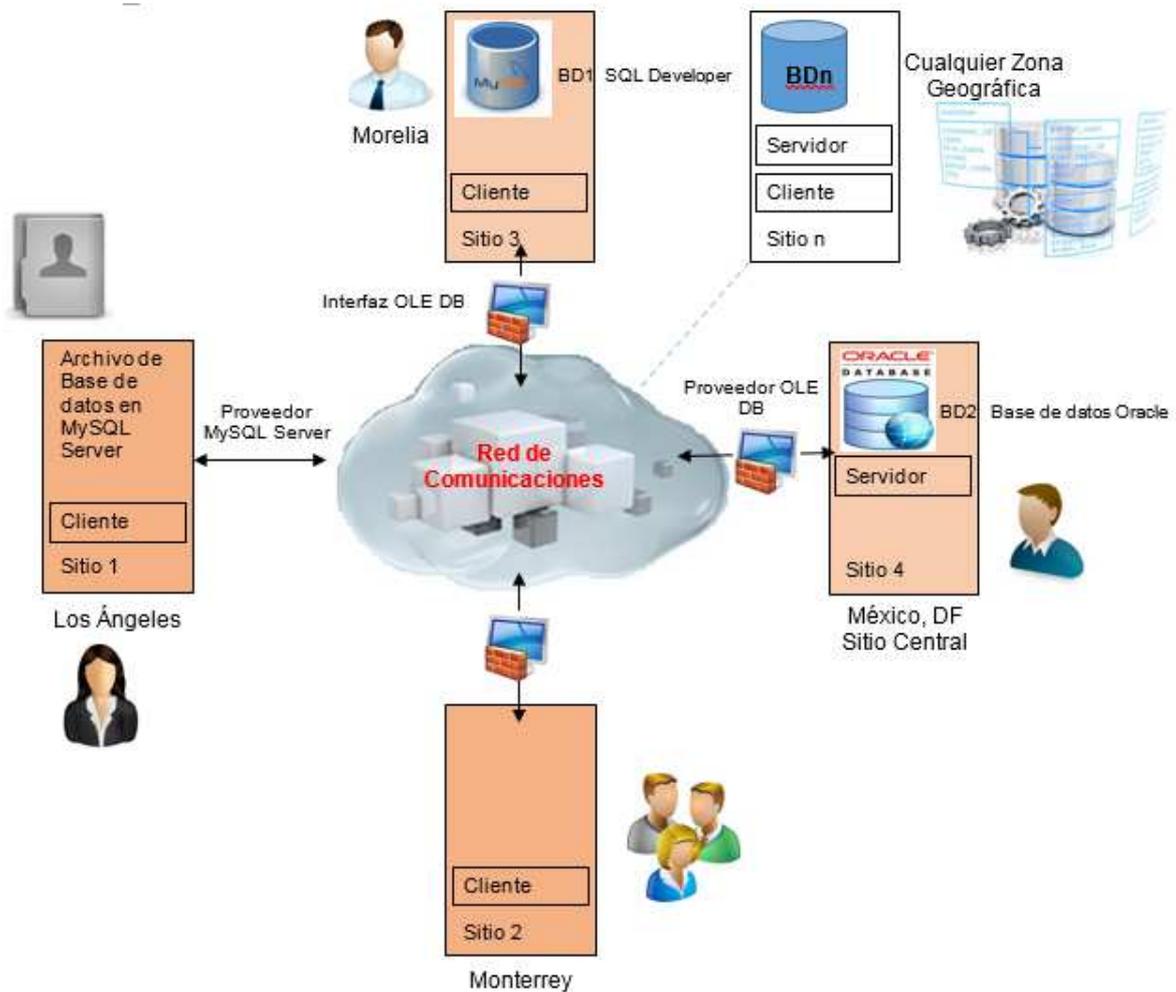


Figura 2.2 Modelo Físico Cliente/Servidor de la Base de Datos Distribuida

Tomando como referencia una Universidad en la cual puede requerirse una base de datos distribuida con entidades como Empleados, Alumnos, Profesores, Proveedores, Activos Fijos, Campus, Bibliotecas, Costos, Becas, Perfiles, Facultades, Programas, Materias, Calificaciones, etc., mientras que las bases de datos instaladas en cada campus puede requerir solamente entidades como Alumnos, Profesores, Perfiles, Bibliotecas, Facultades, Programas, Materias, Calificaciones, etc., por lo que no considera necesario que tenga información sobre costos ya que el pago de colegiaturas se hace en línea o por pago bancario directo al sitio central quien administra y asigna recursos a los diferentes campus. En detalle,

el sitio 1 en Los Ángeles corresponde a un usuario quien hace uso del sistema, el sitio 2 en Monterrey corresponde a varios usuarios quienes hacen uso del sistema, el sitio 3 en Morelia corresponde al Campus Morelia quienes poseen una Base de Datos con información de los alumnos de esa zona geográfica para gestión del mismo campus (y para la cual fue desarrollado este proyecto), el sitio 4 en México, DF corresponde a la central de la Universidad quienes poseen la base de datos matriz pero solo con cierta información sobre facultades, campus, programas, materias, alumnos, etc., a la cual se conectan los alumnos para hacer uso de del servidor de correos o blackboard para poder estudiar en línea.

2.2 Tablas, entidades y relaciones

Es sumamente importante considerar en el diseño conceptual, lógico y físico los diagramas y esquemas que permitan lograr una mayor integridad relacional a través de una estructura correcta de las relaciones (tablas), tuplas (filas) y atributos (columnas) en una base de datos distribuida, con el fin de alcanzar un mejor manejo y gestión de la misma, para tener mayor control de los datos y dar solución a los problemas de administración de la información, como se muestra en la figura 2.3 y 2.4.



Figura 2.3 Modelo de Datos Relacional

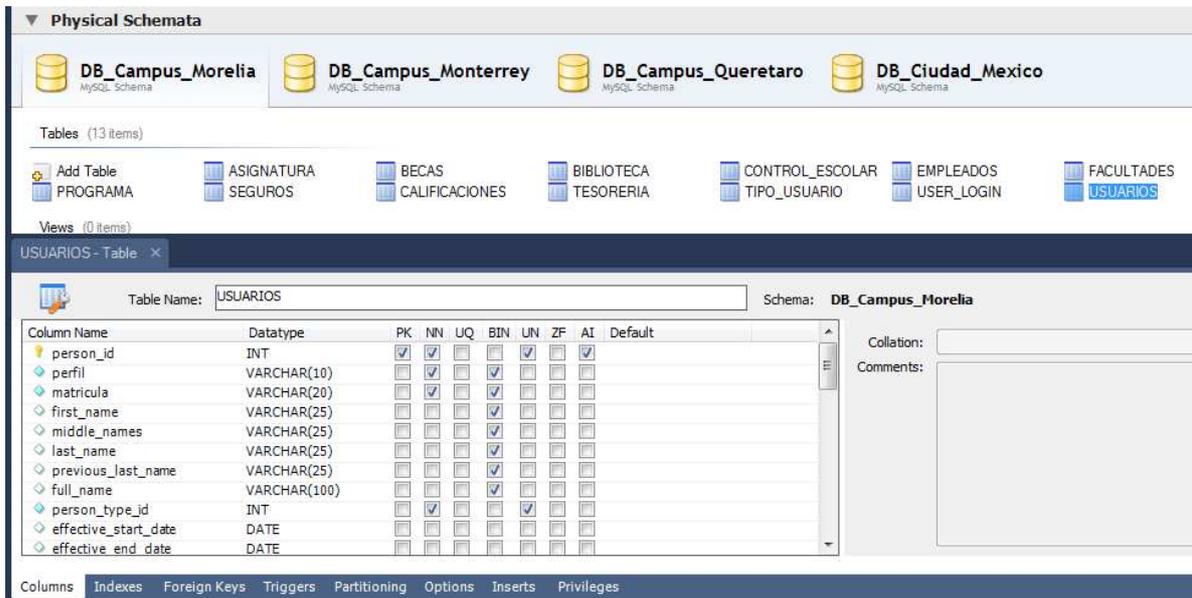


Figura 2.4 Esquema Físico de la Base de Datos Distribuida

Es así que se deben definir las principales tablas con sus entidades y relaciones entre las tuplas y atributos que formarán la base de datos, ya que durante el diseño debemos proveer las tablas necesarias de la base de datos para almacenar la información de la Universidad como pueden ser las calificaciones, el nombre de la maestría e institución educativa a la que pertenecen los alumnos y profesores. Como una muestra de algunas de las tablas para este proyecto, se presenta la siguiente propuesta sobre el análisis de algunas tablas que, posteriormente, serán creadas y definidas a través de MySQL WorkBench 5.2 CE, las cuales se muestran en le figura 2.5 y 2.6.

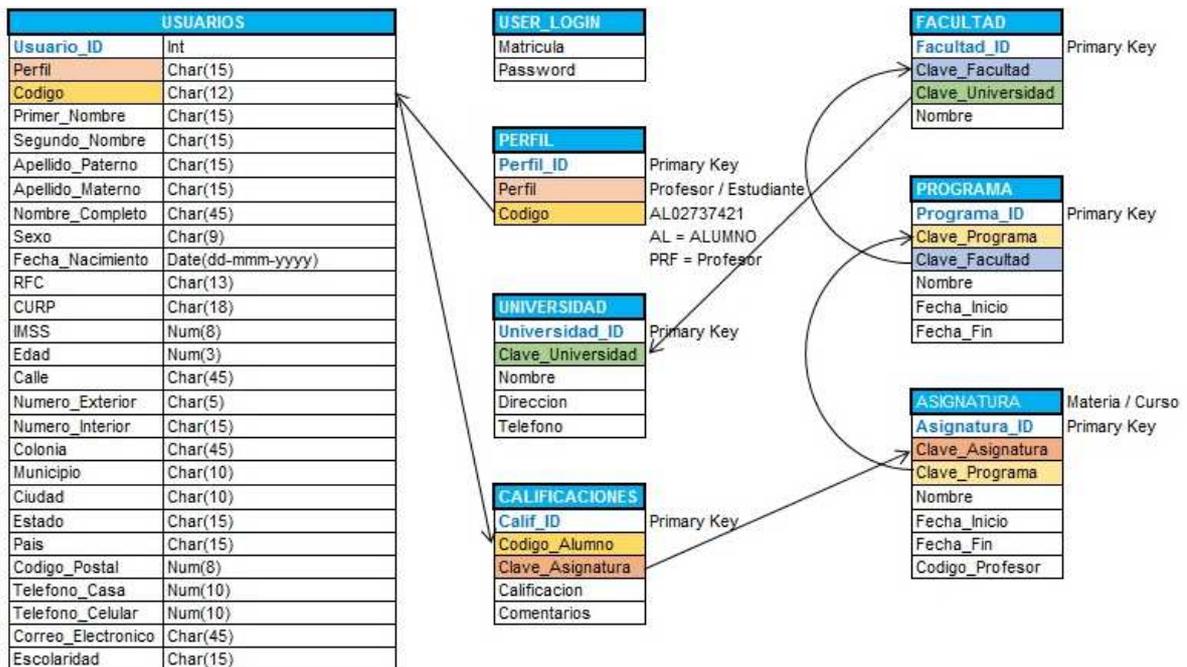


Figura 2.5 Análisis del diseño y desarrollo de la DB

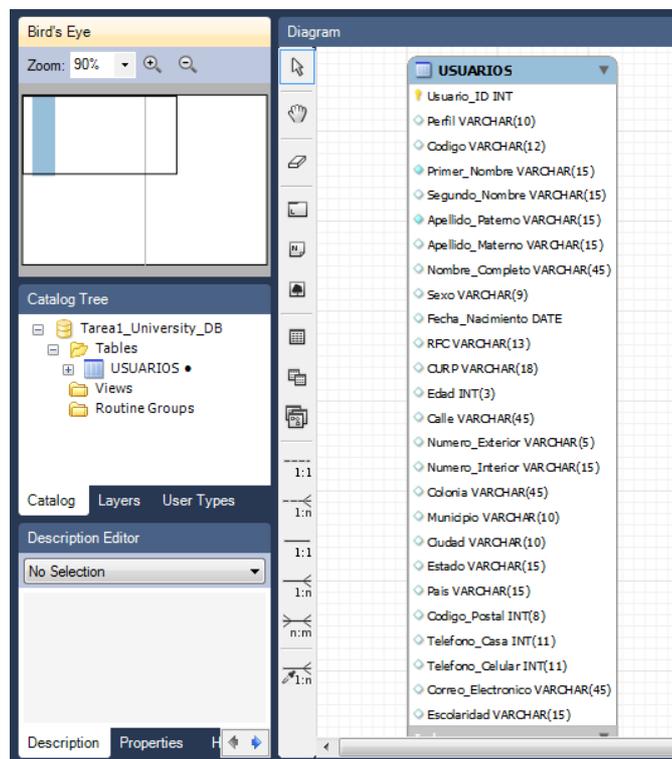


Figura 2.6 Diseño y estructura de la tabla USUARIOS

Para la creación de las tablas, entidades y relaciones, se presentarán algunas de las instrucciones como CREATE y TRIGGERS así como las sentencias INSERT, DELETE y UPDATE necesarias para agregar, borrar o cambiar valores durante la creación y diseño de la base de datos relacional, así como también para las consultas por medio de la sentencias SELECT (ver figura 2.7) y sub-consultas JOIN y EXIST entre distintas tablas adicionando clausulas y condiciones WHERE, GROUP BY y HAVING para obtener información de los registros y datos almacenados en las base de datos.

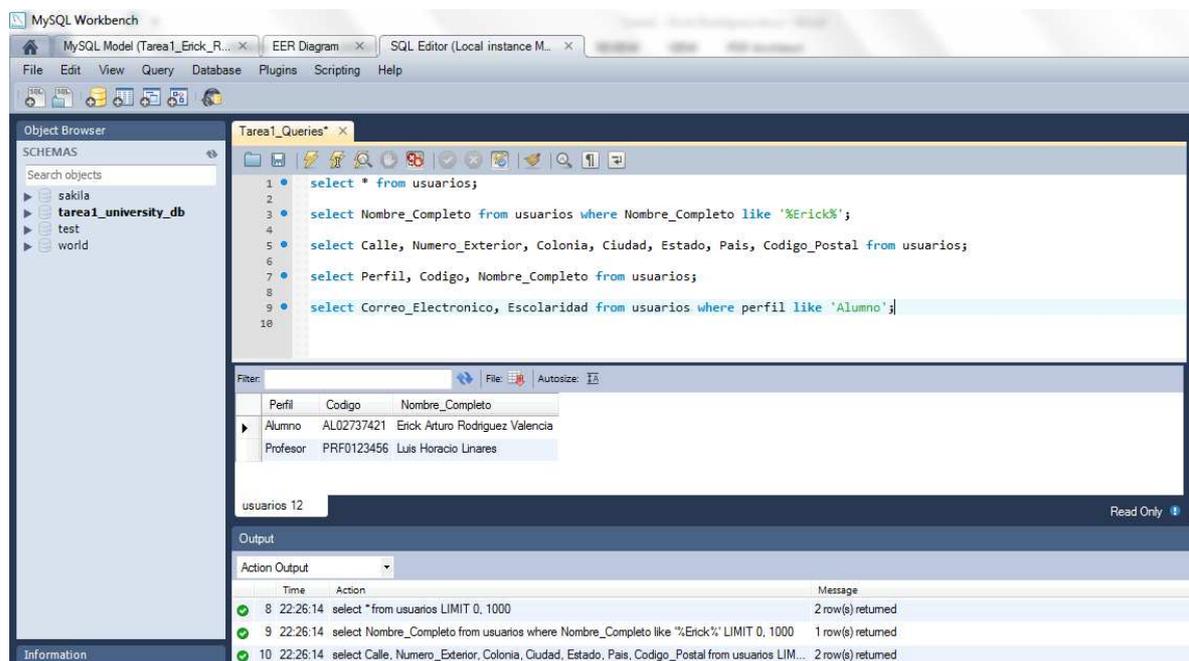


Figura 2.7 Sentencias SELECT sobre la tabla USUARIOS desde MySQL Worckbench

Como ejemplo de Trigger, validar que el alumno no tenga más de 3 libros en préstamo:

```

create trigger insertar AFTER INSERT ON prestamo_libro
FOR EACH ROW
insert into prestamo_libro (Numero_Prestamo, Fecha_Inicio, Fecha_Fin,
Dias_Prestamo, Status_ID, Status, Libro_ID_FK, Nombre_Libro, Alumno_ID_FK,
Matricula)
values ('ABC456', '2015-06-27', '2015-06-30', 3, 1, 'Vigente', 2, 'An
introduction to Database Systems', 1, 'AL02737421');

```

En la figura 2.8 se presenta la relación, estructura y esquemas entre algunas de las tablas de las bases de datos que representen la información de ALUMNO, PRESTAMO_LIBRO y LIBRO para el caso de la biblioteca:

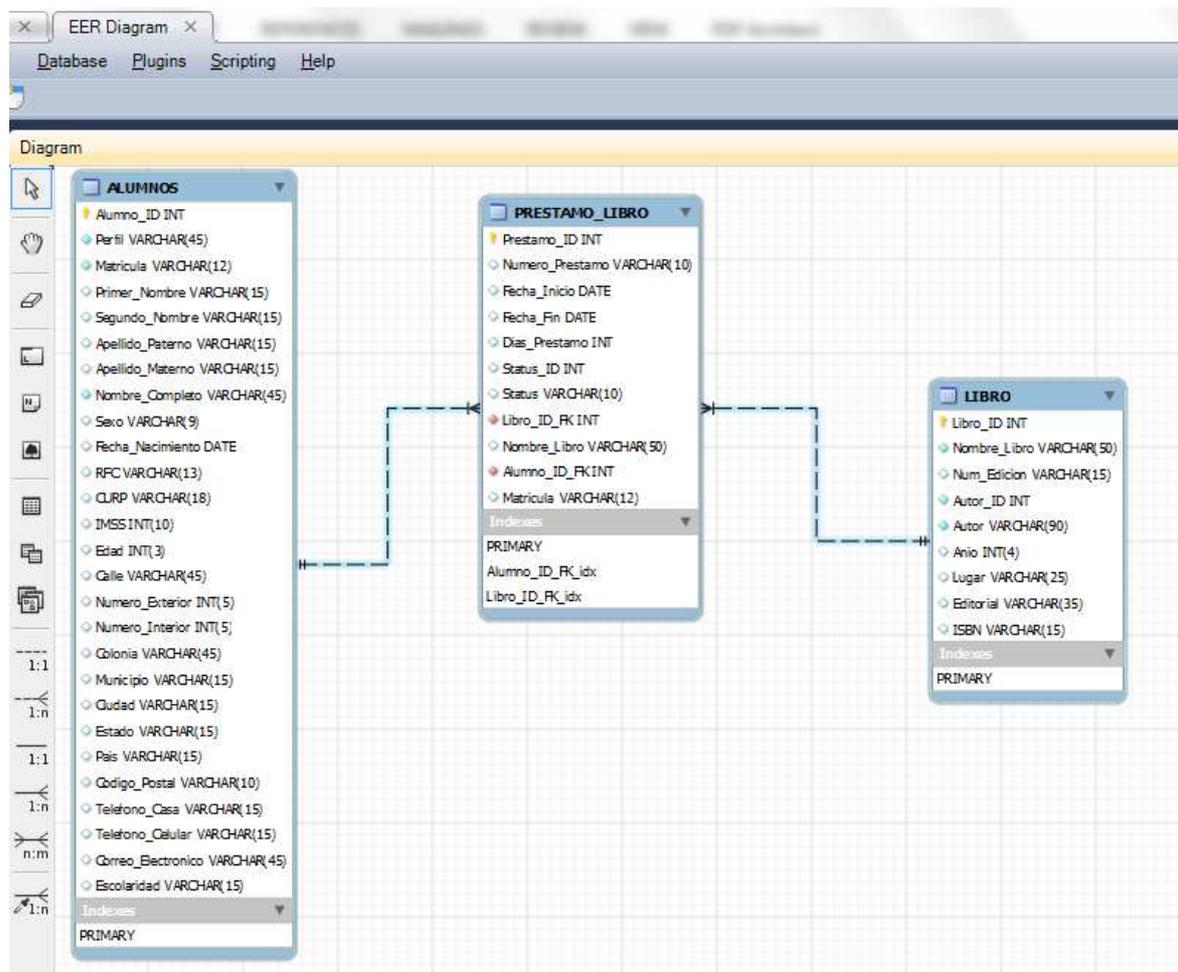


Figura 2.8 Relación de las tablas Alumnos, Prestamo_Libro y Libro necesarias para Biblioteca

El desarrollar una mejor estructura de las tablas, filas y columnas por medio de la normalización permite tener mejor calidad de la información para lograr el mejor desempeño y performance entre las bases de datos ya que evita la duplicidad de los datos.

2.3 Procedimientos y Funciones

La diferencia entre un procedimiento y una función es que el procedimiento no necesariamente debe retornar algún valor, mientras que en la función siempre deberá retornar uno o varios valores. Adicionalmente, el procedimiento utiliza delimitadores y solo es invocado mediante una instrucción CALL para poder realizar una operación de entrada – IN, salida – OUT o ambas – INOUT como consultas SELECT o modificaciones en las tablas a través de las instrucciones INSERT, DELETE o UPDATE, mientras que la función es invocada cuando la solicites mediante una expresión y sus parámetros son considerados como entrada – IN y utiliza la cláusula RETURN para devolver el valor o conjunto de valores, siendo necesario definir el tipo de datos que será devuelto.

Como se menciona en (Elmasri R. y., 2007) “Los procedimientos almacenados y las funciones son módulos del programa que se compilan la primera vez que son ejecutados, por lo que en las ejecuciones posteriores se minimizará el tiempo de respuesta al no realizar nuevamente el proceso de compilación. Los procedimientos almacenados y las funciones se utilizan particularmente para la implementación de reglas y forman parte de la definición global de la base de datos, se les invoca correctamente cuando ocurren ciertas condiciones.”

La sintaxis de un procedimiento vs la función también es diferente, como lo ejemplifico con la figura 2.9.

PROCEDIMIENTO	FUNCION
<pre> delimiter // CREATE PROCEDURE simpleproc (OUT param1 INT) BEGIN SELECT COUNT(*) INTO param1 FROM t; END// delimiter: CALL simpleproc(@a); SELECT @a; +-----+ @a +-----+ 3 +-----+ 1 row in set (0.00 sec) </pre>	<pre> CREATE FUNCTION hello (s CHAR(20)) RETURNS CHAR(50) DETERMINISTIC RETURN CONCAT('Hello, ',s,'!'); SELECT hello('world'); +-----+ hello('world') +-----+ Hello, world! +-----+ 1 row in set (0.00 sec) </pre>

Figura 2.9 Diferencia entre un procedimiento almacenado y una función

La principal ventaja al utilizar y centralizar los procedimientos, funciones y triggers en el SMBD es provocar que aumente la velocidad de la aplicación y facilite el mantenimiento del código SQL.

Es necesario comprender cuando debemos hacer uso de un procedimiento o una función como mecanismos de control de acceso a la base de datos, los procedimientos los utilizamos a través de delimitadores para enviar valores usando variables de salida y son llamados para realizar consultas o modificaciones de datos mediante los comandos Select, Insert, Delete y Update y no necesariamente deben devolver algún valor de salida; mientras que las funciones siempre regresan valores y es necesario definir el tipo de dato que van a retornar. Los procedimientos los podemos usar para mejorar el performance de la base de datos pues solo se compila la primera vez que se define y es posible reutilizar código evitando inconsistencias, reducen el tráfico en la red y proporcionan flexibilidad, se utilizan para obtener mayor seguridad y cuidar la integridad de la información ya que se pueden controlar los procesos y actividades de las operaciones en los objetos de las bases de datos sobre los usuarios y programas, de modo que solo aquellos usuarios que cuenten con permisos puedan hacer uso de él, pues no hay acceso directo a las tablas de la

base de datos por parte de los usuarios. El uso de parámetros en los procedimientos ayuda a prevenir un ataque mal intencionado al código. No obstante, en caso de haber cambios entre las tablas de los procedimientos se deberán crear nuevamente para no afectar el rendimiento.

El uso de triggers nos permite definir reglas para poder manipular y modificar los datos de la base de datos para cuidar y mantener la calidad de la información, ya que es posible obtener información sobre la ejecución de las operaciones realizadas en las tablas y generar mensajes de prevención e historial sobre el acceso a la base de datos, de tal manera que sea posible cuidar la integridad de la información evitando la corrupción de los datos, así como la relación y dependencia funcional para normalizar la información dentro de la base de datos.

Capítulo 3 REVISIÓN TÉCNICA

3.0 Descripción y Funcionalidad de la Base de Datos

Para poder diseñar una base de datos debemos comprender plenamente el modelo de los usuarios, es decir la representación de las actividades de los usuarios e identificar las cosas que se almacenarán en la base de datos, por lo que es necesario documentar todos los requerimientos necesarios para definir la estructura y las relaciones entre todas las cosas y así poder inferir y desarrollar las interfaces y reportes acordes a las necesidades de los usuarios. Ya que al comprender el manejo de una base de datos del modelo relacional, como se muestra en la figura 3.1 [1], será posible gestionar la información y almacenar los datos en diferentes registros, cuidando la integridad y seguridad de los datos para garantizar la disponibilidad de la información.

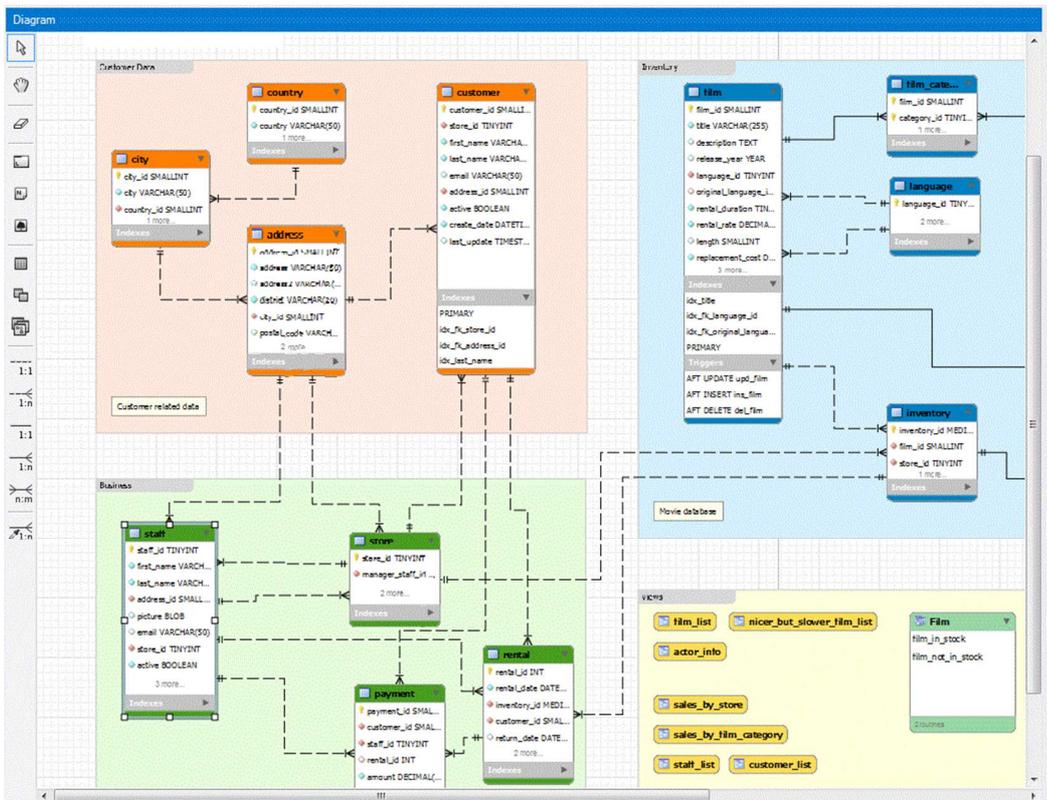


Figura 3.1 Modelo Relacional, obtenido de (Corporation, 2015) solo con fines informativos

3.1 Normalización

Durante el diseño de la estructura de la base de datos debemos contemplar también el diseño conceptual para mantener la relación entre entidades y poder manipular posteriormente el diseño de las relaciones y sus vínculos, como se muestra en la figura 3.2 [3].

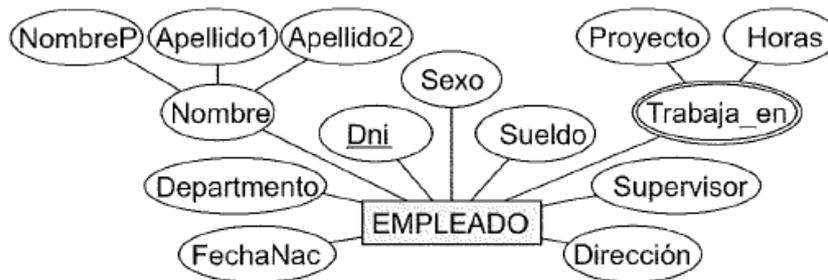


Figura 3.2 Normalización, imagen obtenida de (Elmasri R. y., 2007) solo con fines informativos

Para ello debemos considerar los diferentes componentes como tablas, entidades y vínculos, filas (tuplas) y columnas (atributos), y poder llevar a cabo la normalización de la base de datos para no duplicar información, como se muestra en la figura 3.3.

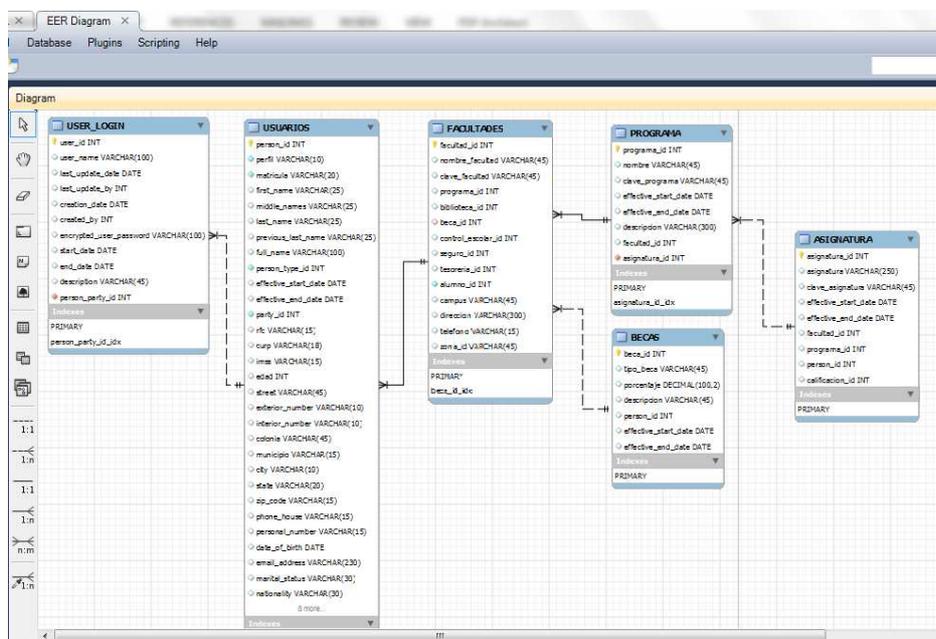


Figura 3.3 Normalización de la base de datos para no duplicar información

3.2 Calidad de la Información

La calidad es una cuestión de percepción para considerar si algún producto o servicio es mejor que otro con base en sus propiedades, atributos, características y costos, lo cual te permitirá tomar una decisión para poder elegir siempre y cuando cumpla con tus necesidades.

El asegurar la calidad de la información en una base de datos relacional es importante porque nos permite mantener un sistema eficaz y eficiente con información veraz, clara y confiable, evitando información redundante, inconsistente e incorrecta durante la consulta de información o presentación de reportes ante directivos y público en general para la toma de decisiones. Ya que en caso contrario puede ocasionar problemas al tener un gran volumen de información almacenada, pues al llegar a consultarla podríamos estar adquiriendo información falsa generando conflictos.

Al considerar el tema de calidad de la información debemos tomar en cuenta la definición de los datos, así como de las especificaciones en el diseño para establecer las restricciones de clave y de integridad referencial, ya que los sistemas que utilizan bases de datos deben garantizar que la información se encuentre disponible las 24hrs los 365 días, sea confiable y segura durante los resultados de las consultas mismas que deberán ser eficientes y rápidas para generarlas en un lapso de tiempo corto. Adicional que se debe contar con un plan de acción sobre el control de concurrencia para el acceso y modificación de los datos cuando hay varios usuarios con acceso a la base de datos, para prever nuevos cambios/requisitos y adaptaciones futuras, para realizar auditorías y conocer los estados o cambios en la información y quienes han tenido acceso a la misma, todo ello con el fin de mantener y salvaguardar la información así como evitar incoherencias en la misma.

Al momento de construir consultas en lenguajes SQL podemos enfrentarnos a ciertos problemas como deficiencia en la información si es que la base de datos tuvo un mal diseño, o bien los datos pueden ser inútiles, pueden ser redundantes e innecesarios,

pueden no tener integridad en la información, puede ser información no actualizada, puede haber anomalías e inconsistencia en la estructura de las tablas, se puede desconocer la estructura de las tablas, puedes tener poca experiencia para generar consultas dificultando el tiempo y desarrollo de los queries, pueden ejecutar una consulta en un lapso de tiempo mayor debido a una unión incorrecta entre las tablas. Por ello es necesario adaptarnos a métodos, técnicas y procesos durante y después del diseño de la base de datos, como puede ser el Data Profiling.

Por ello, el asegurar la calidad de los datos en una base de datos relacional es importante porque nos permite mantener un sistema eficiente, ya que la información debe ser clara y confiable para evitar información redundante o incorrecta, pues en caso contrario puede ocasionar problemas al tener un gran volumen de información almacenada la cual sea inconsistente, por lo cual al llegar a consultarla podríamos estar adquiriendo información falsa generando conflictos durante la toma de decisiones o durante la presentación y generación de reportes.

De acuerdo a (Elmasri R. y., 2007) “En ocasiones el utilizar restricciones específicas para definir estados de relación e indicar una dependencia funcional se implementa como una herramienta para analizar la calidad de los diseños de relación y para "normalizar" las relaciones que mejoran su calidad. Por ello es necesario medir formalmente por qué un conjunto de agrupaciones de atributos en un esquema de relación es mejor que otro.”

Al hacer uso de bases de datos relacionales es posible asegurar la calidad de los datos al implementar restricciones desde el DBMS (de dominio, de unicidad, de integridad referencial y reglas de negocio) para evitar que un usuario no pueda realizar ciertas funciones o introducir ciertos datos, con el fin de tener una base central por el cual deberán pasar todos los cambios de información, ya que el DBMS posee mayor capacidad de ejecución y solo es necesario codificar una sola vez, a menos que desde la aplicación se imponga una regla lo cual puede generar pérdida

de tiempo e inconsistencias en las mismas reglas por lo que se debe tener cuidado en estos casos.

Tipos de restricciones:

De Dominio: Son conjuntos de valores que un atributo o campo poseen, las cuales tienen un componente semántico y uno físico. Por ejemplo, que en determinado momento se guarde un dato en un campo y que dicho campo sea de tipo alfanumérico de 35 posiciones.

De Unicidad: Se aplican mediante el DBMS para crear estructuras de datos con atributos necesarios para verificar rápidamente la unicidad a través de llaves primarias o índices con el atributo UNIQUE.

De Relación Referencial: Se aplican por medio de llaves externas, siempre y cuando primero exista un registro en la tabla donde hace referencia la llave foránea. Por ejemplo, no es posible registrar libros en la tabla Biblioteca si no existen categorías y materias a las cuales estarán asociados.

De relación de cardinalidad: Son las reglas de negocio y se codifican desde la aplicación o desde el DBMS por medio de triggers.

3.3 Fragmentación y Concurrencia

La base de datos distribuida llevará el control de la información a través de una fragmentación vertical u horizontal dependiendo del sitio en el cual se almacena o consulta la información sobre sus alumnos, personal docente, facultades, programas, asignaturas, calificaciones y demás información que deseen registrar en la universidad. Al utilizar la fragmentación es posible garantizar la integridad y evitar la pérdida de información al no tener los datos de manera centralizada para tener mayor eficiencia y rendimiento en los procesos de negocio y poder aumentar el grado de concurrencia por medio de subconsultas al estar los sistemas en paralelo, así

como lograr mayor seguridad de los datos ya que no se almacenan en un solo sitio (nodo) evitando el acceso a usuarios no autorizados.

Como se menciona en (Tecnología, 2015) “Existen tres tipos de fragmentación: vertical, horizontal y mixta. Los dos primeros son los principales. Fragmentación horizontal: Agrupa los registros de una relación. Se generan especificando un predicado que imponga una restricción a las tuplas de la relación. Fragmentación vertical: Agrupa los atributos de una relación que son utilizados de manera conjunta por las transacciones de mayor importancia. Fragmentación mixta: También se denomina fragmentación híbrida. El proceso de partición de la relación hace uso de la fragmentación horizontal y también de la fragmentación vertical.”

Las figuras 3.4 y 3.5 muestran un ejemplo de la información que contendrán cada uno de los sitios (tablas o fragmentos de tablas) y el tipo de fragmentación diseñada para las bases de datos:

FRAGMENTACION VERTICAL

Empleados		Empleado_Morelia		Empleado_Monterrey		Empleado_Cancun	
Empleado_ID	Int	Empleado_ID	0001	Empleado_ID	0010	Empleado_ID	0100
Perfil	Varchar(10)	Perfil	Empleado	Perfil	Empleado	Perfil	Empleado
Matricula	Varchar(12)	Matricula	EMP01MLM	Matricula	EMP10MTY	Matricula	EMP100CNC
Cedula	Varchar(30)	Cedula	null	Cedula	null	Cedula	null
Nombre_Completo	Varchar(45)	Nombre_Completo	Jose Francisco Perez Segura	Nombre_Completo	Ruben Jaramillo Preciado	Nombre_Completo	Anabel Sanchez Medina
RFC	Varchar(13)	RFC	PESJ770423	RFC	JAOR821212	RFC	SAMA850101
CURP	Varchar(18)	CURP	PESJ770423HMNDLR06	CURP	JAOR821212HMNDRL08	CURP	SAMA850101MGDRL10
Fecha_Nacimiento	Date(dd-mm-yyyy)	Fecha_Nacimiento	23-ABR-1977	Fecha_Nacimiento	12-DIC-1982	Fecha_Nacimiento	01-01-1985
Direccion	Varchar(50)	Direccion	null	Direccion	null	Direccion	null
Telefono	Varchar(10)	Telefono	null	Telefono	null	Telefono	null
Correo_Electronico	Char(45)	Correo_Electronico	null	Correo_Electronico	null	Correo_Electronico	null
Escolaridad	Char(15)	Escolaridad	Maestria	Escolaridad	Doctorado	Escolaridad	Maestria

Figura 3.4 Fragmentación Vertical

Facultades	
Campus	Varchar(15)
Campus ID	Int Primary Key
Direccion	Varchar(50)
Telefono	Varchar(10)
Area	Varchar(20)

FRAGMENTACION HORIZONTAL

Campus	Morelia
Campus ID	0001
Direccion	Bld. Bosques de Ciruelos 182 Col. Tres Marias CP. 58254 Morelia Michoacán, México
Telefono	4431131900
Area	Oeste

Campus	Cuernavaca
Campus ID	0002
Direccion	Meleleucas No. 179 Col. Lomas de Cuernavaca Temixco, Morelos C.P. 62589
Telefono	7772300351
Area	Poniente

Campus	Cancún
Campus ID	0003
Direccion	Av. Bonampak 371, Sm-10, Mza2 Cancún, Quintana Roo. C.P. 77528
Telefono	9988814700
Area	Suroeste

Figura 3.5 Fragmentación Horizontal

Los tipos de fragmentación horizontal (mismo sitio), vertical (diferente sitio) o mixta (ambos tipos) son utilizados con el fin de dar solución sobre algunos conflictos en el manejo y protección de la información de las bases de datos, así como permitir la concurrencia entre los diferentes medios de comunicación por medio de enlaces o conexiones virtuales a través de redes y servidores con distintos manejadores de bases de datos (DBMS) en diferentes zonas geográficas.

3.4 Seguridad y herramientas con las que contará el sistema:

El mantener una base de datos segura con información consistente es importante para cualquier organización, ya que de esta información dependerán muchos procesos del negocio y también la toma de decisiones, pues se deben implementar procesos para asegurar la integridad, consistencia, seguridad y disponibilidad de la información, para evitar la duplicidad de la información para mantener la información consistente; lo cual puede conseguirse a través de las siguientes características:

- Validación de clientes a través de Login y Password.
- Métodos de control de acceso a la información a través del SMBD utilizando privilegios de usuario o creando roles y niveles de usuarios.
- Software de Encriptación y cifrado para protección de datos.
- Bases de datos separadas para transacciones e información de clientes.
- Firewalls.
- Sistemas de Respaldo y Backups.
- Antivirus para protección del equipo contra código malicioso.
- Certificados SSL para la protección de sitios web.
- Garantía y evaluación de vulnerabilidades.

Una transacción es considerada como una unidad lógica de trabajo y debe contar con las propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID) y el SMBD debe implementarlas.

Es importante mantener la seguridad en una base de datos para proteger, resguardar y mantener la integridad de la información, ya que el asegurar la una base de datos nos permite mantener un sistema eficaz y eficiente con información veraz, clara, consistente y confiable durante los procesos del negocio, sobre todo al tener un gran volumen de información almacenada durante su gestión y consulta, con el fin de evitar duplicidad en los datos e información redundante, inconsistente o incorrecta, y con ello poder generar reportes o consultar datos para una mejor toma de decisiones. Todo ello con el fin de mantener y salvaguardar la información así como evitar incoherencias en la misma, pues en caso contrario al estar adquiriendo información falsa o duplicada generaría conflictos y errores; adicional de que se puede poner en riesgo la confidencialidad e integridad de los datos e información, pues las personas brindan su confianza con base en la seguridad, políticas de privacidad, garantías y experiencia de las empresas (quienes ofrecen los servicios) considerando aspectos legales o éticos en relación con el derecho de acceso a la información, así como al realizar las transacciones o proporcionar su información personal.

De acuerdo a (Elmasri R. y., 2007) “En un sistema de base de datos multiusuario, el DBMS debe proporcionar técnicas que permitan a determinados usuarios o grupos de usuarios el acceso a partes concretas de una base de datos sin tener acceso al resto de esta última. Esto resulta muy importante cuando una base de datos integrada de gran tamaño va a ser utilizada por muchos usuarios distintos dentro de la misma organización. Por ejemplo, la información confidencial como los salarios de los empleados o sus evaluaciones de rendimiento debería permanecer como confidenciales para la mayoría de los usuarios del sistema de base de datos. Un DBMS incluye, por lo general, un subsistema de autorizaciones y seguridad en bases de datos responsable de garantizar la seguridad de partes de una base de datos frente a accesos no autorizados.

Además de considerar temas políticos y de relación con la información dependiendo la zona geográfica donde se encuentre la base de datos o se consulte la información. Por ello debemos tomar en cuenta la definición de los datos, así como de las especificaciones en el diseño para establecer las restricciones de clave y de integridad referencial, ya que los sistemas que utilizan bases de datos deben garantizar que la información se encuentre disponible, sea confiable y segura durante los resultados de las consultas, mismas que deberán ser eficientes y rápidas para generarlas en un lapso de tiempo corto.

Como se menciona en (Elmasri R. y., 2007) “Las amenazas a las bases de datos tienen como consecuencia la pérdida o la degradación de todos o de algunos de los siguientes objetivos de seguridad comúnmente aceptados: integridad, disponibilidad y confidencialidad. Para proteger las bases de datos contra estos tipos de amenazas, es habitual implementar cuatro tipos de medidas

de control: control de accesos, control de inferencias, control de flujo y cifrado.”

Por ello, las principales ventajas de las bases de datos multidimensionales es que favorecen y ayudan durante la toma de decisiones pues son óptimas para recuperación de información y son utilizadas para resolver consultas complejas, así mismo permiten desarrollar aplicaciones concretas al representar dimensiones o métricas por medio de tablas, pues estas se aplican sobre el sistema OLAP (On-Line analytical processing) también cubo multidimensional o hipervínculo los cuales se clasifican ROLAP, MOLAP y HOLAP. Sin embargo su principal desventaja es que son lentas durante el procesamiento de transacciones.

Considerando que en las bases de datos multidimensionales la principal ventaja es que son óptimas para recuperación de información y son utilizadas para resolver consultas complejas pues favorecen y ayudan durante la toma de decisiones, así como también permiten desarrollar aplicaciones concretas al representar dimensiones o métricas por medio de tablas, a pesar del tiempo de procesamiento en transacciones, por ello su uso es recomendable en escenarios como:

- La recuperación de datos.
- Optimizar el almacenamiento masivo de datos.
- Generar un análisis en un gran volumen de datos.
- Aplicaciones Data Warehouse.
- Generar copias de datos transaccionales.
- Almacenar datos históricos para procesar información.
- Como mecanismos de apoyo en decisiones importantes y complejas.

Así pues, la seguridad en una base de datos es importante para proteger, resguardar y mantener la integridad de la información sobre los recursos de usuarios (personas) o contra software no autorizado y mal intencionado, con el fin de lograr mantener un sistema eficaz y eficiente con información veraz, clara, consistente y confiable para

los procesos del negocio y toma de decisiones. Sobre todo al tener un gran volumen de información para generar consultas en un lapso de tiempo corto y garantizar que los datos y equipos se encuentren siempre disponibles y sean confiables, con ello evitar la corrupción y duplicidad en los datos, o tener información redundante, inconsistente e incorrecta, o ataques y violaciones al sistema.

Por ello es necesario implementar métodos para el control de acceso a la información a través del SMBD como otorgar ciertos privilegios de usuario (discrecional) o crear roles y niveles de usuarios (obligatorio), así como el uso de firewalls y antivirus, adicional de aplicar las herramientas de software y mecanismos para el control del flujo en la red de comunicación, prevención de ataques de software, intentos de hackeo, recuperar datos corrompidos a través de sistemas de respaldo y backups. Todo ello con el fin de mantener y salvaguardar la información así como evitar inconsistencias en la misma, pues en caso contrario se estaría adquiriendo información falsa o duplicada lo que generaría conflictos y errores, adicional de que se puede poner en riesgo la confidencialidad e integridad de los datos e información, pues las personas brindan su confianza con base en la seguridad, políticas de privacidad, garantías y experiencia de las empresas (quienes ofrecen los servicios) las cuales deben considerar aspectos legales o éticos en relación con el derecho de acceso a la información, además de considerar temas políticos y de relación con la información dependiendo la zona geográfica donde se encuentre la base de datos o se consulte la información, así como al realizar las transacciones o proporcionar su información personal, ya que los sistemas que utilizan bases de datos deben garantizar que la información se encuentre siempre disponible, sea confiable y segura.

3.5 Inversión e Infraestructura

Como parte de la planeación de proyectos de tecnologías de la información debemos considerar las metodologías, técnicas y diferentes herramientas para poder estructurar, documentar y programar los recursos disponibles, las tareas o

actividades, la duración del proyecto, factores como el tiempo y costo, así como las prioridades para poder gestionar los recursos y poder asignarlos.

A través de las herramientas y software para la planeación de proyectos es posible documentar y presentar de manera rápida y efectiva todas las actividades que deberán llevarse a cabo conforme a la planeación del proyecto, dichas tareas deberán ser claras y concisas para evitar confusiones entre los integrantes del proyecto.

El poder utilizar, el saber interpretar y presentar un diagrama EDT, la matriz de tiempos y prioridades, la matriz de recursos y costos, los diagramas de Gantt y/o el diagrama de PERT, así como el Presupuesto del proyecto, podemos obtener habilidades y conocimientos para desarrollar una planeación estratégica mediante la metodología de administración de proyectos para el área de TI, y con ello poder priorizar y distinguir lo urgente de lo importante para establecer lo que es relevante para el proyecto lo cual puede guiar para alcanzar el éxito, así como establecer métricas y límites de control adecuados a los objetivos de cada proyecto con base en el esfuerzo, el tiempo y los costos de tal manera que sea posible atender y cumplir los detalles, necesidades y expectativas de los clientes para lograr su satisfacción considerando el valor del negocio.

El costo de la implementación de la base de datos distribuida es de aproximadamente \$30,000.00 (treinta mil dólares 00/100) considerando eventos imprevistos, el cual será implementado en lapso de 2 meses incluyendo:

- DBMS para cada sitio corporativo (local o central).
- Plan de acciones para el control de concurrencia.
- Acceso por Internet y VPN.
- VLAN (1 pública y dos privadas).
- Soporte Técnico 24x7.

La implementación de la comunicación de nodos y redes de comunicación, así como demás hardware necesario para la configuración de la base de datos distribuida, será llevado a cabo por un área de consultoría y cuyo costo del proyecto será de aproximadamente \$15,000.00 USD (quince mil dólares) el cual será programado en el mismo período del proyecto.

De acuerdo a las fases del proyecto consideramos los siguientes resultados conforme a la investigación realizada.

Cronograma del Proyecto

Esta herramienta permite visualizar una secuencia de actividades de un proyecto, aquí es posible ver las fechas o periodos en los cuales se desarrolla cada una de las actividades, las cuales tienen un punto de inicio y un punto de finalización. Las actividades incluidas pueden ser críticas para el proyecto o bien pueden tener un margen de holgura determinado por aquella de mayor duración que se ejecuta en forma paralela, como se muestra en la figura 3.6.

	Name	Duration	Start	Finish	Precedence
1	Desarrollo de un sistema de apoyo a producción	60 days	8/1/15 8:00 AM	10/23/15 5:00 PM	
2	Análisis - Acuerdos con clientes - Preliminares	4 days	8/1/15 8:00 AM	8/6/15 5:00 PM	
3	Análisis - Acuerdos con clientes - Finales	3 days	8/21/15 8:00 AM	8/25/15 5:00 PM	4
4	Definición requerimientos - Reuniones	10 days	8/7/15 8:00 AM	8/20/15 5:00 PM	
5	Diseño y desarrollo - Diseño del sistema - Elaborar Diseño	8 days	8/31/15 8:00 AM	9/9/15 5:00 PM	4
6	Diseño y desarrollo - Diseño del sistema - Validar Diseño	3 days	9/10/15 8:00 AM	9/14/15 5:00 PM	5
7	Elaboración de módulos del sistema - Crear módulos sistema	15 days	9/15/15 8:00 AM	10/5/15 5:00 PM	6
8	Elaboración de módulos del sistema - Pruebas de los módulos del sistema	5 days	10/6/15 8:00 AM	10/12/15 5:00 PM	7
9	Elaboración de módulos del sistema - Pruebas integración del sistema	5 days	10/13/15 8:00 ...	10/19/15 5:00 PM	8
10	Postdesarrollo - Proponer mejoras	5 days	10/6/15 8:00 AM	10/12/15 5:00 PM	7
11	Postdesarrollo - Actualizar sistema con mejoras	3 days	10/13/15 8:00 ...	10/15/15 5:00 PM	10

Figura 3.6 Cronograma de Proyecto utilizando ProjectLibre Open PROJ

Diagrama de Gantt

Al utilizar el diagrama de Gantt podemos proporcionar un panorama claro y amplio sobre la programación de las actividades y tareas durante el proyecto el cual puede ser transmitido y distribuido entre los integrantes del proyecto. Para poder estructurar el Diagrama de Gantt es necesario definir las actividades, su duración, la fecha inicial y predecesora, en caso de depender de alguna otra tarea, así como los recursos y costos por unidad de tiempo, como se representa en la figura 3.7.

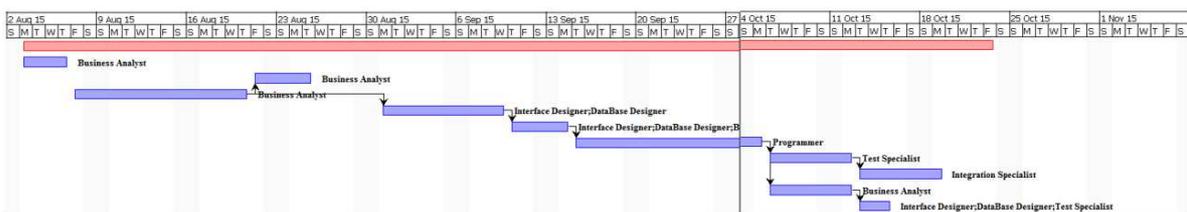


Figura 3.7 Diagrama de Gantt utilizando ProjectLibre Open PROJ

Al utilizar los diagramas de Gantt debemos tener cuidado en ingresar correctamente la información del proyecto y las relaciones de dependencia para no tener diferencias en el tiempo o generar un presupuesto mayor o menor, así como en la asignación de recursos.

Diagrama de PERT

El diagrama de PERT o de RED es utilizado como una técnica de evaluación y revisión de los programas, a través de las actividades o tareas definidas en el proyecto, en el cual se identifican las fechas de inicio y fin para poder establecer la ruta crítica del plan con el objetivo de prevenir riesgos que afecten el plan del proyecto tanto en costos como en tiempo, como se muestra en la figura 3.8.

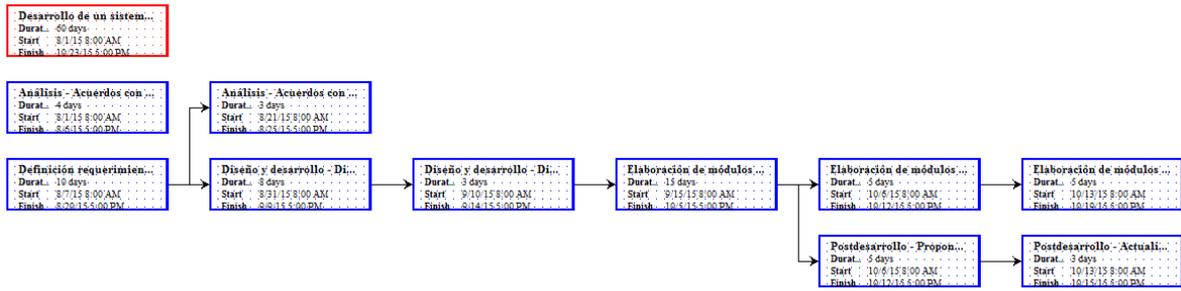


Figura 3.8 Diagrama de Pert utilizando ProjectLibre Open PROJ

Con el diagrama de PERT podemos enfocarnos en la evaluación de los riesgos que pueden afectar los costos y programación de las actividades, así como de la ruta crítica para poder tomar acciones que conlleven al éxito del proyecto.

Presupuesto del Proyecto

Con la definición del presupuesto podemos establecer un margen de inversión, costos y gastos para poder establecer el impacto financiero del proyecto y poder determinar el margen de utilidad o pérdida cualquiera que sea el caso, con el fin de poder tomar la decisión más conveniente para el negocio.

Durante las actividades del proyecto se asignan varios recursos dependiendo de sus funciones y responsabilidades, con los cuales se determina la base para el cálculo del presupuesto de acuerdo a la matriz de recursos y costos que fue proporcionada, como se muestra en la figura 3.9.

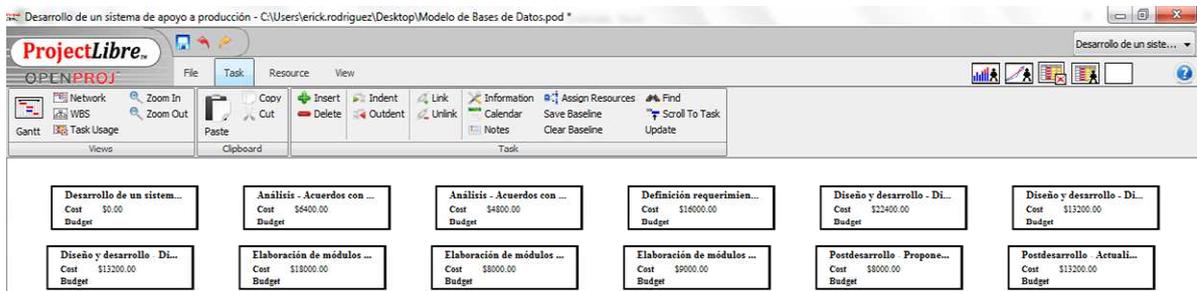


Figura 3.9 Presupuesto del Proyecto utilizando ProjectLibre Open PROJ

No obstante después de sacar el presupuesto, es recomendable calcular las reservas económicas como prevención de riesgos y contemplar contingencias o posibles eventos para poder cubrir imprevistos.

Así, con base en el análisis realizado sobre el modelo de base de datos, diseño, planeación, estrategias, recursos, tecnología, herramientas y software, que serán gestionados durante el ciclo de vida del proyecto, así como del sustento y estudios de acuerdo a las actividades del proyecto, Diagrama de Gantt, el Diagrama de PERT y la generación del Presupuesto del proyecto podemos determinar que la iniciativa es viable para su realización e implementación acorde a las fechas establecidas, tomando en consideración a la ruta crítica del proyecto.

Capítulo 4 CONCLUSIONES Y TRABAJO FUTURO

En la presente Monografía fueron aplicados los conocimientos adquiridos del curso de Modelos de Bases de Datos de la maestría en Gestión de Tecnologías de la Información del Tecmilenio Campus Morelia con la finalidad de poner en práctica todo el proceso que conlleva el elaborar una base de datos distribuida considerando aspectos de económicos, de seguridad y tecnológicos, así como las competencias adquiridas para poder proponer el diseño e implementación de una base datos distribuida haciendo uso de MySQL Workbench 5.0 CE, como iniciativa del DBMS y herramienta con la cual se llevó a cabo el diseño, desarrollo e implementación de la base de datos enfocada al manejo y gestión de la información que utiliza el personal de una Universidad en áreas administrativas, funcionales y operativas, con el fin de mejorar la coordinación, sincronización, integración, consistencia y organización de la información para su procesamiento.

Como conclusión considero que al implementar bases de datos distribuidas es esencial cuidar el diseño, su desarrollo, pruebas e implementación, así como contar con la tecnología adecuada para cada base de datos, pues el desarrollar una buena base de datos, tener la estructura y diseño adecuados, y mantener las relaciones de las tablas y conceptos, nos permitirá tener una ventaja competitiva por sobre cualquier otra organización o empresa, por lo que debemos considerar la flexibilidad, escalabilidad y velocidad con la que a futuro deberá funcionar para poder manejar grandes cantidades de datos. Adicionalmente es vital cuidar la calidad de la información tomando en cuenta la definición de los datos, así como de las especificaciones en el diseño para establecer las restricciones de clave y de integridad referencial, pues los sistemas que utilizan bases de datos deben garantizar que la información se encuentre disponible las 24hrs los 365 días, sea confiable y segura durante los resultados de las consultas, mismas que deberán ser eficientes y rápidas para generarlas en un lapso de tiempo corto.

Las bases de datos distribuidas nos permiten enlazarnos entre diferentes bases de datos que son operadas y gestionadas desde distintas zonas geográficas, con el fin de compartir información entre la colección de las bases de datos las cuales pueden ser independientes a la réplica de los datos y pueden operar desde diferentes redes de comunicaciones y dispositivos así como utilizar diferente hardware y software para las aplicaciones, sin embargo es necesario mantener los propios nodos para cuidar y resguardar la autonomía de cada uno para que permitan el acceso a la información fragmentada de los datos distribuidos debido a que cada sitio posee su propio SMBD y utilizan recursos locales para garantizar el performance y evitar el fallo del sistema si se llega a caer algún nodo y mantener la disponibilidad del mismo. Por ello es necesario poder almacenar, recuperar y resguardar la información de los sitios independientes a través de un sistema integral e incremental de respaldos de información con el fin de gestionar y controlar la concurrencia y recuperación de la información.

Respecto a la seguridad en una base de datos considero que es importante para proteger, resguardar y mantener la integridad de la información sobre los recursos de usuarios o contra software no autorizado y mal intencionado, con el fin de lograr mantener un sistema eficaz y eficiente con información veraz, clara, consistente y confiable para los procesos del negocio y toma de decisiones a través de métodos de control de acceso a la información desde el SMBD aplicando privilegios de usuario o creando roles y niveles de usuarios, así utilizar firewalls y antivirus, adicional de aplicar las herramientas de software y mecanismos para el control del flujo en la red de comunicación para prevención de ataques de software o recuperar datos corrompidos a través de sistemas de respaldo y backups. Todo ello con el fin de salvaguardar la información y evitar inconsistencias en la misma, ya que se puede poner en riesgo la confidencialidad e integridad de los datos e información, y los sistemas que utilizan bases de datos deben garantizar que la información se encuentre siempre disponible, sea confiable y segura.

Como la información es poder, el llevar a cabo su gestión será siempre un elemento clave para el éxito de las organizaciones y de las personas, ya que el manejar información nos permite consultar, analizar e interpretar datos lo que se convierte en una parte fundamental para generar nuevas estrategias, proyectos, establecer objetivos, tomar decisiones, o determinar necesidades de los usuarios y clientes quienes hacen uso del sistema.

BIBLIOGRAFÍA

- [1] Corporation, O. (13 de Junio de 2015). MySQL Workbench. Obtenido de mysql.com: <https://www.mysql.com/products/workbench/>
- [2] Elmasri, R. y. (1996). Sistemas de Bases de Datos Conceptos Fundamentales. México, DF.: Addison-Wesley Iberoamericana.
- [3] Elmasri, R. y. (2007). Fundamentos de Sistemas de Bases de Datos (5a ed.). Madrid, España: Pearson – Addison Wesley.
- [4] Hansen, G. W. (1997). Diseño y Administración de Bases de Datos. Madrid, España: Prentice Hall.
- [5] IBM. (14 de Diciembre de 2012). Dominios de datos para definiciones de reglas de calidad de datos predefinidas. Obtenido de IBM Knowledge Center: http://www-01.ibm.com/support/knowledgecenter/SSZJPZ_9.1.0/com.ibm.swg.im.iis.ia.quality.doc/topics/pdr_purpose.html?lang=es
- [6] Koch, K. L. (2002). Oracle 9i The Complete Reference. Berkeley, California: McGraw-Hill.
- [7] Lapuente, M.J. (08 de Diciembre de 2013). SGBD y STRID. Obtenido de Hipertexto: El nuevo concepto de documento en la cultura de la imagen: <http://www.hipertexto.info/documentos/sghbd.htm>
- [8] Laudon, K. y. (2012). Sistemas de información gerencial (12ª ed.). México: Pearson Educación.
- [9] McGrawHill. (13 de Junio de 2015). Sistemas de Gestores de Base de Datos. Obtenido de mcgraw-hill.es: <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>
- [10] Microsoft. (01 de Julio de 2015). Microsoft Azure. Obtenido de azure.microsoft.com: <http://azure.microsoft.com/en-us/services/sql-database/>
- [11] Microsoft. (22 de Junio de 2015). Características y tareas de Data Quality Services. Obtenido de Developer Network: <https://msdn.microsoft.com/es-es/library/hh213042.aspx>
- [12] Morales, J. D. (30 de Junio de 2014). Características y Tipos de Bases de Datos. Obtenido de IBM developerWorks: http://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/index.html
- [13] Oracle. (22 de Junio de 2015). Warehouse Builder Data Modeling, ETL, and Data Quality Guide. Obtenido de Oracle Help Center: http://docs.oracle.com/cd/E11882_01/owb.112/e10935/data_profiling.htm#WBETL18000
- [14] Silberschatz, A., F.Korth, H., & Sudarshan, S. (2006). Fundamentos de bases de datos (5ª ed.). Madrid: McGraw Hill.
- [15] Tecmilenio. (13 de Junio de 2015). Modelos de Bases de Datos. Obtenido de Blackboard: http://bbsistema.tecmilenio.edu.mx/webapps/portal/frameset.jsp?tab_ta

b_group_id=_2_1&url=%2Fwebapps%2Fblackboard%2Fexecute%2Flauncher%3Ftype%3DCourse%26id%3D_212243_1%26url%3D

- [16] Ullman, J. D., & Widom, J. (1999). Introducción a los sistemas de bases de datos. México.

ÍNDICE DE FIGURAS

Figura 2.1 MySQL WorkBench 5.2 CE con el cual se llevó a cabo el diseño y desarrollo de la DB.....	3
Figura 2.2 Modelo Físico Cliente/Servidor de la Base de Datos Distribuida	6
Figura 2.3 Modelo de Datos Relacional	7
Figura 2.4 Esquema Físico de la Base de Datos Distribuida	8
Figura 2.5 Análisis del diseño y desarrollo de la DB	9
Figura 2.6 Diseño y estructura de la tabla USUARIOS	9
Figura 2.7 Sentencias SELECT sobre la tabla USUARIOS desde MySQL Worckbench	10
Figura 2.8 Relación de las tablas Alumnos, Prestamo_Libro y Libro necesarias para Biblioteca.....	11
Figura 2.9 Diferencia entre un procedimiento almacenado y una función.....	13
Figura 3.1 Modelo Relacional, obtenido de (Corporation, 2015) solo con fines informativos.....	15
Figura 3.2 Normalización, imagen obtenida de (Elmasri R. y., 2007) solo con fines informativos.....	16
Figura 3.3 Normalización de la base de datos para no duplicar información	16
Figura 3.4 Fragmentación Vertical	20
Figura 3.5 Fragmentación Horizontal	21
Figura 3.6 Cronograma de Proyecto utilizando ProjectLibre Open PROJ.....	27
Figura 3.7 Diagrama de Gantt utilizando ProjectLibre Open PROJ.....	28
Figura 3.8 Diagrama de Pert utilizando ProjectLibre Open PROJ.....	29
Figura 3.9 Presupuesto del Proyecto utilizando ProjectLibre Open PROJ.....	29
Figura 4.1 Procedimiento almacenado para obtener una dirección Email de un Usuario.....	42
Figura 4.2 Triggers de Operaciones INSERT, DELETE y UPDATE.....	43

GLOSARIO DE TÉRMINOS

Almacenamiento principal o primario: Incluye los medios de almacenamiento en los que el CPU (unidad central de procesamiento) puede operar, como la memoria principal del computador y las memorias caché (más pequeñas, pero más rápidas). El almacenamiento principal normalmente proporciona un acceso rápido a los datos, pero tiene una capacidad de almacenamiento limitada. (Tecnología, 2015)

Almacenamiento secundario y terciario: Esta categoría incluye los discos magnéticos, los discos ópticos y las cintas. Las unidades de disco duro se clasifican como almacenamiento secundario, mientras que los medios removibles o extraíbles están considerados como almacenamiento terciario. (Tecnología, 2015)

Base de Datos Relacional: Es una colección de relaciones, en donde cada relación asemeja a una tabla de valores y cada fila de la tabla representa una colección de valores de datos relacionados entre sí, describiendo una entidad o vínculo entre entidades del mundo real. (Tecnología, 2015)

Base de Datos Distribuida (DDB – Distributed DataBase): Es una colección de múltiples bases de datos distribuidas interrelacionadas de forma lógica sobre una red de computadores. (Elmasri R. y., 2007)

Calidad de datos: Consiste en actualizar, completar y mejorar toda la información que contiene la base de datos, de manera que sea fiable y apta para el fin que se creó. (Tecnología, 2015)

Control de Concurrencia: El objetivo es asegurar la consistencia de la base de datos cuando hay accesos concurrentes por diversos usuarios. Debe garantizar el aislamiento de transacciones concurrentes. Las operaciones deben ser coordinadas para prevenir valores inconsistentes. (Tecnología, 2015)

Diseño de estructura de base de datos: Es aquí en donde se definen los conceptos en el modelo de datos, además de determinar las relaciones y designar los atributos clave. (Elmasri R. y., 2007)

Estructura de una base de datos: Se refiere a los tipos de datos, relaciones y restricciones, que deben mantenerse para los datos. (Elmasri R. y., 2007)

Fragmentación horizontal: Agrupa los registros de una relación. Se generan especificando un predicado que imponga una restricción a las tuplas de la relación. (Tecnología, 2015)

Fragmentación vertical: Agrupa los atributos de una relación que son utilizados de manera conjunta por las transacciones de mayor importancia. (Tecnología, 2015)

Fragmentación mixta: También se denomina fragmentación híbrida. El proceso de partición de la relación hace uso de la fragmentación horizontal y también de la fragmentación vertical. (Tecnología, 2015)

Herramientas de Diseño: Se utilizan en la fase de diseño (conceptual, lógico y físico) de los sistemas de bases de datos, además de realizar las tareas de comprobación de coherencia e integridad, tomando en cuenta las especificaciones a medida que avanza el diseño. (Tecnología, 2015)

Integridad: Consiste en tener datos correcto y completos que garanticen la calidad de los datos de la base de datos. (Tecnología, 2015)

Integridad Referencial: Protege las relaciones definidas entre las tablas, cuando se crean o se eliminan filas. Garantiza que los valores de clave sean coherentes en las distintas tablas, que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos. (Tecnología, 2015)

Modelo de datos: Es una colección de conceptos, que se pueden utilizar para describir la estructura de una base de datos. (Elmasri R. y., 2007)

Normalización: Tiene como objetivo transformar tablas complejas en tablas simples. El principal objetivo de este proceso es evitar la redundancia de los datos, mejorando el diseño del sistema y garantizando el cumplimiento de los requisitos del usuario. (Tecnología, 2015)

Respaldo y recuperación: Es el proceso de copiar los ficheros de la base de datos con el fin de poder utilizarlos cuando se presenta una pérdida o eventualidad que no permita acceder a ellos. (Tecnología, 2015)

Sistema de Administración de Bases de Datos Distribuidas (DDBMS – Distributed DataBase Management System): Es el software encargado de administrar la base de datos distribuida mientras hace la distribución transparente para el usuario. (Elmasri R. y., 2007)

Transacción: Una transacción es considerada como una unidad lógica de trabajo y debe contar con las propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID), y el SDBD debe implementarlas. (Tecnología, 2015)

Apéndice 1 Listado del programa

Lenguaje SQL:

1. Consulta que involucre dos tablas:

```
SELECT facu.nombre  
  , univ.clave_universidad  
  , univ.nombre  
  , univ.direccion  
  , univ.telefono  
FROM UNIVERSIDAD univ  
  , FACULTAD facu  
WHERE facu.clave_universidad = univ.clave_universidad ;
```

2. una consulta que involucre tres tablas.

```
SELECT facu.nombre  
  , prog.clave_programa  
  , prog.nombre  
  , prog.fecha_inicio  
  , prog.fecha_fin  
  , univ.clave_universidad  
  , univ.nombre  
  , univ.direccion  
  , univ.telefono  
FROM UNIVERSIDAD univ  
  , FACULTAD facu  
  , PROGRAMA prog  
WHERE facu.clave_facultad = prog.clave_facultad  
AND facu.clave_universidad = univ.clave_universidad ;
```

3. Procedimiento almacenado que valide que la edad proporcionada sea la permitida para un trabajador.

```
delimiter //  
  
CREATE PROCEDURE valida_edad (OUT empleados INT)  
  
  BEGIN  
    SELECT NUM_EMPLEADO  
      INTO empleados  
    FROM TRABAJADORES  
    WHERE EDAD BETWEEN 18 AND 50 ;  
  END//  
  
delimiter ;  
  
CALL valida_edad(@empleados) ;
```

Definición, sintaxis y creación de procedimientos con MySQL Workbench

```
CREATE
  [DEFINER = { user | CURRENT_USER }]
  PROCEDURE sp_name ([proc_parameter[,...]])
  [characteristic ...] routine_body

CREATE
  [DEFINER = { user | CURRENT_USER }]
  FUNCTION sp_name ([func_parameter[,...]])
  RETURNS type
  [characteristic ...] routine_body

proc_parameter:
  [ IN | OUT | INOUT ] param_name type

func_parameter:
  param_name type

type:
  Any valid MySQL data type

characteristic:
  COMMENT 'string'
  | LANGUAGE SQL
  | [NOT] DETERMINISTIC
  | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
  | SQL SECURITY { DEFINER | INVOKER }
```

```
routine_body:
  Valid SQL routine statement
```

Obtenido de: <http://dev.mysql.com/doc/refman/5.6/en/create-procedure.html>

4. Procedimiento almacenado o función que valide una dirección de correo electrónico considerando la tabla USUARIOS haciendo uso de MySQL WorkBench:

```
-- Creacion del Procedimiento para obtener el email del usuario
delimiter $
create procedure obtiene_email (in nombre_usuario varchar(45))
begin
  select Correo_Electronico
  from USUARIOS
  where Nombre_Completo = nombre_usuario;
end $

-- Llamar al Procedimiento para obtener el email del usuario
call obtiene_email ('Erick Arturo Rodriguez Valencia');
```

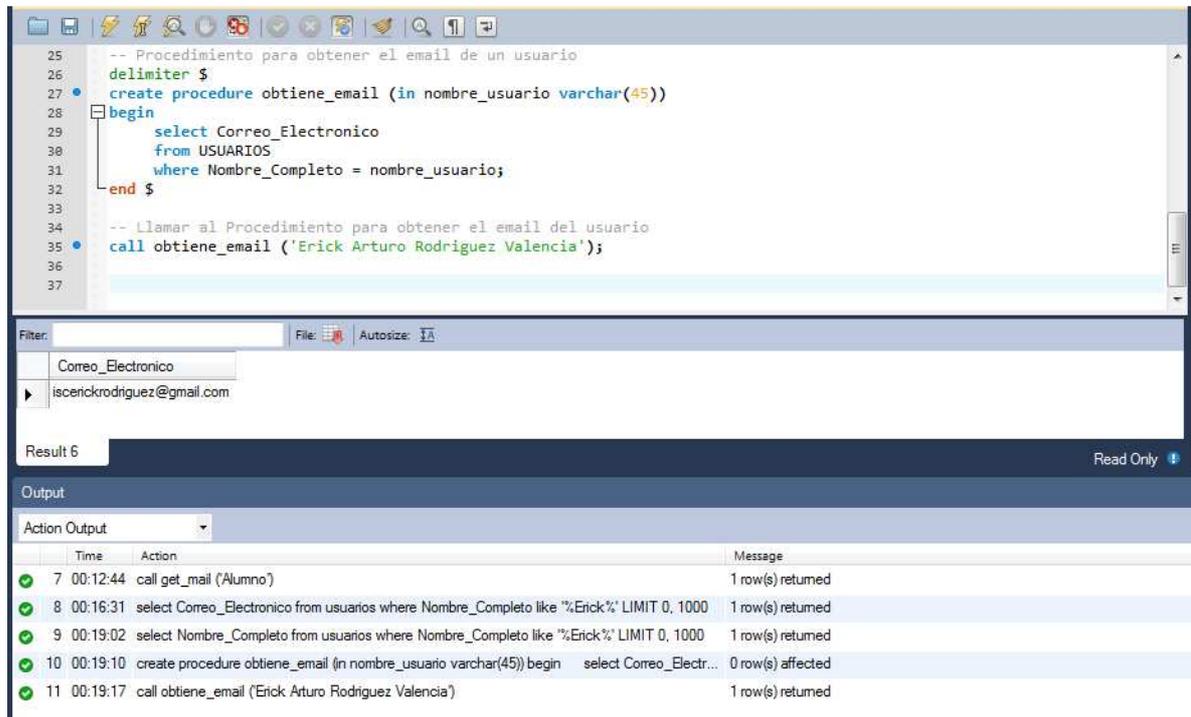


Figura 4.1 Procedimiento almacenado para obtener una dirección Email de un Usuario

5. Triggers de operaciones INSERT, DELETE y UPDATE que realizan cambios en cascada sobre tablas relacionadas, como se muestra en la figura 5.1.

Operación INSERT:

```

CREATE TRIGGER insertar AFTER INSERT ON alumnos
FOR EACH ROW
insert into usuarios(matricula, nombre_completo, fecha_nacimiento,
edad, email, telefono, descripcion)
values (new.matricula, new.nombre_completo, new.fecha_nacimiento,
new.edad, new.email, new.telefono, 'Usuario Ingresado');

insert into alumnos(matricula, nombre_completo, edad, email)
values ('AL02737412', 'Erick Rodríguez', 34,
'iscerickrodriguez@gmail.com');

select * from usuarios;
select * from alumnos;

```

Operación DELETE:

```

CREATE TRIGGER eliminar AFTER DELETE ON alumnos
FOR EACH ROW
insert into usuarios(matricula, nombre_completo, fecha_nacimiento,
edad, email, telefono, descripcion)

```

```

    values (new.matricula, new.nombre_completo, new.fecha_nacimiento,
new.edad, new.email, new.telefono, 'Usuario Eliminado');

delete from alumnos where alumnos.matricula = 'AL02737412';

select * from usuarios;
select * from alumnos;

```

Operación UPDATE:

```

CREATE TRIGGER modificar AFTER UPDATE ON alumnos
FOR EACH ROW
    insert into usuarios(matricula, nombre_completo, fecha_nacimiento,
edad, email, telefono, descripcion)
    values (new.matricula, new.nombre_completo, new.fecha_nacimiento,
new.edad, new.email, new.telefono, 'Usuario Actualizado');

update alumnos
set alumnos.email = 'al02737421@tecmilenio.mx'
where alumnos.matricula = 'AL02737412';

select * from usuarios;
select * from alumnos;

```

The screenshot shows a window titled 'Triggers' with a list of SQL statements. The code is as follows:

```

15 CREATE TRIGGER insertar AFTER INSERT ON alumnos
16 FOR EACH ROW
17   insert into usuarios(matricula, nombre_completo, fecha_nacimiento, edad, email, telefono, descripcion)
18   values (new.matricula, new.nombre_completo, new.fecha_nacimiento, new.edad, new.email, new.telefono, 'Usuario Ingresado');
19
20 insert into alumnos(matricula, nombre_completo, edad, email)
21 values ('AL02737412', 'Erick Rodríguez', 34, 'isckerickrodriguez@gmail.com');
22
23 select * from usuarios;
24
25 CREATE TRIGGER modificar AFTER UPDATE ON alumnos
26 FOR EACH ROW
27   insert into usuarios(matricula, nombre_completo, fecha_nacimiento, edad, email, telefono, descripcion)
28   values (new.matricula, new.nombre_completo, new.fecha_nacimiento, new.edad, new.email, new.telefono, 'Usuario Actualizado');
29
30 update alumnos
31 set alumnos.email = 'al02737421@tecmilenio.mx'
32 where alumnos.matricula = 'AL02737412';
33
34 select * from usuarios;
35
36 CREATE TRIGGER eliminar AFTER DELETE ON alumnos
37 FOR EACH ROW
38   insert into usuarios(matricula, nombre_completo, fecha_nacimiento, edad, email, telefono, descripcion)
39   values (new.matricula, new.nombre_completo, new.fecha_nacimiento, new.edad, new.email, new.telefono, 'Usuario Eliminado');
40
41 delete from alumnos where alumnos.matricula = 'AL02737412';
42
43 select * from usuarios;
44

```

Figura 4.2 Triggers de Operaciones INSERT, DELETE y UPDATE

6. Conexión a la Base de Datos

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

DROP SCHEMA IF EXISTS `University_DB` ;

```

```

CREATE SCHEMA IF NOT EXISTS `University_DB` ;
USE `University_DB` ;

-----
-- Table `University_DB`.`USUARIOS`
-----
DROP TABLE IF EXISTS `University_DB`.`USUARIOS` ;

CREATE TABLE IF NOT EXISTS `University_DB`.`USUARIOS` (
  `Usuario_ID` INT UNSIGNED NOT NULL AUTO_INCREMENT COMMENT 'La tabla
USUARIOS tiene la opción de seleccionar un perfil y un código para poder
filtrar información referente a Alumnos (AL) o Profesores (PRF), como
ejemplo el código será la matrícula de los alumnos.' ,
  `Perfil` VARCHAR(10) BINARY NULL ,
  `Codigo` VARCHAR(12) BINARY NULL ,
  `Primer_Nombre` VARCHAR(15) BINARY NOT NULL ,
  `Segundo_Nombre` VARCHAR(15) BINARY NULL ,
  `Apellido_Paterno` VARCHAR(15) BINARY NOT NULL ,
  `Apellido_Materno` VARCHAR(15) BINARY NULL ,
  `Nombre_Completo` VARCHAR(45) BINARY NULL ,
  `Sexo` VARCHAR(9) BINARY NULL ,
  `Fecha_Nacimiento` DATE NULL ,
  `RFC` VARCHAR(13) BINARY NULL ,
  `CURP` VARCHAR(18) BINARY NULL ,
  `Edad` INT(3) NULL ,
  `Calle` VARCHAR(45) BINARY NULL ,
  `Numero_Exterior` VARCHAR(5) BINARY NULL ,
  `Numero_Interior` VARCHAR(15) BINARY NULL ,
  `Colonia` VARCHAR(45) BINARY NULL ,
  `Municipio` VARCHAR(10) BINARY NULL ,
  `Ciudad` VARCHAR(10) BINARY NULL ,
  `Estado` VARCHAR(15) BINARY NULL ,
  `Pais` VARCHAR(15) BINARY NULL ,
  `Codigo_Postal` INT(8) NULL ,
  `Telefono_Casa` INT(10) NULL ,
  `Telefono_Celular` INT(10) NULL ,
  `Correo_Electronico` VARCHAR(45) BINARY NULL ,
  `Escaridad` VARCHAR(15) BINARY NULL ,
  PRIMARY KEY (`Usuario_ID`) )
ENGINE = InnoDB;

USE `University_DB` ;

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

```

7. Sincronización

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

ALTER TABLE `University_DB`.`USUARIOS` CHANGE COLUMN `Perfil` `Perfil`
VARCHAR(10) BINARY NULL DEFAULT NULL , CHANGE COLUMN `Codigo` `Codigo`
VARCHAR(12) BINARY NULL DEFAULT NULL , CHANGE COLUMN `Primer_Nombre`

```

```
`Primer_Nombre` VARCHAR(15) BINARY NOT NULL , CHANGE COLUMN
`Segundo_Nombre` `Segundo_Nombre` VARCHAR(15) BINARY NULL DEFAULT NULL ,
CHANGE COLUMN `Apellido_Paterno` `Apellido_Paterno` VARCHAR(15) BINARY
NOT NULL , CHANGE COLUMN `Apellido_Materno` `Apellido_Materno`
VARCHAR(15) BINARY NULL DEFAULT NULL , CHANGE COLUMN `Nombre_Completo`
`Nombre_Completo` VARCHAR(45) BINARY NULL DEFAULT NULL , CHANGE COLUMN
`Sexo` `Sexo` VARCHAR(9) BINARY NULL DEFAULT NULL , CHANGE COLUMN `RFC`
`RFC` VARCHAR(13) BINARY NULL DEFAULT NULL , CHANGE COLUMN `CURP` `CURP`
VARCHAR(18) BINARY NULL DEFAULT NULL , CHANGE COLUMN `Calle` `Calle`
VARCHAR(45) BINARY NULL DEFAULT NULL , CHANGE COLUMN `Numero_Exterior`
`Numero_Exterior` VARCHAR(5) BINARY NULL DEFAULT NULL , CHANGE COLUMN
`Numero_Interior` `Numero_Interior` VARCHAR(15) BINARY NULL DEFAULT NULL
, CHANGE COLUMN `Colonia` `Colonia` VARCHAR(45) BINARY NULL DEFAULT NULL
, CHANGE COLUMN `Municipio` `Municipio` VARCHAR(10) BINARY NULL DEFAULT
NULL , CHANGE COLUMN `Ciudad` `Ciudad` VARCHAR(10) BINARY NULL DEFAULT
NULL , CHANGE COLUMN `Estado` `Estado` VARCHAR(15) BINARY NULL DEFAULT
NULL , CHANGE COLUMN `Pais` `Pais` VARCHAR(15) BINARY NULL DEFAULT NULL
, CHANGE COLUMN `Correo_Electronico` `Correo_Electronico` VARCHAR(45)
BINARY NULL DEFAULT NULL , CHANGE COLUMN `Escolaridad` `Escolaridad`
VARCHAR(15) BINARY NULL DEFAULT NULL ;
```

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