**INSTITUTO TECNOLÓGICO DE TLALNEPANTLA**



Reporte de Residencia Profesional

**Implementación del módulo de gestión de base de datos del servicio social en el Instituto Tecnológico Gustavo A. Madero II, Fase 1.**

Instituto Tecnológico de Gustavo A. Madero II.

**Salinas Oregon Sofia. 17251084**

Carrera: Ingeniería en Tecnologías de la Información y Comunicación. Especialidad en Ingeniería en Software.

Asesor interno: Martínez Romero Julia.

Periodo de realización: Agosto-diciembre 2022.

Periodo de realización:

Semestre Mes inicio-Mes final Año

Ciudad de México, diciembre 2022

**CARTA DE PRESENTACION**

|  |
| --- |
| Incluir documento escaneado |

**CARTA DE ACEPTACION**

|  |
| --- |
| Incluir documento escaneado |

**COPIA DE ASIGNACION DE ASESOR INTERNO**

Incluir documento escaneado

**CARTA DE TERMINACIÓN**

|  |
| --- |
| Incluir documento escaneado |

**HOJA DE EVALUACION DE DESAMPEÑO Y/O ANEXO XXIX. FORMATO DE EVALUACION Y SEUMIENTO DE RESIDENCIAS PROFESIONALES**

Incluir documento escaneado

**ANEXO XXX. FORMATO DE EVALUACIÓN DE REPORTE DE RESIDENCIA PROFESIONAL**

Incluir documento escaneado

**AGRADECIMIENTOS**

# resumen

En resumen, se implementa una base de datos afiliada a la página web del SASS (servicio Automatizado del Servicio Social), donde se abordarán y apoyan procesos, temas fundamentales y tramites de importancia para la institución. Para el desarrollo de proyecto, como sistemas operativos de red, perfiles de usuario y servidor, así mismo su seguridad y correcta administración.

Esto beneficiara de gran forma al personal administrativo, en la seguridad de su archivo, una administración centralizada, integración en los servicios de Windows o aplicaciones de terceros y escalabilidad.

De igual manera, ayuda a que el tramite sea más rápido, eficiente y se pueda realizar desde la comodidad del usuario.

Con esto se beneficia al grupo estudiantil para acceder de manera de usuarios al trámite y correcta realización del mismo. De igual manera, y como solución principal, se le da la solución al proceso del trámite de servicio social, ayudando al personal del ITGAM II para que dicho proceso se hiciera de manera automatizada y no de manera manual y física como hasta antes de la solución del proyecto.

Contenido

[resumen 9](#_Toc117514926)

[Índice de Figura iii](#_Toc117514927)

[**Índice de tablas** iv](#_Toc117514928)

[Introducción 1](#_Toc117514929)

[capitulo 1. GENERALIDADES del proyecto 3](#_Toc117514930)

[1.1 Características del área de desarrollo 4](#_Toc117514931)

[1.1.1 ORGANIGRAMA GENERAL: 5](#_Toc117514932)

[1.1.2 Actividades que desarrolla la Institución. 5](#_Toc117514933)

[1.1.3 Departamento donde se realizaron residencias profesionales. 7](#_Toc117514934)

[1.2 Problemas a Resolver 8](#_Toc117514935)

[1.3 Objetivos 8](#_Toc117514936)

[1.3.1 Objetivo general 8](#_Toc117514937)

[1.3.2 Objetivos específicos 8](#_Toc117514938)

[1.4 Justificación 10](#_Toc117514939)

[1.5 Alcances y limitaciones del proyecto 11](#_Toc117514940)

[1.5.1 Delimitación espacial. 11](#_Toc117514941)

[1.5.2 Delimitación temporal. 11](#_Toc117514942)

[capitulo 2. MARCO TEÓRICO 12](#_Toc117514943)

[2.1 Ingeniería de software 13](#_Toc117514944)

[2.1.1 Definición de Software 13](#_Toc117514945)

[2.1.2 ¿Qué es ingeniería del software? 13](#_Toc117514946)

[2.1.3 Objetivos de la ingeniería de software 13](#_Toc117514947)

[2.1.4 Etapas de la Ingeniería de Software. 14](#_Toc117514948)

[2.1.5 Capas de la ingeniería de software 14](#_Toc117514949)

[2.2 Base de Datos 15](#_Toc117514950)

[2.2.1 TABLAS. 15](#_Toc117514951)

[2.2.2 Tipos de datos. 16](#_Toc117514952)

[2.2.3 Tipo de datos en SQL. 17](#_Toc117514953)

[2.2.4 Relaciones de tabla. 19](#_Toc117514954)

[2.2.5 Tipos de relaciones de tablas. 19](#_Toc117514955)

[2.2.6 KEYS. 20](#_Toc117514956)

[2.2.7 Modelamiento. 21](#_Toc117514957)

[2.2.8 Formularios. 21](#_Toc117514958)

[2.2.9 Informes. 21](#_Toc117514959)

[2.2.10Consultas. 21](#_Toc117514960)

[2.2.11 Tipo de Datos en las Bases de Datos 22](#_Toc117514961)

[2.2.12 ¿Qué es el lenguaje de consulta estructurada (SQL)? 22](#_Toc117514962)

[2.2.13 Tipos de bases de datos. 23](#_Toc117514963)

[2.2.14 Tipos de modelos de bases de datos. 25](#_Toc117514964)

[2.2.3 ¿Qué es un software de base de datos? 26](#_Toc117514965)

[2.2.4 ¿Qué es un sistema de gestión de bases de datos (DBMS)? 27](#_Toc117514966)

[2.2.5 ¿Qué es una base de datos MySQL? 27](#_Toc117514967)

[2.2.6 ¿Qué es un gestor de base de datos? 27](#_Toc117514968)

[2.2.7 Diccionario de datos 28](#_Toc117514969)

[2.2.8 Administrador de la base de datos 28](#_Toc117514970)

[2.2.9 Lenguajes de sistema gestor de BD 28](#_Toc117514971)

[2.2.10 Enfoque metodológico de una base de datos 29](#_Toc117514972)

[2.2.11 Visión de los datos 29](#_Toc117514973)

[2.2.12 Modelos de los datos 30](#_Toc117514974)

[capitulo 3. DESARROLLO 33](#_Toc117514975)

[PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS 34](#_Toc117514976)

[2. CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA 39](#_Toc117514977)

[3. Competencias DESARROLLADAS Y/O APLICADAS EN EL PROYECTO 40](#_Toc117514978)

[Referencias (FUENTES DE INFORMACIÓN) 41](#_Toc117514979)

[**Anexo A. lista de estilos disponibles** 42](#_Toc117514980)

[**Viñetas** 42](#_Toc117514981)

[**Figuras** 42](#_Toc117514982)

[**Tablas** 42](#_Toc117514983)

[**Ecuaciones** 43](#_Toc117514984)

# Índice de Figura

Ilustración 1 Vista satelital del Instituto Tecnológico Gustavo A. Madero 4

Ilustración 0‑2 Organigrama general del Instituto Tecnológico de Gustavo A. Madero II. 5

Ilustración 0‑3Organigrama Centro de Computo del Instituto Tecnológico Gustavo A. Madero 2. (TECNM campus Gustavo A. Madero ll, 2022 7

0‑1 3.,1. Capacpq 14

Ilustración 0‑2 Estructura básica de una base de datos 15

Ilustración 0‑1Tipo de llaves en base de datos 20

Ilustración 0‑2 ¿Que permite hacer el SQL con la base de datos? 23

**Índice de tablas**

Tabla 1 TABLA TIPO DE RELACIONES DE TABLAS 12

Tabla 2 TIPOS DE DATOS EN BD 13

Tabla 3 2.2 3 TABLA DE SINONIMOS DE LOS TIPOS DE DATOS DEFINIDOS 14

Tabla 4TABLA TIPO DE RELACIONES DE TABLAS 16

Tabla 5TABLA TIPO DE RELACIONES DE TABLAS 16

# Introducción

La implementación del módulo de gestión de base de datos del servicio social en el Instituto Tecnológico Gustavo A. Madero II (ITGAM II), Fase 1, una vez en funcionamiento, el personal administrativo que se encuentra a cargo de realizar las actividades de registro del servicio social, se verá beneficiada, ahorrando así, tiempo y esfuerzo, debido a que en la actualidad cada uno de los registro de los alumnos se realiza de manera manual (archivos físicos), sin contar que el ITGAM II cuenta con 150 alumnos registrados al servicio social, y se esperan la misma cantidad o más de registros para el siguiente periodo de registro.

Diseñar e implementar una base de datos conectada a una interfaz web, para gestionar y agilizar el Sistema de Automatización de Servicio Social (SASS), en el departamento de Gestión Tecnológica y Vinculación en el Instituto Tecnológico Gustavo A. Madero II, es el objetivo general. Dividiendo la carga de trabajo en los siguientes objetivos específicos. Diseñar e implementar una base de datos que aloja la información necesaria para el proceso de registro y liberación del servicio social. Construir procedimientos almacenados que al ser ejecutados enviará una petición de usuario, ejecutándose directamente del motor de base de datos que corre en un servidor separado. Normalizar la base de datos, para que se encuentre dentro de los estándares de las bases de datos. Hacer uso de un diagrama entidad- relación para tener una vista general para la base de datos, y la cual ayudará en la escalabilidad del proyecto. Desarrollar el entorno grafico de los formularios que permitan la inserción de los datos dentro de la base de datos, esto mediante el uso de tecnologías de parte del cliente HTML, CSS y su framework Bootstrap 5, JavaScript y Figma. Y adaptar las vistas graficas necesarias para la gestión de información en la base de datos. Cubrirá un periodo de 4 meses, de agosto a diciembre del año 2022. Este proyecto está contemplado por la institución ITGAM II (Instituto Tecnológico Gustavo A. Madero II) para ser un proyecto que contemple cuatro diferentes fases. Fase1: Servicio Social. Fase 2: Residencias Profesionales. Fase 3: Registro de dependencias. Fase 4: Integración total de fases en un mismo sistema. El proyecto por petición de la organización se ha decidido implementar la fase 1 únicamente por lo que, no cuenta con registro de dependencias, ya que es tema de fases posteriores, no incluidas dentro de esta fase.

.

# capitulo 1. GENERALIDADES del proyecto

# 1.1 Características del área de desarrollo

**Nombre**: Instituto Tecnológico Gustavo A. Madero II.

**Giro:** Educativo.

**Capital**: Educativo

**Dirección** Calle Morelos No. 193, Col: Loma La Palma, Alcaldía Gustavo A. Madero, México CDMX, C.P. 07160.

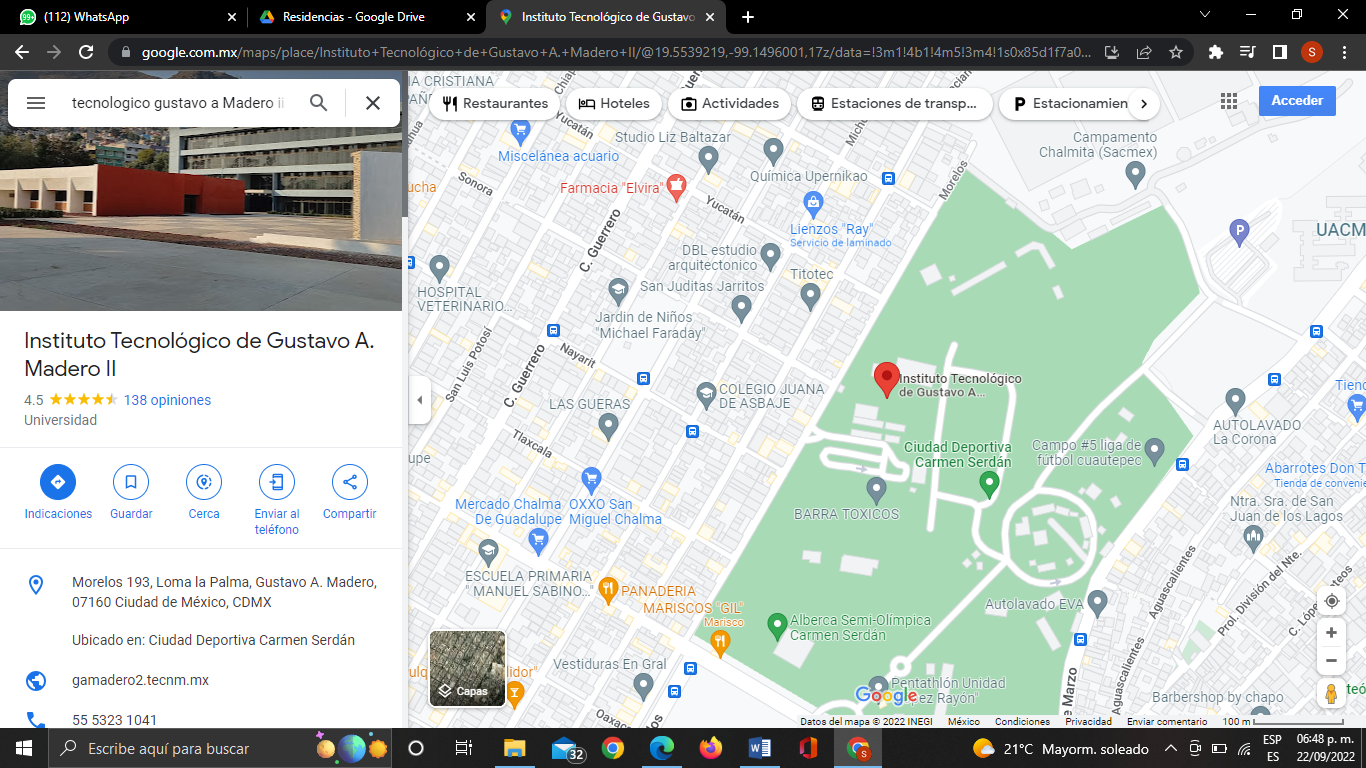


Ilustración 1 Vista satelital del Instituto Tecnológico Gustavo A. Madero

Google. (s.f.). [Mapa de Google Maps del Instituto Tecnológico Gustavo A. Madero II]. Recuperado el 20 de septiembre, 2022, de <https://goo.gl/maps/uC9zpQrAAio2U3G99>

.

### 1.1.1 ORGANIGRAMA GENERAL:

El Instituto Tecnológico Gustavo A. Madero II se encuentra integrado con una dirección, cuatro subdirecciones y diez y nueve departamentos internos. (Figura 2).



Ilustración 0‑2 Organigrama general del Instituto Tecnológico de Gustavo A. Madero II.

### 1.1.2 Actividades que desarrolla la Institución.

**Subdirección de Planeación y Vinculación.**

**Servicios Escolares.**

1. Determinar necesidades de servicios a estudiantes y de control escolar en el Instituto Tecnológico Gustavo A. Madero II.
2. Integrar la información y documentación requerida para el registro y certificación del alumno.
3. Coordinar y controlar los tramites de acreditación, certificación, convalidación, revisión, revalidación, equivalencia y regularización de estudios que se generen en el Instituto Tecnológico Gustavo A. Madero II.
4. Organizar y controlar la atención médica a estudiantes, así como la orientación psicológica profesional cuando esta se requiera, de igual forma, el proceso de inscripción al régimen facultativo del I.M.S.S.
5. Recibir el expediente del egresado para gestionar ante las instancias correspondientes su título, cedula profesional o grado que corresponda.

**Actividad Extraescolar.**

1. Analizar, detectar y elaborar la relación de recursos necesarios requeridos (humanos, financieros, materiales y servicios) para el logro de objetivos y metas del programa Operativo Anual de la oficina a su cargo y someterla a la consideración del jefe de Departamento de Actividades Extraescolares.
2. Promover y coordinar los eventos de tipo deportivos, culturales y cívicos del alumnado del Instituto.
3. Controlar y evaluar la instrucción deportiva, cultural y cívica del alumnado del Instituto.
4. Gestionar el proceso para lograr el traslado de grupos deportivos, grupos culturales y grupos cívicos entre el Instituto Tecnológico y otras instituciones, áreas o lugares correspondientes en la región.
5. Vigilar y controlar que el equipo y las instalaciones a su cargo se mantengan en buen estado físico y en óptimo funcionamiento.

**Gestión Tecnológica y Vinculación.**

Genera Vinculación con el sector productivo generando acuerdos de colaboración con las empresas privadas, dependencias de gobierno u otras organizaciones para la Planeación, Organización, Operación, Control y Cumplimiento de:

* Servicio Social.
* Residencias Profesionales.
* Visita a Empresas.
* Cursos
* Diplomados.
* Bolsa de trabajo.
* Proyectos Tecnológicos.
* Egresados.

### 1.1.3 Departamento donde se realizaron residencias profesionales.

**Centro de computo**

Organigrama del departamento. (Figura 3).

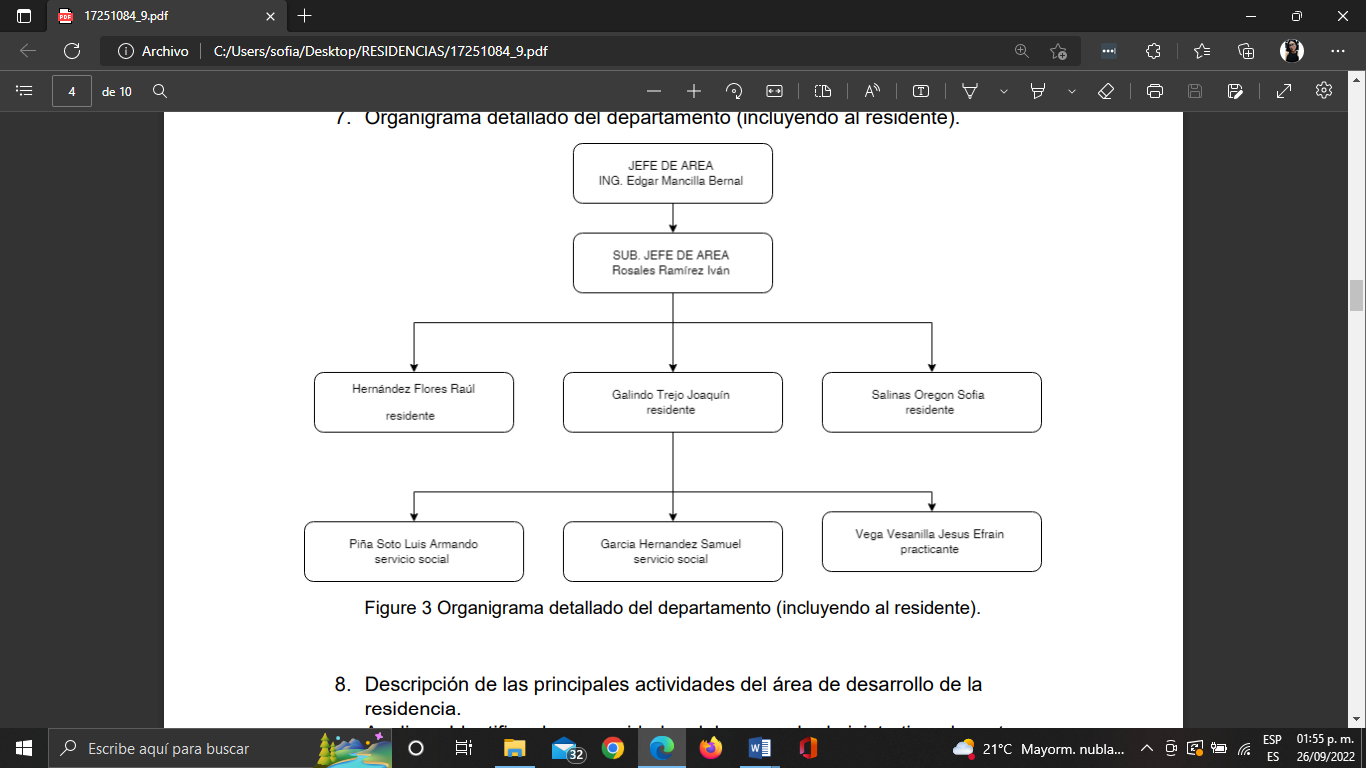


Ilustración 0‑3Organigrama Centro de Computo del Instituto Tecnológico Gustavo A. Madero 2. (TECNM campus Gustavo A. Madero ll, 2022

1. **Analizar**: Identificar las necesidades del personal administrativo, docentes y alumnos para brindar el apoyo en relación a las tecnologías de la información y comunicación, dando todas las herramientas para que puedan realizar sus tareas diarias dentro del plantel.

2. **Gestionar y organizar:** El site del Instituto Tecnológico de Gustavo A. Madero II es sinónimo de trabajo en equipo, así que gestionar y organizar a cada integrante otorga la confianza para realizar las actividades inherentes en tiempo y forma.

3. **Planear**: Es fundamental prevenir inconvenientes relacionados con los equipos de cómputo y las redes informáticas, así que una parte fundamental del equipo es planificar mantenimientos preventivos constantes que no intervengan con las actividades del personal, generando un lugar de trabajo eficiente y confiable para todos los usuarios.

4. **Reparar:** Existen problemas que se pueden presentar de manera inesperada y es fundamental para los colaboradores del instituto estar atentos a ellas para solucionar cualquier eventualidad de manera casi inmediata para así mantener las actividades de la comunidad del plantel inalteradas.

### 1.2 Problemas a Resolver

El proyecto se implementará en el Departamento de Gestión Tecnológica y Vinculación en el Instituto Tecnológico Gustavo A. Madero II; para gestionar y agilizar el sistema de servicio social (SASS), dado que el departamento se encuentra actualmente de manera presencial con todos los tramites que dicho registro demanda haciéndose de manera manual y física, quitando tiempo, haciendo carga de trabajo y alargando el proceso.

Por dichos sucesos, se necesita digitalizar el proceso de sistema de servicio social, logrando agilizar y gestionar el proceso del sistema.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Diseñar e implementar una base de datos conectada a una interfaz web, para gestionar y agilizar el sistema de servicio social (SASS), en el departamento de Gestión Tecnológica y Vinculación en el Instituto Tecnológico Gustavo A. Madero II.

### 1.3.2 Objetivos específicos

* Diseñar e implementar la base de datos que aloja la información necesaria para el proceso del sistema de servicio social (SASS).
* Realizar diagrama de flujo con la finalizad de agilizar el proceso de desarrollo para la interfaz web.
* Hacer uso de un diagrama entidad- relación para tener una vista general para la base de datos.
* Construir procedimientos almacenados que al ser ejecutados enviara una petición de usuario ejecutándose directamente del motor de base de datos que corre en un servidor separado.
* Adaptar las vistas graficas necesarias para la gestión de información en la base de datos.
* Desarrollar el entorno grafico de los formularios que permitan la inserción de los datos en la base de datos, esto mediante el uso de HTML.
* Generar el registro para el servicio social (SASS) por medio del sistema.

# 1.4 Justificación

El proyecto se implementará en el Departamento de Gestión Tecnológica y Vinculación en el Instituto Tecnológico Gustavo A. Madero II; para gestionar y agilizar el sistema de servicio social (SASS), dado que el departamento se encuentra actualmente de manera presencial con todos los tramites que dicho registro demanda haciéndose de manera manual y física, quitando tiempo, haciendo carga de trabajo y alargando el proceso.

Por dichos sucesos, se necesita digitalizar el proceso de sistema de servicio social, logrando agilizar y gestionar el proceso del sistema.

También, en la actualidad los datos de las empresas siempre serán prioridad, donde necesitan proporcionar varios servicios a un gran número de usuarios ya que no existe una seguridad ni un control, debido a que se necesita dar una instalación y configuración inicial para así verificar de forma segura la identidad de sus respectivos usuarios. Así como las empresas ya que deben tener un controlador de dominio para gestionar de forma óptima la información que se maneja y el acceso a sus usuarios que se concedan.

Es por esta razón se implementa un módulo de gestión base de datos en el servicio social en el ITGAM II ya que tendrá beneficios como la seguridad con la gestión de políticas y permisos dentro de los equipos, la administración centralizada desde el mismo servidor podrá tener un control a los equipos conectados al dominio incluyendo los objetos que son los usuarios, grupos de trabajo y Unidades organizativas de este modo ayuda a dar un servicio rápido eficiente a los usuarios donde podrán realizar tareas en tiempo y forma, teniendo así un 90 % en la administración de los usuarios del tecnológico.

Muy importante el tema de escalabilidad ya que, con la implementación de un dominio en el tecnológico, se podrá en un futuro llegar a implementarse en un árbol o bosque de dominio así administrándolo a gran escala obteniendo un 80 %.

## 1.5 Alcances y limitaciones del proyecto

### 1.5.1 Delimitación espacial.

Este proyecto está contemplado por la institución ITGAM II (Instituto Tecnológico Gustavo A. Madero II) para ser un proyecto que contemple cuatro diferentes fases. Fase1: Servicio Social. Fase 2: Residencias Profesionales. Fase 3: Registro de dependencias. Fase 4: Integración total de fases en un mismo sistema. El proyecto por petición de la organización para que las siguientes fases sean parte de posteriores proyectos de residencias profesionales, se ha decidido implementar la fase 1 únicamente por lo que, no cuenta con registro de dependencias, ya que es tema de fases posteriores, no incluidas dentro de esta.

### 1.5.2 Delimitación temporal.

Cubrirá un periodo de 4 meses, de agosto a diciembre del año 2022.

.

# capitulo 2. MARCO TEÓRICO

## 2.1 Ingeniería de software

### 2.1.1 Definición de Software

* 1. El software es Instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados.
  2. estructura de datos que permiten que los programas manipulen en forma adecuada la información

### 2.1.2 ¿Qué es ingeniería del software?

La ingeniería del software es una disciplina que implica el uso de estructuras, herramientas y técnicas para construir programas informáticos.

Así mismo, incluye el análisis previo de la situación, la redacción del proyecto, la creación del software y las pruebas necesarias para garantizar el correcto funcionamiento del software antes de poner el sistema en funcionamiento.

Esta ingeniería aborda todas las fases del ciclo de vida de desarrollo de cualquier tipo de sistema de información y es aplicable a una amplia gama de ámbitos de la informática y la ciencia de los ordenadores, como el diseño de compiladores, sistemas operativos y tecnologías de Intranet/Internet.

### 2.1.3 Objetivos de la ingeniería de software

Los objetivos de la ingeniería de software son muy diversos, pero podemos destacar los siguientes más importantes:

* Crear programas informáticos que satisfagan las necesidades de la sociedad y empresas.
* Guiar y coordinar el desarrollo de una programación difícil.
* Intervenir en el ciclo de vida de un producto.
* Estimar los costos y el plazo de ejecución de un proyecto.
* Actuar como líder del equipo de desarrollo de software.
* Diseño, desarrollo y administración de bases de datos.
* Durante la creación de la aplicación, liderar y dirigir a los programadores.
* Incluir procesos de calidad en las aplicaciones, como la medición de métricas y medidas y la evaluación de la calidad del software.

### 2.1.4 Etapas de la Ingeniería de Software.

Dentro de la ingeniería de software entendemos que también se encuentra todo el proceso de elaboración del software, que se denomina ciclo de vida. Está formado por cuatro etapas:

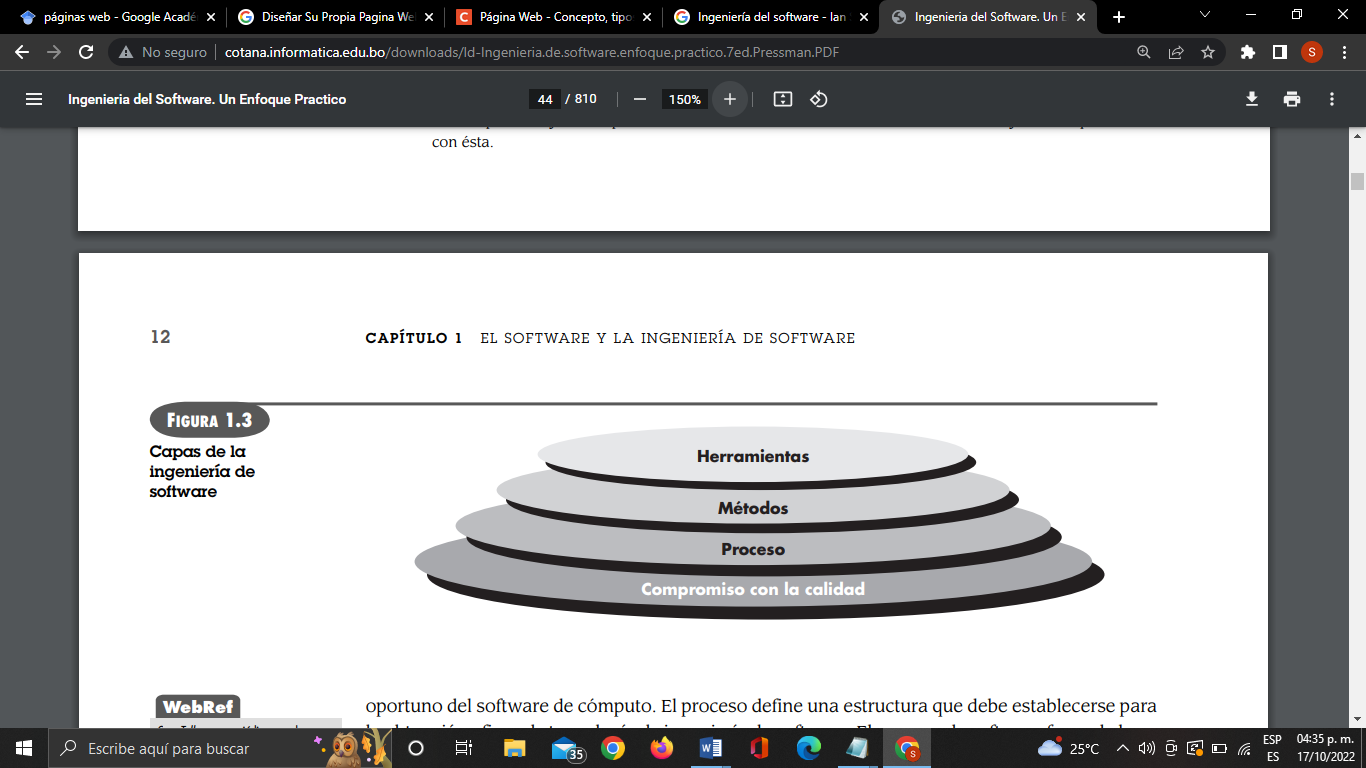
* Concepción. En esta primera fase se desarrolla el modelo de negocio. Es decir, conocemos las necesidades que debe de tener un software y empezamos a buscar las herramientas para cubrirlas.
* Elaboración. Se detalla las características de la estructura del software.
* Construcción. Tal y como su nombre indica en este paso empezamos a elaborar de forma tangible todo aquello que, de momento, solo hemos plasmado en forma de ideas.
* Transición. Es el momento de la implementación y el desarrollo para los clientes o usuarios. Deben tener tiempo para familiarizarse con el nuevo software.

### 2.1.5 Capas de la ingeniería de software

La ingeniería de software es una tecnología con varias capas, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares10 alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.

El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de

software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.



0‑1 3.,1. Capacpq

## 2.2 Base de Datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un [sistema de gestión de bases de datos (DBMS)](https://www.oracle.com/mx/database/what-is-database/#WhatIsDBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.



Ilustración 0‑2 Estructura básica de una base de datos

## 2.2.1 TABLAS.

Una tabla de base de datos es similar en apariencia a una hoja de cálculo en cuanto a que los datos se almacenan en filas y columnas. Por ende, es bastante fácil importar una hoja de cálculo en una tabla de base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y almacenarlos en una base de datos es la forma en la que están organizados los datos.

Para aprovechar al máximo la flexibilidad de una base de datos, los datos deben organizarse en tablas para que no se produzcan redundancias.

Cada fila de una tabla se denomina registro. En los registros se almacena información. Cada registro está formado por uno o varios campos. Los campos equivalen a las columnas de la tabla.

Una tabla tiene registros (filas) y campos (columnas). Los campos tienen diferentes tipos de datos, como texto, números, fechas e hipervínculos.

* Un registro: contiene datos específicos, como información acerca de un determinado empleado o un producto.
* Un campo: contiene datos sobre un aspecto del asunto de la tabla, como el nombre o la dirección de correo electrónico.
* Un valor de campo: cada registro tiene un valor de campo.

Una variable es un objeto del lenguaje cuyo valor se puede cambiar. Antes de utilizar una variable ésta debe de ser declarada. Al declarar una variable, se le asocia un identificador, es decir, un nombre, con un tipo de almacenamiento cuya forma determina la visibilidad y existencia de la variable.

El tipo de la variable nos indica el conjunto de valores que puede tomar y las operaciones que pueden realizarse con ella.

## 2.2.2 Tipos de datos.

Existen cinco tipos de datos básicos:

|  |  |  |
| --- | --- | --- |
| Tipos de Datos | Descripción | Memoria |
| int | Tipo entero | 2 bytes |
| Char | Carácter | 1 byte |
| Float | Almacena valores reales en punto flotante. | 4 bytes |
| Doublé | Almacena valores reales en doble precisión. | 8 bytes. |
| void | Se utiliza para definir una función que no devuelve ningún valor. |  |

Tabla 1 TABLA TIPO DE RELACIONES DE TABLAS

## 2.2.3 Tipo de datos en SQL.

Los tipos de datos SQL se clasifican en 13 tipos de datos primarios y de varios sinónimos válidos reconocidos por dichos tipos de datos. Los tipos de datos primarios son:

|  |  |  |
| --- | --- | --- |
| TIPO DE DATOS | LONGITUD | DESCRIPCION |
| Binary | 1 byte | Para consultas sobre tabla adjunta de productos de BD que define un tipo de dato Binario. |
| BIT | 1 byte | Valores Si/No o True/False |
| BYTE | 1 byte | Un valor entero entre 0 y 255. |
| COUNTER | 4 bytes | Un numero incrementado automáticamente de tipo (LONG) |
| CURRENCY | 8 bytes | Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807. |
| DATETIME | 8 bytes | Un valor de fecha u hora entre los años 100 y 9999. |
| SINGLE | 4 bytes | Un valor en punto flotante de precisión simple con un rango de - 3.402823\*1038 a -1.401298\*10-45 para valores negativos, 1.401298\*10- 45 a 3.402823\*1038 para valores positivos, y 0. |
| DOUBLE | 8 bytes | Un valor en punto flotante de doble precisión con un rango de - 1.79769313486232\*10308 a -4.94065645841247\*10-324 para valores negativos, 4.94065645841247\*10-324 a 1.79769313486232\*10308 para valores positivos, y 0 |
| SHORT | 2 bytes | Un entero corto entre -32,768 y 32,767. |
| LONG | 4 bytes | Un entero largo entre -2,174,483,648 y 2,147,483,647. |
| LONGTEXT | 1 byte por carácter | De cero a un máximo de 1.2 gigabytes. |
| LONGBINARY | Según se necesite | De cero 1 gigabytes. Utilizado para objetos OLE. |
| TEXT | 1 byte por carácter | De cero a 255 caracteres. |

Tabla 2 TIPOS DE DATOS EN BD

La siguiente tabla recoge los sinónimos de los tipos de datos definidos:

|  |  |
| --- | --- |
| Tipo de Dato | Sinónimos |
| BINARY | VARBINARY |
| BIT | BOOLEAN  LOGICAL  LOGICAL 1  YESNO |
| BYTE | INTEGER 1 |
| COUNTER | AUTOINCREMENT |
| CURRENCY | MONEY |
| DATETIME | DATE  TIME  TIMESTAMP |
| SINGLE | FLOAT 4  IEEESINGLE  REAL |
| DOUBLE | FLOAT  FLOAT 8  IEEEDOUBLE  NUMBER  NUMERIC |
| SHORT | INTEGER 2  SMALLINT |
| LONG | INT  INTEGER  INTEGER 4 |
| LONGBINARY | GENERAL  OLEOBJECT |
| LONGTEXT | LONGCHAR  MEMO  NOTE |
| TEXT | ALPHANUMERIC  CHAR- CHARACTER  STRING- VARCHAR |
| VARIANT (NO ADMITIDO) | VALUE |

Tabla 3 TABLA DE SINONIMOS DE LOS TIPOS DE DATOS DEFINIDOS

## 2.2.4 Relaciones de tabla.

Aunque en cada tabla se almacenan datos sobre un tema distinto, en las tablas de una base de datos de Access suelen almacenarse datos sobre temas que están relacionados entre sí. Por ejemplo, puede que una base de datos contenga:

* Una tabla de clientes con los clientes de su empresa y sus direcciones.
* Una tabla de productos con los productos a la venta, incluidos los precios e imágenes de cada artículo.
* Una tabla de pedidos en la que se realice un seguimiento de los pedidos de los clientes.

Dado que almacena datos sobre diferentes temas en tablas independientes, necesita alguna manera de vincular los datos para que pueda combinar fácilmente datos relacionados de esas tablas independientes. Para conectar los datos almacenados en tablas diferentes, cree relaciones. Una relación es una conexión lógica entre dos tablas que especifica los campos que las tablas tienen en común.

Para ello, se colocan campos comunes en las tablas que estén relacionadas y se definen las relaciones entre las tablas.

### 2.2.5 Tipos de relaciones de tablas.

Existen tres tipos de relaciones de tablas:

|  |  |
| --- | --- |
| Tipo de relación de tabla. | Descripción. |
| UNO A UNO | Cuando cada elemento de cada tabla solo aparece una vez. Por ejemplo, cada empleado puede tener solo un coche de la empresa a su disposición. |
| UNO A VARIOS | Cuando un elemento en una tabla puede tener una relación con varios elementos de otra tabla. Por ejemplo, cada pedido de compra puede incluir varios productos. |
| VARIOS A VARIOS | Cuando uno o más elementos en una tabla puede tener una relación con uno o más elementos de otra tabla. Por ejemplo, cada pedido puede tener varios productos, y cada producto puede aparecer en varios pedidos. |

Tabla 4 TABLA TIPO DE RELACIONES DE TABLAS

### 2.2.6 KEYS.

Los campos que forman parte de una relación de tabla se denominan keys (llaves). Una clave consta por lo general en un campo, pero puede estar compuesta por más de un campo. Existen dos tipos de keys:

|  |  |
| --- | --- |
| TIPO DE KEY`S | Descripción. |
| KEY PRINCIPAL | Una tabla puede tener solamente una key principal. Una key principal está compuesta de uno o más campos que identifican cada registro almacenado en la tabla de forma exclusiva. A menudo, hay un número de identificación único, como un número de identificador, un número de serie o un código, que sirve de key principal. |
| KEY FORANEA | Una tabla puede tener una o más keys foráneas. Este tipo de keys contiene valores que se corresponden con los valores de la clave principal de otra tabla. Por ejemplo, puede tener una tabla Pedidos en la que cada pedido tenga un número de identificación de cliente que se corresponda con un registro de una tabla Clientes. El campo de identificador de cliente es una keys foránea de la tabla Pedidos. |

Tabla 5 TABLA TIPO DE RELACIONES DE TABLAS

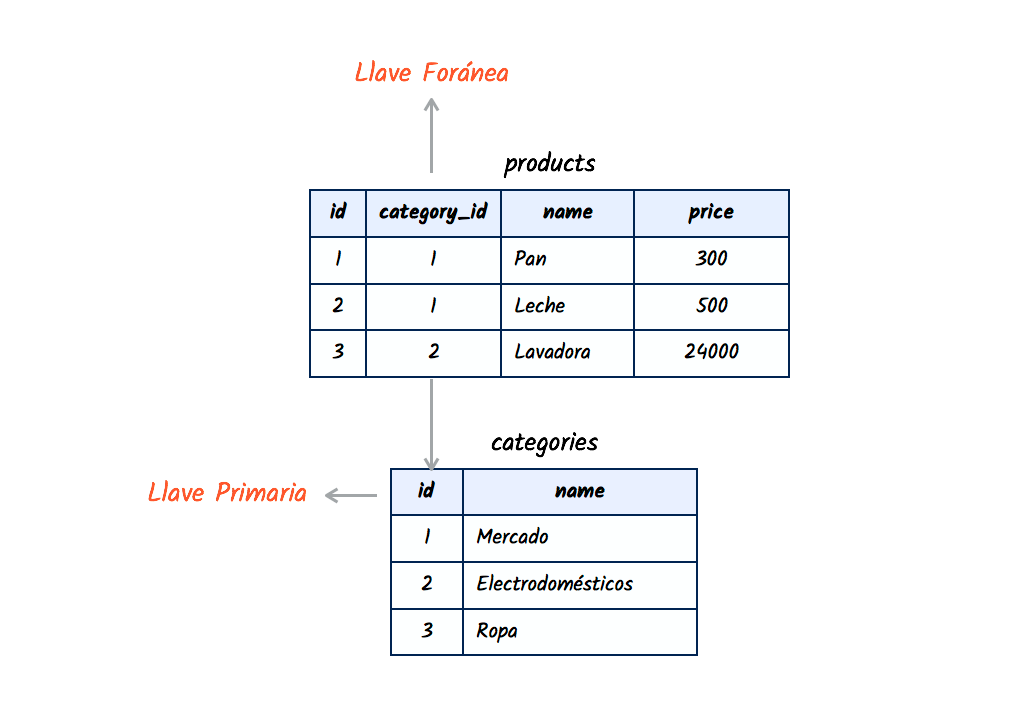


Ilustración 0‑1Tipo de llaves en base de datos

## 2.2.7 Modelamiento.

Al proceso de diseñar el diagrama con la estructura de la base de datos (tablas, columnas y relaciones) se le conoce como modelamiento de bases de datos y es uno de los primeros pasos en el proceso de desarrollo de software.

En un diagrama de bases de datos la forma de representar una tabla es utilizando un rectángulo que muestra el nombre de la tabla y sus columnas. Para mayor detalle se puede incluir el tipo de columna y sus características (tamaño, restricciones, etc.):

### 2.2.8 Formularios.

Los formularios permiten crear una interfaz de usuario en la que puede escribir y modificar datos. Los formularios a menudo contienen botones de comandos y otros controles que realizan distintas tareas. Puede crear una base de datos sin usar formularios con tan solo modificar los datos en las hojas de datos de la tabla. Sin embargo, la mayoría de los usuarios de bases de datos prefieren usar formularios para ver, escribir y modificar datos en las tablas.

Puede programar botones de comandos para determinar qué datos aparecen en el formulario, abrir otros formularios o informes, o ejecutar otras tareas. Por ejemplo, puede tener un formulario llamado "Formulario de cliente" en el que trabaja con los datos de los clientes. El formulario de cliente puede tener un botón que abra un formulario de pedido en el que puede especificar un pedido nuevo para el cliente.

Los formularios también le permiten controlar de qué manera otros usuarios interactúan con los datos de la base de datos. Por ejemplo, puede crear un formulario que muestre solo determinados campos y permita que se realicen únicamente ciertas operaciones. Esto ayuda a proteger los datos y a asegurarse de que los datos se especifican correctamente.

### 2.2.9 Informes.

Los informes se usan para dar formato a los datos, resumirlos y presentarlos. Por lo general, un informe responde a una pregunta específica como: "¿Cuánto dinero recibimos de cada cliente este año?" o "¿En qué ciudades residen nuestros clientes?". A cada informe se le puede dar formato para presentar la información de la manera más legible posible.

Se puede ejecutar un informe por vez y siempre se reflejan los datos actuales de la base de datos. Generalmente, se les da formato a los informes para imprimirlos, pero también pueden verse en pantalla, exportarse a otro programa o enviarse como datos adjuntos en un correo electrónico

### 2.2.10Consultas.

Las consultas pueden realizar diversas funciones en una base de datos. La función más común es recuperar datos específicos de las tablas. Los datos que quiere ver generalmente están distribuidos en varias tablas y las consultas le permiten verlos en una única hoja de datos. Además, debido a que muchas veces no quiere ver todos los registros a la vez, las consultas le permiten agregar criterios para "filtrar" los datos y obtener solo los registros que quiere.

Ciertas consultas son "actualizables", es decir, puede modificar los datos de las tablas subyacentes mediante la hoja de datos de la consulta. Si está trabajando en una consulta actualizable, recuerde que los cambios se realizan en realidad en las tablas, no solo en la hoja de datos de la consulta.

Hay dos variedades básicas de consultas:

|  |  |
| --- | --- |
| Tipo de consultas | Descripción |
| Consultas de selección | Recupera los datos y los pone a disposición para su uso. |
| Consultas de acciones. | Realiza una tarea con los datos. Se pueden usar para crear tablas nuevas, agregar datos a las tablas existentes o actualizar o eliminar datos. |

Tabla 6 TABLA DE TIPO DE CONSULTAS

## 2.2.11 Tipo de Datos en las Bases de Datos

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

### 2.2.12 ¿Qué es el lenguaje de consulta estructurada (SQL)?

El SQL es un lenguaje de programación que utilizan casi todas las bases de datos relacionales para consultar, manipular y definir los datos, además de para proporcionar control de acceso.

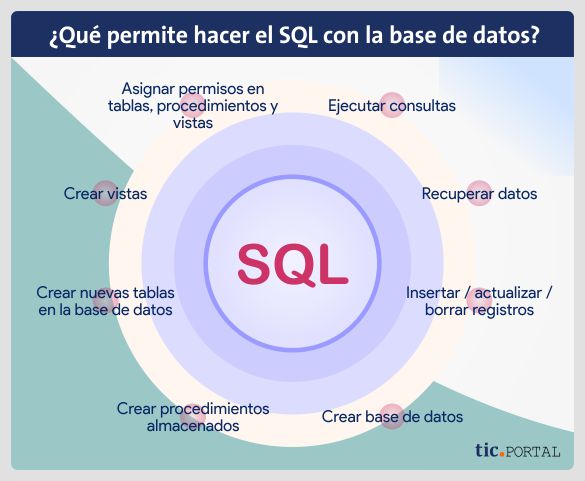


Ilustración 0‑2 ¿Que permite hacer el SQL con la base de datos?

### 2.2.13 Tipos de bases de datos.

|  |  |
| --- | --- |
| Bases de datos relacionales | * [Las bases de datos relacionales](https://www.oracle.com/mx/database/what-is-a-relational-database/) se hicieron predominantes en la década de 1980. Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada. |
| Bases de datos orientadas a objetos | * La información de una base de datos orientada a objetos se representa en forma de objetos, como en la programación orientada a objetos. |
| Bases de datos distribuidas | Una base de datos distribuidas consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes. |
| Almacenes de datos | Un repositorio central de datos, un data warehouse es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos. |
| Bases de datos NoSQL | Una base de datos [NoSQL](https://www.oracle.com/mx/database/nosql-cloud.html), o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas |
| Bases de datos orientadas a grafos. | Una base de datos orientada a grafos almacena datos relacionados con entidades y las relaciones entre entidades. |
| Base de datos OLTP. | Una base de datos OLTP es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones. |
| Bases de datos en la nube. | Una base de datos en la nube es una recopilación de datos, estructurados o no estructurados, que reside en una plataforma de cloud computing privada, pública o híbrida. Existen dos tipos de modelos de bases de datos en la nube: el modelo tradicional y el de base de datos como servicios (database as a service, DBaaS). Con DBaaS, un proveedor de servicios realiza las tareas administrativas y el mantenimiento. |
| Base de datos multimodelo. | Las bases de datos multimodelo combinan distintos tipos de modelos de bases de datos en un único servidor integrado. Esto significa que pueden incorporar diferentes tipos de datos. |
| Bases de datos de autogestión. | El tipo de base de datos más nuevo e innovador, las bases de datos de autogestión (también conocidas como bases de datos autónomas) están basadas en la nube y utilizan el machine learning para automatizar el ajuste de la base de datos, la seguridad, las copias de seguridad, las actualizaciones y otras tareas de gestión rutinarias que tradicionalmente realizan los administradores de bases de datos. |

Tabla 7Tipos de Bases de Datos

### 2.2.14 Tipos de modelos de bases de datos.

#### 2.2.14.1 MODELO DE BD PLANA.

En un modelo de base de datos plano, hay dos dimensiones (estructura plana) de conjunto de datos. Hay una columna de información y dentro de esta columna, se supone que cada dato tendrá que ver con la columna.

Por ejemplo, un modelo de base de datos plana que sólo incluye códigos postales. Dentro de la base de datos, sólo habrá una columna y cada nueva fila dentro de una columna será un nuevo código postal.

Todos los registros se almacenan en un solo lugar. La estructura es simple.

Funciona bien para bases de datos pequeñas y tiene requisitos mínimos de software y hardware.

#### 2.2.14.2 EL MODELO DE BASE DE DATOS JERÁRQUICA.

El modelo jerárquico de bases de datos se asemeja a la estructura de un árbol, tal como Microsoft Windows organiza las carpetas y archivos. En un modelo jerárquico de bases de datos, cada enlace es anidado con el fin de conservar los datos organizados en un orden particular en un mismo nivel de lista.

#### 2.2.14.3 EL MODELO DE BASE DE DATOS DE RED

En un modelo de red, la característica definitoria es que se almacena un registro con un enlace a otros registros - en efecto, una red.

Estas redes (o, a veces, a que se refiere como punteros) puede ser una variedad de diferentes tipos de información como números de nodo de un disco o incluso la dirección.

Al igual que una base de datos jerárquica, los componentes de una base de datos de red pueden tener varias relaciones padre-hijo. Su capacidad para mantener datos es más que la del modelo jerárquico.

Dado que siempre hay un vínculo entre un padre y un hijo, se mantiene la integridad de los datos.

La base de datos debe dedicar menos tiempo a mantener registros debido a la eliminación de tablas redundantes, lo que mejora la eficiencia y el rendimiento general.

#### 2.2.14.4 El modelo relacional

El modelo relacional es el más popular tipo de base de datos y una herramienta extremadamente potente, no sólo para almacenar información, también para acceder a ella.

Las bases de datos relacionales son organizadas en forma de tablas. La belleza de estos cuadros es que la información se puede acceder o añadir sin reorganizar las tablas.

Una tabla puede tener muchos registros y cada registro puede tener muchos campos.

Se evita la duplicación de datos, manteniendo así el tamaño de la base de datos pequeño. Esto reduce el requerimiento de memoria y también aumenta la eficiencia.

### 2.2.3 ¿Qué es un software de base de datos?

El software de base de datos se utiliza para crear, editar y mantener archivos y registros de bases de datos, lo que facilita la creación de archivos y registros, la entrada de datos, la edición de datos, la actualización y la creación de informes. El software también maneja el almacenamiento de datos, las copias de seguridad y la creación de informes, así como el control de acceso múltiple y la seguridad. La seguridad sólida de las bases de datos es especialmente importante hoy en día, ya que el robo de datos es cada vez más frecuente. En ocasiones, el software de base de datos también se denomina "sistema de gestión de bases de datos" (DBMS).

El software de base de datos simplifica la gestión de datos, ya que permite a los usuarios almacenar datos de forma estructurada y acceder posteriormente a ellos. Por lo general, tiene una interfaz gráfica que ayuda a crear y administrar los datos y, en algunos casos, los usuarios pueden construir sus propias bases de datos mediante el uso de software de base de datos.

### 2.2.4 ¿Qué es un sistema de gestión de bases de datos (DBMS)?

Una base de datos requiere un programa de software de bases de datos completo, conocido como sistema de gestión de bases de datos (DBMS). Un DBMS sirve como interfaz entre la base de datos y sus programas o usuarios finales, lo que permite a los usuarios recuperar, actualizar y gestionar cómo se organiza y se optimiza la información. Un DBMS también facilita la supervisión y el control de las bases de datos, lo que permite una variedad de operaciones administrativas como la supervisión del rendimiento, el ajuste, la copia de seguridad y la recuperación.

Algunos ejemplos de software de bases de datos o DBMS populares incluyen MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database y dBASE.

### 2.2.5 ¿Qué es una base de datos MySQL?

MySQL es un sistema de gestión de datos relacionales de código abierto basado en SQL. Se diseñó y se optimizó para las aplicaciones web y puede utilizarse en cualquier plataforma. A medida que surgían nuevos y diferentes requisitos con Internet, MySQL se convirtió en la plataforma preferida por los desarrolladores web y las aplicaciones basadas en web. Dado que está diseñado para procesar millones de consultas y miles de transacciones, MySQL es una elección popular para las empresas de comercio electrónico que necesitan gestionar múltiples transferencias de dinero. La flexibilidad on-demand es la principal función de MySQL.

### 2.2.6 ¿Qué es un gestor de base de datos?

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

* Definición de los datos
* Mantenimiento de la integridad de los datos dentro de la base de datos
* Control de la seguridad y privacidad de los datos
* Manipulación de los datos

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales.  
Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor.  
  
El gestor almacena una descripción de datos en lo que llamamos diccionario de datos, así como los usuarios permitidos y los permisos.  
Tiene que haber un usuario administrador encargado de centralizar todas estas tareas

### 2.2.7 Diccionario de datos

Es una base de datos donde se guardan todas las propiedades de la base de datos, descripción de la estructura, relaciones entre los datos, etc.

El diccionario debe contener:

* La descripción externa, conceptual e interna de la base de datos
* Las restricciones sobre los datos
* El acceso a los datos
* Las descripciones de las cuentas de usuario
* Los permisos de los usuarios
* Los esquemas externos de cada programa

### 2.2.8 Administrador de la base de datos

Es una persona o grupo de personas responsables del control del sistema gestor de base de datos.

Las principales tareas de un administrador son:

* La definición del esquema lógico y físico de la base de datos
* La definición de las vistas de usuario
* La asignación y edición de permisos para los usuarios
* Mantenimiento y seguimiento de la seguridad en la base de datos
* Mantenimiento general del sistema gestor de base de datos.

### 2.2.9 Lenguajes de sistema gestor de BD

Un sistema gestor de base de datos debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos. Estos lenguajes son los siguientes:

* Lenguaje de definición de datos (DDL). Para definir los esquemas de la base de datos
* Lenguaje de manipulación de datos (DML). Para manipular los datos de la base de datos
* Lenguaje de control de datos(DCL). Para la administración de usuarios y seguridad en la base de datos.

### 2.2.10 Enfoque metodológico de una base de datos

La modelación de bases de datos es un proceso que parte del análisis y precisión de los requisitos que deben cumplir los RI para ser insertados en el repositorio de manera que sirvan a los propósitos definidos para el contexto de uso de la base: usuarios, funcionalidades del software y restricciones. La modelación constituye una guía para la implementación y posterior mantenimiento incorporando mejoras en el contenido y en el desempeño operacional de la base. El proceso de modelación abarca desde el nivel conceptual -independiente del software y de la implementación hasta el nivel de realización mediante un sistema y los RI que se manejan. En cada nivel se recurre a diversos tipos de herramientas para construir los modelos. Dichas herramientas también denominadas "modelos".

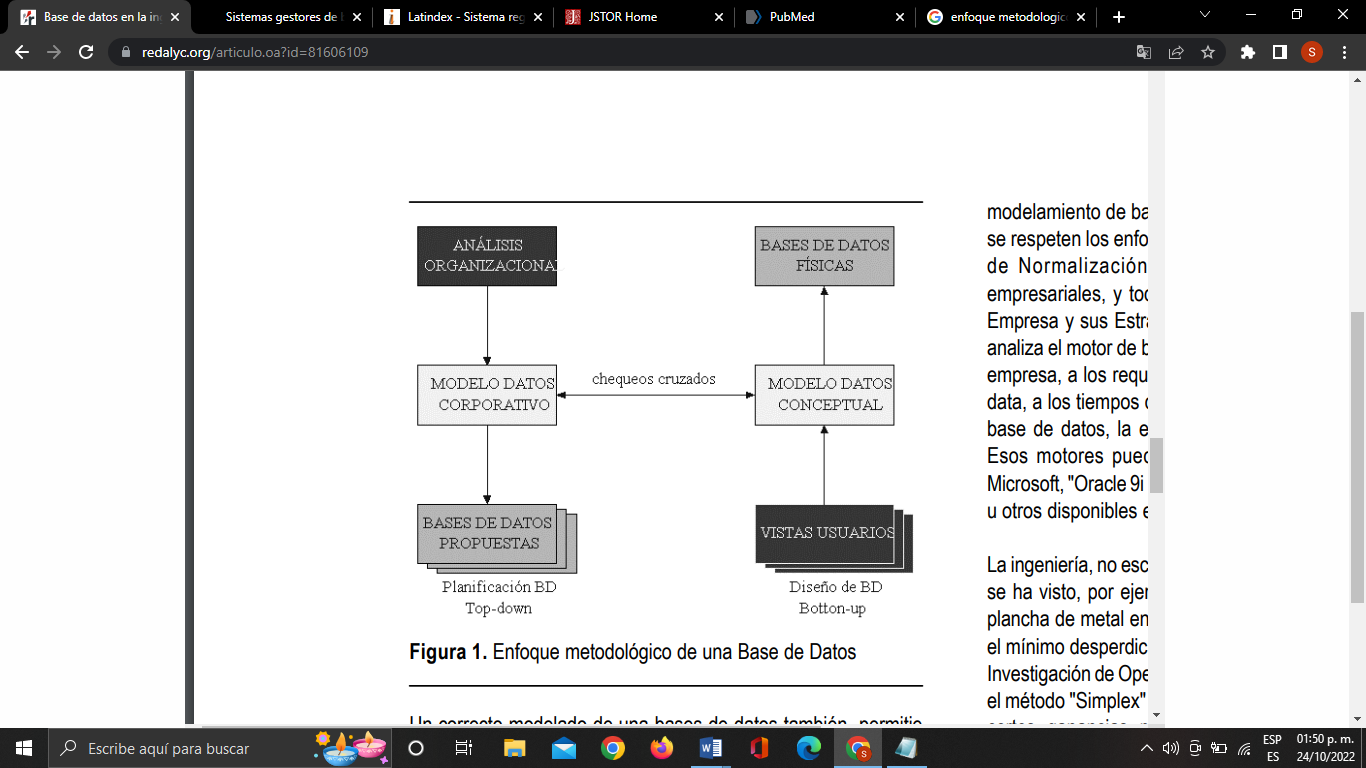


Ilustración 2 1 Enfoque metodológico de una base de datos

### 2.2.11 Visión de los datos

Un sistema de base de datos es una colección de archivos interrelacionados y un conjunto de programas que permitan a los usuarios a acceder y modificar estos archivos. Uno de los propósitos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

#### 2.2.11.1 Abstracción de datos

para que el sistema sea útil debe recuperar los datos eficientemente. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos. Los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema.

* Nivel físico: El nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.
* Nivel lógico: El siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
* Nivel de vistas: El nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios del sistema de base de datos no necesitan toda esta información. En su lugar, tales usuarios necesitan acceder sólo a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos

### 2.2.12 Modelos de los datos

Bajo la estructura de la base de datos se encuentra el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Para ilustrar el concepto de un modelo de datos, describimos dos modelos de datos en este apartado: el modelo entidad-relación y el modelo relacional. Los diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos.

#### 2.2.12.1 Modelo entidad-relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de atributos.

Una relación es una asociación entre varias entidades. Por ejemplo, una relación impositor asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R, que consta de los siguientes componentes:

* Rectángulos, que representan conjuntos de entidades.
* Elipses, que representan atributos.
* Rombos, que representan relaciones entre conjuntos de entidades.
* Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Además de entidades y relaciones, el modelo E-R representa ciertas restricciones que los contenidos de la base de datos deben cumplir. Una restricción importante es la correspondencia de cardinalidades, que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones. Por ejemplo, si cada cuenta puede pertenecer sólo a un cliente, el modelo puede expresar esta restricción.

#### 2.2.12.2 Modelo relacional

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

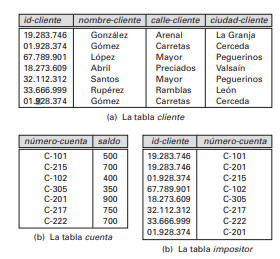


Ilustración 2 2 Ejemplo de base de datos relacional

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

#### 2.2.12.3 Otros modelos de datos

El modelo de datos orientado a objetos es otro modelo de datos que está recibiendo una atención creciente.

El modelo orientado a objetos se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos (funciones) e identidad de objetos.

El modelo de datos orientado a objetos, está basado en el paradigma de los lenguajes de programación orientados a objetos, que en este momento tienen un gran uso. La herencia, la identidad de objetos, y el encapsulamiento (información oculta), con métodos para proporcionar una interfaz a los objetos, están entre los conceptos clave de la programación orientada a objetos que han encontrado aplicaciones en los modelos de datos. Los modelos de datos orientados a objetos también soportan un rico sistema de tipos, incluyendo tipos colección y estructurados. Mientras la herencia y, hasta cierto punto, los tipos complejos están también presentes en el modelo E-R, el encapsulamiento y la identidad de objetos distinguen el modelo de datos orientado a objetos frente al modelo E-R.

##### 2.2.12.3.1 estructura de los objetos

Hablando en general, los objetos se corresponden con las entidades del modelo E-R. El paradigma orientado a objetos está basado en el encapsulamiento de los datos y del código relacionados con cada objeto en una sola unidad cuyo contenido no es visible desde el exterior. Conceptualmente, todas las interacciones entre cada objeto y el resto del sistema se realizan mediante mensajes. Por tanto, la interfaz entre cada objeto y el resto del sistema se define mediante un conjunto de mensajes permitidos.

En general, cada objeto está asociado con:

* Un conjunto de variables que contiene los datos del objeto; las variables se corresponden con los atributos del modelo E-R.
* Un conjunto de mensajes a los que responde; cada mensaje puede no tener parámetros, tener uno o varios.
* Un conjunto de métodos, cada uno de los cuales es código que implementa un mensaje; el método devuelve un valor como respuesta al mensaje.

El término mensaje en un entorno orientado a objetos no implica el uso de mensajes físicos en redes informáticas. Por el contrario, hace referencia al intercambio de solicitudes entre los objetos independientemente de los detalles concretos de su implementación. Se utiliza a veces la expresión invocar a un método para denotar el hecho de enviar un mensaje a un objeto y la ejecución del método correspondiente.

Los métodos de cada objeto pueden clasificarse como sólo de lectura o de actualización. Los métodos sólo de lectura no afectan al valor de las variables de los objetos, mientras que los métodos de actualización sí pueden modificarlo. Los mensajes a los que responde cada objeto pueden clasificarse de manera parecida como sólo de lectura o de actualización, según el método que los implemente.

Los atributos derivados de las entidades del modelo E-R pueden expresarse en el modelo orientado a objetos como mensajes sólo de lectura.

En el modelo orientado a objetos hay que expresar cada atributo de las entidades como una variable y un par de mensajes del objeto correspondiente. La variable se utiliza para guardar el valor del atributo, uno de los mensajes se utiliza para leer el valor del atributo y el otro mensaje se utiliza para actualizar ese valor.

##### 2.2.12.3.2 Clases de objetos

Generalmente, en una base de datos hay muchos objetos similares. Por similar se entiende que responden a los mismos mensajes, utilizan los mismos métodos y tienen variables del mismo nombre y del mismo tipo. Sería un derroche definir por separado cada uno de estos objetos. Por tanto, los objetos parecidos se agrupan para formar una clase.

Todos los objetos de una clase comparten una definición común, pese a que se diferencien en los valores asignados a las variables.

El concepto de clase del modelo orientado a objetos se corresponde con el concepto de entidad del modelo E-R.

El concepto de clases es parecido al concepto de los tipos abstractos de datos. Sin embargo, hay varios aspectos adicionales en el concepto de clase respecto al de tipos abstractos de datos. Para representar estas propiedades adicionales, cada clase se trata como si fuera un objeto.

Un objeto clase incluye:

* + Una variable de tipo conjunto cuyo valor es el conjunto de todos los objetos que son ejemplares de la clase.
  + La implementación de un método para el mensaje nuevo, que crea un nuevo ejemplar de la clase.

##### 2.2.12.3.3 Herencia

Los esquemas de las bases de datos orientadas a objetos suelen necesitar gran número de clases. Frecuentemente, sin embargo, varia de las clases son parecidas entre sí.

Herencia en los sistemas orientados a objetos es el concepto de posibilidad de sustitución: cualquier método de una clase dada.

La **herencia en base de datos orientada a objetos** hace referencia a que los objetos heredan comportamientos dentro de una jerarquía de clases, es decir, una clase se deriva de otra de manera que extiende su funcionalidad.

**Herencia múltiple.**

En la mayor parte de los casos una organización de clases con estructura de árbol resulta adecuada para describir las aplicaciones; en la organización con estructura de árbol, cada clase puede tener a lo sumo una superclase. Sin embargo, hay situaciones que no pueden representarse bien en una jerarquía de clases con estructura de árbol.

La herencia múltiple permite a las clases heredar variables y métodos de múltiples superclases. La relación entre clases y subclases se representa mediante un grafo acíclico dirigido (GAD) en el que las clases pueden tener más de una superclase**.**

Cuando se utiliza la herencia múltiple aparece una ambigüedad potencial si se puede heredar la misma variable o el mismo método de más de una superclase.

##### 2.2.12.3.4 Identidad de los objetos.

Los objetos de las bases de datos orientadas a objetos suelen corresponder a entidades del sistema modelado por la base de datos. Las entidades conservan su identidad aunque algunas de sus propiedades cambien con el tiempo. De manera parecida, los objetos conservan su identidad aunque los valores de las variables o las definiciones de los métodos cambien total o parcialmente con el tiempo. Este concepto de identidad no se aplica a las tuplas de las bases de datos relacionales. En los sistemas relacionales las tuplas de una relación sólo se distinguen por los valores que contienen.

La identidad de los objetos es un concepto de identidad más potente que el que suele hallarse en los lenguajes de programación o en los modelos de datos que no se basan en la programación orientada a objetos.

A continuación, se ilustran varios ejemplos de identidad:

* Valor. Se utiliza un valor de datos como identidad. Esta forma de identidad se utiliza en los sistemas relacionales. Por ejemplo, el valor de la clave primaria de una tupla identifica a la tupla.
* Nombre. Se utiliza como identidad un nombre proporcionado por el usuario. Esta forma de identidad suele utilizarse para los archivos en los sistemas de archivos. Cada archivo recibe un nombre que lo identifica de manera unívoca, independientemente de su contenido.
* Incorporada. Se incluye el concepto de identidad en el modelo de datos o en el lenguaje de programación y no hace falta que el usuario proporcione ningún identificador. Esta forma de identidad se utiliza en los sistemas orientados a objetos. Cada objeto recibe del sistema de manera automática un identificador en el momento en que se crea

La identidad de los objetos es una noción conceptual; los sistemas reales necesitan un mecanismo físico que identifique los objetos de manera unívoca. Para los seres humanos se suelen utilizar como identificadores los nombres, junto con otra información como la fecha y el lugar de nacimiento. Los sistemas orientados a objetos proporcionan el concepto de identificador del objeto para identificar a los objetos. Los identificadores de los objetos son únicos; es decir, cada objeto tiene un solo identificador y no hay dos objetos que tengan el mismo identificador.

#### 2.2.12.4 Lenguajes de bases de datos

Un sistema de bases de datos proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones. En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes separados; en su lugar simplemente forman partes de un único lenguaje de bases de datos, tal como SQL, ampliamente usado.

##### 2.2.12.4.1 lenguaje de definición de datos

Un esquema de base de datos se especifica meidnate un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (LDD).

Un diccionario de datos contiene metadatos, es decir, datos acerca de los datos. El esquema de una tabla es un ejemplo de metadatos. Un sistema de base de datos consulta el diccionario de datos antes de leer o modificar los datos reales. Especificamos el almacenamiento y los métodos de acceso usados por el sistema de bases de datos por un conjunto de instrucciones en un tipo especial de LDD denominado lenguaje de almacenamiento y definición de datos. Estas instrucciones definen los detalles de implementación de los esquemas de base de datos, que se ocultan usualmente a los usuarios.

##### 2.2.12.4.2 Lenguaje de manipulación de datos

La manipulación de datos es:

* + La recuperación de información almacenada en la base de datos.
  + La inserción de información nueva en la base de datos.
  + El borrado de información de la base de datos.
  + La modificación de información almacenada en la base de datos.

Un lenguaje de manipulación de datos (LMD) es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado. Hay dos tipos básicamente:

• LMDs procedimentales. Requieren que el usuario especifique qué datos se necesitan y cómo obtener esos datos.

• LMDs declarativos (también conocidos como LMDs no procedimentales). Requieren que el usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.

Los LMDs declarativos son más fáciles de aprender y usar que los LMDs procedimentales. Sin embargo, como el usuario no especifica cómo conseguir los datos, el sistema de bases de datos tiene que determinar un medio eficiente de acceder a los datos. El componente LMD del lenguaje SQL es no procedimental.

Una consulta es una instrucción de solicitud para recuperar información. La parte de un LMD que implica recuperación de información se llama lenguaje de consultas. Aunque técnicamente sea incorrecto, en la práctica se usan los términos lenguaje de consultas y lenguaje de manipulación de datos como sinónimos.

#### 2.2.12.5 Usuarios y administradores de la base de datos

# capitulo 3. DESARROLLO

# PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

Se debe comenzar una breve introducción que indique el propósito de esta sección del documento.

Considerando las características del proyecto realizado, en esta sección se deberá presentar una descripción de las actividades reales realizadas durante la residencia donde se plasme lo realizado para el cumplimiento de los objetivos y que permita constatar las 500 horas realizadas de la residencia, para lo cual se sugiere tomar como **referente el modelo, metodología utilizada para el desarrollo del proyecto, y que previamente se indicó de manera teórica en la sección de marco investigativo.**

Por ejemplo, si el proyecto fue el desarrollo de un sistema de información, se considerarían las fases generales de análisis y diseño indicadas en el modelo o metodología elegida, tales como: definición de requisitos, análisis, diseño, codificación, implementación y pruebas. Y para cada una de esas fases indicar: ¿Qué actividad se hizo? ¿Cómo se hizo? ¿Qué resultados parciales se obtuvieron derivados de cada actividad o fase?

Ésta es una sección muy importante ya que se especifican los resultados generalmente de acuerdo a la metodología adoptada. Cuando se tienen resultados repetitivos, se acostumbra a detallar uno como ejemplo y los demás resumirlos en forma de una tabla o gráfica. Cuando son muchos resultados se pueden poner en un anexo y dejar en esta sección los más importantes o representativos.

Lo más importante a tener en mente, es que esta sección debe mantener la cualidad de *legibilidad*, es decir, uno debe poder leerla. Nadie leería siete tablas de números que ocupan cuatro páginas cada una. Tablas así se deben mandar la sección de anexos, donde el lector interesado pueda consultarlas, pero que no abulten la sección de resultados.

Las gráficas, impresiones de pantalla, mapas de sitio web, mapas de bases de datos y similares deberán estar en un formato legible. Si se trata de impresiones de pantalla eliminar en la imagen la barra donde indica el navegador web que se está utilizando, lo mismo para el pié o lateral de imagen, que no aparezca la barra de tareas del sistema operativo.

Las imágenes o logotipos representativos de marcas comerciales o incluso de software libre no deben incluirse como una figura. Por ejemplo: logotipo de MySql.

Los resultados deben ir siendo explicados conforme se presentan. Todas las tablas y figuras deben mencionarse en el texto.

1. **RESULTADOS**

Se debe comenzar una breve introducción que indique el propósito de esta sección del documento.

Ésta es una sección muy importante ya que se especifican los resultados generalmente de acuerdo a la metodología adoptada. Cuando se tienen resultados repetitivos, se acostumbra a detallar uno como ejemplo y los demás resumirlos en forma de una tabla o gráfica. Cuando son muchos resultados se pueden poner en un anexo y dejar en esta sección los más importantes o representativos.

Lo más importante a tener en mente, es que esta sección debe mantener la cualidad de *legibilidad*, es decir, uno debe poder leerla. Nadie leería siete tablas de números que ocupan cuatro páginas cada una. Tablas así se deben mandar la sección de anexos, donde el lector interesado pueda consultarlas, pero que no abulten la sección de resultados.

Las gráficas, impresiones de pantalla, mapas de sitio web, mapas de bases de datos y similares deberán estar en un formato legible. Si se trata de impresiones de pantalla eliminar en la imagen la barra donde indica el navegador web que se está utilizando, lo mismo para el pié o lateral de imagen, que no aparezca la barra de tareas del sistema operativo.

Las imágenes o logotipos representativos de marcas comerciales o incluso de software libre no deben incluirse como una figura. Por ejemplo: logotipo de MySql.

Los resultados deben ir siendo explicados conforme se presentan. Todas las tablas y figuras deben mencionarse en el texto. Se debe incluir los resultados, planos, gráficas, prototipos, manuales, programas, análisis estadísticos, modelos matemáticos, simulaciones, normatividades, regulaciones y restricciones, entre otros.

Para proyectos que por su naturaleza lo requieran: se debe incluir estudio de mercado, estudio técnico y estudio económico.

# CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA

De todo el documento, ésta es la parte más importante. Muchas veces, una persona va a leer de entrada los objetivos y luego saltarse directamente a las conclusiones, y así abarca lo que se quería lograr y lo que se logró. Por esta razón, ***las conclusiones deben compaginarse con los objetivos,*** de tal modo que para cada objetivo específico haya al menos una conclusión claramente identificable.

Esta sección, no implica decir de nuevo qué se hizo o si salió bien, aunque sí es conveniente mencionar aquí de nuevo el resultado más importante o significativo, para aumentar el énfasis en el éxito del proyecto.

Hay básicamente dos tipos de recomendaciones. El primer tipo es lo que se obtiene directamente como valor agregado del trabajo de residencia para la organización o departamento donde se llevó a cabo la residencia. El segundo tipo de recomendaciones es el trabajo a futuro, por ejemplo: ¿Qué mejoras se pueden hacer al proyecto realizado? ¿Qué nuevas aplicaciones se puede encontrar a esta idea? La respuesta a estas preguntas puede representar tu propuesta para que **el trabajo de residencia se convierta en tu proyecto de titulación**.

# Competencias DESARROLLADAS Y/O APLICADAS EN EL PROYECTO

En ésta se especifican las competencias que el residente íntegro y administro para contribuir a la productividad y el logro de los objetivos estratégicos de la organización, así como aquellas que contribuyen a su perfil profesional como Ingeniero en Tecnologías de Información y Comunicaciones del ITTla.

# Referencias (FUENTES DE INFORMACIÓN)

Usar el estilo de referencia la norma APA.

A continuación, se muestran algunos ejemplos de referencias bibliográficas, incluyendo artículos científicos, libros, capítulos en libros, tesis y reportes de residencia. Emplear el estilo “Bibliografía” y ordenar alfabéticamente. Cuando la referencia sea localizable en Internet, agregar la dirección completa de la página donde se encuentre la información, con el estilo “Bibliografía-Enlace”.

Asociación Mexicana de Internet. (2013). *Estudio de Comercio Electrónico México 2013.* México: VISA.

*Gobierno Abierto.* (22 de Agosto de 2014). Obtenido de http://www.gobabiertocdmx.org/#!procesos/cqh

ISO/IEC. (Guía 2:2005). *ISO/IEC 17799Tecnología de la Información - Técnicas de seguridad - Código para la práctica de la gestión de la seguridad de la información.*

National Institute of Standards and Technology (NIST). (26 de Noviembre de 2001). *Advanced Encryption Standard (AES) FIPS PUB-197.* (NIST, Ed.) Recuperado el 28 de Junio de 2017, de Advanced Encryption Standard (AES) FIPS PUB-197: https://doi.org/10.6028/NIST.FIPS.197

**Anexo A. lista de estilos disponibles**

A continuación se presentan recomendaciones para dar formato al documento.

**Viñetas**

* Lista con viñetas con punto negro como el que se usa en esta lista
* Usar este tipo de lista para cuando no importe el orden.

Cuando se mencionen procedimientos numerarlos:

Listar todos los pasos de los procedimientos en listas numeradas.

Usar verbos en infinitivo (pesar, transferir, etcétera).

No olvidar el punto al final de cada paso.

**Figuras**

Se entiende por figura cualquier elemento gráfico que no sea una tabla o una ecuación. En esta categoría se incluyen diagramas esquemáticos, fotografías, organigramas, etcétera. Hay varias maneras de conseguir que la figura quede colocada en el sitio adecuado, cada una con sus ventajas y desventajas. Aquí se describe la manera más simple y confiable (en opinión personal).

Colocar la figura en línea con el texto (no flotando), en su propio párrafo con el estilo **Figura**. La leyenda va en el párrafo siguiente, con el estilo **Figura Leyenda**. Finalmente, a la parte “Figura #.#” aplicarle el estilo **Figura Número**. Este estilo tiene también un borde rojo visible sólo en Word que ayuda al revisar que todas las figuras estén correctamente numeradas y referenciadas. Al hacer referencia a una figura, siempre usar inicial mayúscula. Como ejemplo, consultar la Figura 1.1.

Hay dos maneras de incluir una referencia a una figura. La primera, mostrada en el párrafo anterior, es parte del mismo texto del documento. La segunda, es colocar la referencia entre paréntesis (Figura 1.1). Un error frecuente es indicar que la figura se debe ver (véase la Figura 1.1). Esto es redundante, ya que necesariamente se va a tener que ver la figura, no es necesario indicarlo.

**Tablas**

Las tablas van centradas y pueden tener un tamaño de letra más pequeño, siempre y cuando sean legibles. La leyenda debe ir en la parte superior de la tabla, puede ir en una celda sin bordes para asegurarse que no exceda el ancho de la tabla. Hay un poco de flexibilidad en el estilo de los bordes (evitando los muy garigoleados), pero sí es importante verificar que todas las tablas sean consistentes en el estilo de bordes. Las unidades pueden incluirse en el renglón de cabecera. El estilo **Tabla Leyenda** incluye espacio adicional para evitar que la tabla se pegue al párrafo previo. La Tabla 3.1 se presenta como ejemplo.

|  |  |  |
| --- | --- | --- |
| **Tabla 3.1** Medidas de empaques para columnas de absorción | | |
| **Tipo de Empaque** | **Área por unidad de volumen (m²/m³)** | **Factor de empaque (–)** |
| Anillos Raschig ½ plg | 33.8 | 580 |
| Sillas Berl ½ plg | 43.3 | 240 |
| Sillas Intalox ½ plg | 57.9 | 200 |

Fuente: Elaboración propia

**Ecuaciones**

Las ecuaciones son una parte muy importante de un documento técnico. Independientemente de que sea una ecuación en todo el reporte o que haya cientos de ellas, se debe poner atención a los detalles para mantener una presentación de calidad. Hay que emplear el Editor de Ecuaciones o MathType (una versión más completa del editor de ecuaciones).

Al insertar una ecuación en su propio párrafo, emplear el estilo “Ecuación”. Este estilo incluye dos tabulaciones, una al centro para la ecuación y otra en el extremo izquierdo para el número, así como espacios adecuados antes y después de la ecuación.

 (3.1)

Cuando el párrafo que sigue de la ecuación es continuación del párrafo anterior, no debe llevar sangría. La mejor manera de asegurarse que sea así es emplear el estilo “Parrafo continúa” para el segundo párrafo. Por ejemplo, si se tiene la ecuación:

 (3.2)

donde  y  son dos variables cualquiera, se puede ver que en este párrafo aún se está desarrollando la misma frase que en el párrafo anterior (por eso se empleó el estilo “Párrafo continúa”).

También es importante que los símbolos sean consistentes en su apariencia. Si se emplea Times New Roman cursiva para una variable (por ejemplo, ) no debe aparecer luego en el texto como Arial sin cursiva (x) o como Times New Roman negrita (**x**).

Finalmente, a continuación se listan recomendaciones de todos los estilos definidos para este documento.

| **Estilo** | **Descripción** |
| --- | --- |
| Bibliografía | Normal + Sangría: Izquierda: 0", Sangría francesa: 0.5", Espacio Antes: 18 pto, Después: 6 pto, Conservar líneas juntas |
| Bibliografía-Enlace | Normal + Fuente: Code, 9 pt, Sangría: Izquierda: 0.5", Izquierda, Conservar líneas juntas |
| Ecuación | Estilo del párrafo siguiente: Párrafo.  Normal + Izquierda, Interlineado: 1.5 líneas, Espacio Antes: 9 pto, Después: 9 pto, No agregar espacio entre párrafos del mismo estilo, Conservar líneas juntas, Tabulaciones: 3.12", Centrado + 6.25", Derecha |
| Encabezado | Normal + Fuente: Cursiva |
| Figura | Estilo del párrafo siguiente: FiguraLeyenda.  Normal + Centrado, Espacio Antes: 30 pto, Después: 18 pto, Conservar con el siguiente |
| FiguraLeyenda | Estilo del párrafo siguiente: Normal.  Figura + Fuente: 10 pt, Sangría: Izquierda: 1", Derecha: 1", Espacio Antes: 12 pto, Después: 30 pto, No conservar con el siguiente, Conservar líneas juntas |
| FiguraNúmero | Fuente de párrafo predeterminado + Fuente: Negrita, Fila de hormigas rojas |
| Indice Figuras-Tablas | Indice1 + Sangría: Izquierda: 0", Sangría francesa: 0.88", Espacio Antes: 3 pto |
| Indice1 | Normal + Sangría: Izquierda: 0", Sangría francesa: 0.38", Derecha: 1", Izquierda, Espacio Antes: 15 pto, Después: 3 pto, Conservar líneas juntas, Tabulaciones: 0.38" Izquierda + 6.25" Derecha, Relleno: … |
| Indice2 | Indice1 + Sangría: Izquierda: 0.38", Sangría francesa: 0.37", Espacio Antes: 3 pto, Tabulaciones: 0.75" Izquierda + No en 0.38" |
| Indice3 | Indice2 + Sangría: Izquierda: 0.75", Sangría francesa: 0.5", Tabulaciones: 1.25" Izquierda + No en 0.75" |
| Lista Numerada | Lista Viñetas + Numerado + Nivel: 1 + Estilo de numeración: 1, 2, 3, … + Iniciar en: 1 + Alineación: Izquierda + Alineación: 0.25" + Tabulación después de: 0.5" + Sangría: 0.5" |
| Lista Viñetas | Normal + Sangría: Izquierda: 0.25", Sangría francesa: 0.25", Interlineado: 1.5 líneas, Espacio Antes: 18 pto, Después: 18 pto, No agregar espacio entre párrafos del mismo estilo, Tabulaciones: 0.5", Lista con tabulaciones, Con viñetas + Nivel: 1... |
| Normal | Fuente: (Predeterminado) Arial, 12 pt, Español (México), Justificado, Interlineado: sencillo, Control de líneas viudas y huérfanas |
| ObjetivoGeneral | Párrafo + Fuente: Cursiva, Sangría: Izquierda: 0.75", Primera línea: 0", Derecha: 0.75", Espacio Antes: 24 pto, Después: 24 pto, Conservar líneas juntas |
| Párrafo | Normal + Sangría: Primera línea: 0.5", Interlineado: 1,5 líneas |
| Párrafo continúa | Estilo del párrafo siguiente: Párrafo.  Párrafo + Sangría: Primera línea: 0" |
| Pie de página | Normal + Fuente: Cursiva |
| Resaltar | Fuente de párrafo predeterminado + Fila de hormigas rojas |

|  |  |
| --- | --- |
| **Estilo** | **Descripción** |
| TablaLeyenda | FiguraLeyenda + Sangría: Izquierda: 0", Derecha: 0", Izquierda, Espacio Antes: 30 pto, Después: 12 pto |
| TablaNúmero | Fuente de párrafo predeterminado + Fuente: Negrita, Fila de hormigas rojas |
| Título 1 | Estilo del párrafo siguiente: Párrafo  Normal + Fuente: 14 pt Negrita, Mayúsculas, Color de fuente: Color personalizado(RGB(0,0,102)), Espacio ajustado en 16 pt, Centrado, Espacio Antes: 6 pto, Después: 24 pto, No agregar espacio entre párrafos del mismo estilo, Salto de página anterior, Mantener con el siguiente, Mantener líneas juntas, Nivel 1 |
| Título 2 | Estilo del párrafo siguiente: Párrafo  Título 1 + Sin Mayúsculas, Izquierda, Espacio Antes: 24 pto, Después 18 pto, Sin salto de página anterior, Nivel 2 |
| Título 3 | Estilo del párrafo siguiente: Párrafo  Título 2 + Fuente: 13 pt Sin Negrita, Cursiva, Espacio Antes: 18 pto, Después: 12 pto, Nivel 3 |