



**Universidad Autónoma del Estado de México**

Centro Universitario UAEM Valle de Chalco

**DISEÑO DE UN SISTEMA DE INFORMACIÓN EN  
VIRTUD DE MEJORAR LA ADMINISTRACIÓN DE  
DS WRAPS.**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE**

***INGENIERO EN COMPUTACIÓN***

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

FRANCISCO JAVIER MAGDALENO SANCHEZ

**ASESOR:**

MTRO. EN C.C. RICARDO JAVIER BUCIO LÓPEZ

Revisor: I. EN E. EDUARDO HERNÁNDEZ ALVAREZ

Revisor: M. EN C.C GUSTAVO HERNÁNDEZ RUVALCABA

VALLE DE CHALCO SOLIDARIDAD, MÉXICO

FEBRERO 2023



**CUVCH**

**DISEÑO DE UN SISTEMA DE INFORMACIÓN EN VIRTUD  
DE MEJORAR LA ADMINISTRACIÓN DE DS WRAPS.**

# INDICE

|        |   |    |
|--------|---|----|
| I.     | Resumen .....                                       | 10 |
| II.    | Antecedentes de la temática.....                    | 11 |
| III.   | Importancia del problema .....                      | 14 |
| IV.    | Planteamiento del problema.....                     | 15 |
| V.     | Marco conceptual o teórico.....                     | 17 |
| 5.1.   | Requisitos del Sistema.....                         | 20 |
| 5.1.2. | Requisitos mínimos java .....                       | 20 |
| 5.1.3. | Requerimientos recomendados .....                   | 21 |
| 5.2.   | Sistemas de información.....                        | 21 |
| 5.2.1. | Sistema de información basado en computadoras ..... | 23 |
| 5.3.   | Tipos de sistemas de organización.....              | 25 |
| 5.4.   | Ciclo de vida del Sistema.....                      | 28 |
| 5.4.1. | Modelo en cascada. ....                             | 29 |
| 5.5.   | Bases de datos .....                                | 31 |
| 5.5.1. | Evolución de la base de datos. ....                 | 31 |
| 5.5.2. | Tipos de bases de datos. ....                       | 31 |
| 5.5.3. | Desafíos de las bases de datos.....                 | 33 |
| 5.6.   | MySQL .....   | 34 |
| 5.6.1. | SQL.....  | 35 |
| 5.7.   | Programación orientada a objetos .....              | 36 |
| 5.8.   | Java.....   | 38 |
| 5.9.   | UML.....  | 39 |
| 5.10.  | Diagramas de Caso de Uso .....                      | 41 |
| 5.11.  | Diagrama de Secuencia.....                          | 41 |
| 5.12.  | Diagrama de Clases.....                             | 41 |
| VI.    | Metodología Técnicas de Investigación .....         | 42 |
| VII.   | Presentación y discusión de resultados.....         | 44 |
| 7.1    | Análisis de requerimientos. ....                    | 44 |
| 7.2.   | Software por utilizar.....                          | 47 |

|          |   |    |
|----------|---|----|
| 7.3.     | Procedimiento.....                          | 47 |
| 7.3.1.   | Arquitectura y Diseño del sistema.....      | 49 |
| 7.4.     | Diagrama Caso de Uso.....                   | 49 |
| 7.4.1.   | Modelo de la Base de datos.....             | 51 |
| 7.4.2.   | Modelo Relacional.....                      | 51 |
| 7.4.2.1. | Tabla Ingresa.....                          | 54 |
| 7.4.2.2. | Tabla proveedores.....                      | 55 |
| 7.4.2.3. | Tabla empleados.....                        | 55 |
| 7.4.2.4. | Tabla productos.....                        | 56 |
| 7.4.2.5. | Tabla pedidos.....                          | 57 |
| 7.4.3.   | Data Access Object (DAO).....               | 58 |
| 7.5.     | Vistas.....                                 | 59 |
| 7.5.1.   | Vista menú principal de administrador.....  | 59 |
| 7.5.2.   | Vista Nueva Nota.....                       | 60 |
| 7.5.3.   | Vista Registrar Producto.....               | 62 |
| 7.5.4.   | Vista Registrar Empleado.....               | 63 |
| 7.5.5.   | Vista Registrar Proveedor.....              | 63 |
| 7.5.6.   | Vista Registrar Pedido.....                 | 64 |
| 7.5.7.   | Vista Consultar.....                        | 65 |
| 7.5.8.   | Vista Búsqueda.....                         | 69 |
| 7.6.     | Controladores.....                          | 72 |
| 7.7.     | Instalación de software principal.....      | 73 |
| 7.8.     | Desarrollo del sistema.....                 | 73 |
| 7.8.1.   | Creación de métodos con java interface..... | 79 |
| 7.8.2.   | Implementando clases interface.....         | 81 |
| 7.8.3.   | Creación Modelos DT.....                    | 84 |
| 7.8.4.   | Modelo Facade.....                          | 88 |
| VIII.    | Conclusiones y sugerencias.....             | 92 |
| IX.      | Referencias de consulta.....                | 94 |

# INDICE DE ILUSTRACIONES

|  |    |
|--|----|
| Figura 1 Conceptos sistemas de información<br>( <a href="http://www.oceania.fc.unam.mx/rcastrou7_TIC.pdf">www.oceania.fc.unam.mx/rcastrou7_TIC.pdf</a> ).....  | 23 |
| Figura 2 Diseño conceptual del funcionamiento de un sistema de información (Senn, J., 1997).....   | 25 |
| Figura 3 Tipos de sistemas según Kendall & Kendall (Kendall, 1997).....  | 27 |
| Figura 4 Modelo en cascada (Piattini, 2004).....   | 30 |
| Figura 5 Herencia simple y múltiple (Joshep, L.2009).....  | 38 |
| Figura 6 Logotipo Tecnología java ( <a href="http://www.java.com/es/about/">www.java.com/es/about/</a> ).....  | 39 |
| Figura 7 Historia de UML (Booch, OMT, OOSE.1993).....  | 40 |
| Figura 8 Modelo vista controlador ( <a href="https://wordpress.modelo-vista-controlador-mvc.pdf">https://wordpress.modelo-vista-controlador-mvc.pdf</a> )<br>.....   | 48 |
| Figura 9 Representación validación exitosa y fallida (Fuente: Propia,2022) .....   | 49 |
| Figura 10 Diagrama de casos de uso (Figura: Propia, 2019).....   | 50 |
| Figura 11 Modelo de base de datos del sistema Ds Wraps (Fuente: Propia,2022). ....   | 53 |
| Figura 12 Ingresa DTO (Fuente: Propia,2022). ....  | 54 |
| Figura 13 Proveedores DTO (Fuente: Propia,2022).....   | 55 |
| Figura 14 Empleados DTO (Fuente: Propia,2022). ....  | 56 |
| Figura 15 Productos DTO (Fuente: Propia,2022).....   | 57 |
| Figura 16 Pedidos DTO (Fuente: Propia, 2022). ....   | 57 |
| Figura 17 Diagrama Data Access Object. (Fuente:<br><a href="https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/">https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/</a> , 2019<br>..... | 59 |
| Figura 18 Vista principal del administrador (Fuente: Propia, 2022). ....   | 60 |
| Figura 19 Vista nueva nota (Fuente: Propia, 2022).....   | 61 |
| Figura 20 Vista nueva nota validación campos vacíos (Fuente: Propia, 2022) .....   | 61 |
| Figura 21 Vista registrar producto (Fuente: Propia, 2022) .....  | 62 |
| Figura 22 Vista registrar empleados (Fuente: Propia, 2022).....  | 63 |
| Figura 23 Vista registrar proveedor (Fuente: Propia, 2022).....  | 64 |
| Figura 24 Vista registrar pedido (Fuente: Propia, 2022).....   | 65 |
| Figura 25 Vista Consultar todo, Pedidos (Fuente: Propia, 2022).....  | 66 |
| Figura 26 Vista Consultar todo. Productos (Fuente: Propia, 2022).....  | 67 |
| Figura 27 Vista Consultar todo, Proveedores (Fuente: Propia, 2022) .....   | 67 |

|  |    |
|--|----|
| Figura 28 Vista Consultar todo, Nota / Clientes (Fuente: Propia, 2022) .....                           | 68 |
| Figura 29 Vista Consultar todo, Empleados (Fuente: Propia, 2022) .....                                 | 68 |
| Figura 30 Vista Búsqueda, Nota / Clientes (Fuente: Propia, 2022) .....                                 | 69 |
| Figura 31 Vista Búsqueda, Nota / Clientes, Datos (Fuente: Propia, 2022) .....                          | 70 |
| Figura 32 Vista Búsqueda, Nota / Clientes, Generar PDF (Fuente: Propia, 2022) .....                    | 71 |
| Figura 33 Vista Búsqueda, Nota / Clientes, Generar PDF, Guardar como (Fuente: Propia, 2022) .....      | 71 |
| Figura 34 Vista Búsqueda, Nota / Clientes, Generar PDF, Imprimir (Fuente: Propia, 2022) .....          | 72 |
| Figura 35 Inicio de Sesión MySQL (Fuente: Propia, 2022) .....  | 74 |
| Figura 36 Bases de datos disponibles en mysql. (Fuente: Propia, 2022) .....                            | 74 |
| Figura 37 Descripción de comandos use y show tables en mysql (Fuente: Propia, 2022) .....              | 75 |
| Figura 38 Creación de tablas en mysql (Fuente: Propia, 2022) .....                                     | 76 |
| Figura 39 Creación de proyecto nuevo en Java Application (Fuente: Propia, 2022) .....                  | 77 |
| Figura 40 Creación de proyecto, Ruta de proyecto y creación de Main Class (Fuente: Propia, 2022) ..... | 78 |
| Figura 41 Main Class de DS Wraps. (Fuente: Propia, 2022) .....   | 78 |

## INDICE DE TABLAS

|  |    |
|--|----|
| Tabla 1 Conexión hacia base de datos (Fuente Propia, 2022) .....   | 80 |
| Tabla 2 Creación de métodos para manejo de datos (Fuente Propia, 2022)....   | 81 |
| Tabla 3 Método Insert para ingresar nuevos datos (Fuente Propia, 2022) .....   | 82 |
| Tabla 4 Método Select para realizar búsqueda de información (Fuente Propia, 2022).....                                   | 82 |
| Tabla 5 Método Select_All muestra una lista con la información almacenada (Fuente Propia, 2022) .....                    | 83 |
| Tabla 6 Método Delete elimina registros de la base de datos (Fuente Propia, 2022).....                                   | 83 |
| Tabla 7 Método Update actualiza datos en la base de datos (Fuente Propia, 2022).....                                     | 84 |
| Tabla 8 Creación de clase Cliente Dt (Fuente Propia, 2022) .....   | 88 |
| Tabla 9 Creación de clase facade e instanciación de todas las clases que integran el sistema (Fuente Propia, 2022) ..... | 91 |

## I. Resumen

El presente trabajo de tesis trata sobre diseñar y desarrollar un sistema de información en virtud de mejorar el proceso administrativo que se lleva a cabo en la empresa Ds Wraps en este caso las pruebas al sistema se realizaron en las instalaciones de la misma empresa gracias a esto se constató que el sistema funciona de manera correcta y cumple con las necesidad y objetivo principal.

La idea de diseñar e implementación este sistema de información, nace debido a los problemas que presentaba la empresa mes con mes con respecto al manejo erróneo de su información, ya que esta se llevaba mediante hojas de papel impresas que se almacenaban en carpetas y posteriormente se colocaban en anaqueles, se utilizaba una carpeta por mes y en ellas se concentraba toda la información sin importar si esta era referente a clientes, proveedores, materia prima, inventarios, pagos de servicios o información referente al personal, los problemas que se generaban por llevar este tipo de administración cada vez eran más ya que cuando se quería consultar algo sobre pedidos de clientes pasados o sobre inventario de materiales utilizados había mucha falta de información y en el peor de los casos no se encontraba el expediente. Otro motivo fue el de disminuir el uso excesivo de papel y tinta de impresora implementando un sistema que brinde mejores resultados en todo momento ya sea que se necesite consultar información por parte del personal administrativo, consultar el estatus del inventario, ventas realizadas o historial de clientes, todo de manera más sencilla y centrada.

Para el desarrollo de este sistema de información, se empleó el uso de lenguaje de programación orientada a objetos, mismo que trabaja de forma local ya que no se requería acceso de forma remota o acceso desde la web.

## II. Antecedentes de la temática

La informática, definida por la Real Academia como el conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información por medio de computadoras, requiere previamente contestar a determinadas preguntas sobre el funcionamiento de estas máquinas.

Las computadoras funcionan mediante la unión de 2 pilares esenciales, Software y Hardware. El software se refiere a los programas encargados de procedimientos y secuencias que permiten realizar distintas tareas en un sistema informático. Mientras que el Hardware se denomina a todo componente tangible que compone un equipo de cómputo.

Teniendo esto en cuenta con el desarrollo de software dedicado para realizar tareas específicas y con el estudio de nuevas áreas de conocimiento surge la ingeniería que se define como el conjunto de conocimientos científicos y tecnológicos para la innovación, invención, desarrollo y mejora de técnicas y herramientas para satisfacer las necesidades de las empresas, instituciones y sociedad (Jones,2010).

Dentro de la ingeniería existen varias áreas de estudio, nosotros nos enfocaremos en el área de desarrollo de software debido a que el proyecto se centra en el desarrollo de un sistema informático para una empresa dedicada al sector de publicidad impresa.

Ds Wraps es liderada por su fundador y director Alfredo Damián Sánchez Gracia, se dedican principalmente a la venta e instalación de vinil auto adherible, su principal actividad es desarrollada en su establecimiento dado que es ahí donde la mayor parte de sus clientes llega a solicitar sus servicios, la instalación depende mucho de las especificaciones de cada cliente y automóvil.

Además de la instalación de vinilos en autos también se dedica a la instalación de publicidad la cual puede ser colocada en casi cualquier parte dado que es un material muy versátil, ofrecen servicios de estampado en diferentes materiales, los cuales pueden ser textiles como: camisetas, playeras, manteles y overoles, donde se quiera representar una idea, marca, empresa, señalamientos, rotulación en tazas, vasos, botellas, donde se quiera poner recuerdos, dibujos, ideas, etc.

Hay una gran variedad de Vinil tales como: adhesivo flexible transparente, blanco, color, acabado brillante, mate, vinil estático, microperforado, efecto día/noche, con adhesivos removible o permanente, opción de liner imprimible. Son aplicados en casas, departamentos, oficinas, edificios, autos, etc. (DTTEC, 2011, p01).

La impresión en vinil autoadhesivo ofrece una gran durabilidad, extraordinaria reproducción de colores y una fácil aplicación que lo convierte en el material de impresión favorito para la rotulación de superficies preferentemente planas, para evitar la acumulación de burbujas (PRINTIKA, 2016).

Anteriormente la empresa no contaba con un sistema informático que llevara el control de su inventario. La administración se llevaba en papel y se apoyaban con el uso de notas de remisión.

El proyecto se basará en la utilización de Software de desarrollo para poder utilizar los servicios que darán soporte a las funciones que vamos a utilizar y que se ofrecerán al usuario al término de este proyecto.

Un sistema de información de escritorio es aquel que podemos instalar en nuestro ordenador sin necesidad de internet dado que la misma aplicación contiene todo lo necesario para funcionar de manera correcta. Al usar una aplicación de escritorio el programador puede plantearse una estructura global de la aplicación, de esta manera se puede adecuar a las necesidades del cliente.

Cualquiera que haya tenido que lidiar con el código fuente de otro programador o incluso con el propio, pasado algún tiempo sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.

A continuación, se presentan algunos puntos que nos indican la falta de un sistema de información:

- Necesidad de un software que ayude a resolver algunas problemáticas centradas en función al área en el que se desarrolló.
- Falta de sistema de información único en las diferentes sucursales y oficinas de la empresa.
- Mejora definitiva de procesos de producción.
- Mejorar Comunicación entre los departamentos de la empresa y de esta manera mejorar el desempeño de cada una de las áreas involucradas.
- Aumentar el control sobre la ejecución de ventas y pedidos.

### **III. Importancia del problema**

Realizando una investigación para conocer y determinar cuáles son las problemáticas más comunes que presenta Ds Wraps, dado que no contaba con un sistema administrador que le ayude a manejar cada uno de los departamentos que la conforman.

El señor Alfredo Damián en ciertas ocasiones necesitaba tener acceso a datos donde encontrara información sobre pedidos pendientes, ordenes hechas a proveedores, inventario, ventas, productos, pero no podía dado que el control sobre su información era muy ineficiente y al querer consultar información le tomaba demasiado tiempo o no tenía forma de acceder a ella. Así que decidió mejorar el sistema con el que trabajaba.

Le propusimos la implementación de un software que se hiciera cargo de la información que él maneja y de esta manera poder tener mejor administración. Así mismo se podrían facilitar las tareas diarias a las que enfrenta, podría manejar su empresa de manera eficiente, rápida y más segura, solo tendría que acceder al sistema sin la necesidad de estar presente en cada área para poder ver las tareas y trabajos de cada día.

## **IV. Planteamiento del problema**

Uno de los principales problemas a los que se enfrenta cualquier institución o empresa es el manejo, uso y control de su información.

En base a una entrevista con el director general cuya función principal es el manejo de toda información relacionada a cada departamento nos comentó que llevaba una administración poco eficiente dado que todos sus pedidos, ventas y proveedores se manejaban mediante papel y como único comprobante las notas generadas para cada situación.

También nos contó el tipo de administración que sus empleados utilizaban para el manejo de sus áreas y las dificultades que enfrentaban.

Necesito un programa que me ayude a llevar la administración de mis productos dado que he tenido muchos problemas al querer saber la disponibilidad de los productos que ofrezco, también quiero que me ayude a llevar un control sobre mis empleados, que me ofrezca la oportunidad de registrarlos para tener sus datos de contacto en caso de que los llegara a necesitar, también quiero almacenar los datos de mis proveedores para que cuando requiera hacer un pedido mis empleados puedan solicitar lo que necesitamos, y no tengan que esperarme para poder contactar con ellos dado que yo tengo los datos de contacto y que me muestre un historial de mis ventas hechas, los pedido que hemos realizado y un historial de clientes (Alfredo Sánchez, 2019, Ds Wraps).

Así que le comentamos sobre desarrollar un sistema que se encargaría en gran parte de esta tarea haciendo que la administración de su empresa sea más eficaz y fácil de llevar. También le sería de ayuda cuando necesite buscar, verificar y/o agregar un pedio a proveedores. A la hora de realizar una venta nos brindara la posibilidad de llevar un control sobre el inventario de los diferentes materiales ofrecidos dándole así la posibilidad de consultar la situación actual de

la existencia en sus productos.

Por lo anterior planteado se proponen las siguientes preguntas de investigación:

¿Es posible diseñar y desarrollar un nuevo sistema de información que cubra las necesidades de esta empresa?

¿Qué tipo de metodología sería la mejor opción para implementar?

¿Cómo el sistema podría ayudar a mejorar la administración?

¿Qué beneficios se obtendrán con la implementación del sistema?

## V. Marco conceptual o teórico

Dado que el presente trabajo se centra en diseñar y desarrollar un sistema de información que mejore la administración de la empresa el cual comprenda cada una de sus áreas, será necesario que mencionemos algunos conceptos técnicos referentes a ingeniería de software que nos ayuden a comprender de mejor manera los términos a utilizar.

Para empezar, debemos hablar sobre los conceptos básicos con los que vamos a trabajar, más allá de la popular definición “Hardware es lo que golpeas cuando falla el software”, vamos a profundizar un poco en cada concepto para ayudar a la lectura.

El Software como (Sommerville, 2005) menciona “Son programas de ordenador y la documentación asociada. Los productos de software se pueden desarrollar para algún cliente en particular o para un mercado en general. Todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.”

Existen diferentes tipos de software tales como: Software de aplicación, Software de Sistema y Software de Programación, estos serán los principales a utilizar en el desarrollo e implementación de nuestro sistema de información dado que cada uno aporta diferentes tipos de herramientas que serán muy útiles a la hora de realizar pruebas tanto en el desarrollo como en la parte final donde se podrá confirmar que el sistema de información funciona de manera correcta, por otro lado si los resultados no son los esperados se podrá ejecutar un análisis crítico que nos ayudara a encontrar la falla y corregirla de manera oportuna.

A continuación, presentaremos las definiciones y daremos ejemplos de los tipos de software antes mencionados.

**Software de aplicación:** En informática, se entiende por software de aplicación, programas de aplicaciones o en algunos casos aplicaciones, al conjunto de los programas informáticos generalmente instalados en el sistema por el usuario, y diseñados para llevar a cabo un objetivo determinado y concreto, de tipo lúdico, instrumental, comunicativo, informativo, etc.

Dicho de otro modo: todos aquellos programas que no tienen que ver con el funcionamiento del computador, sino que instalamos en él para darle funciones determinadas como herramienta de trabajo (hoja de cálculo, procesador de palabras, programas de diseño gráfico, etc.), de ocio (videojuegos, reproductores de audio o video, etc.) o de información (enciclopedias digitales, navegador de Internet, etc.), entre otras.

El software de aplicación normalmente es diseñado y comercializado aparte del software de sistema que viene incorporado de fábrica al computador, y su selección e instalación en el sistema es a total libre albedrío del usuario. No es, sin embargo, imposible que muchos programas de este tipo vengan preinstalados –debido a convenios empresariales y promociones específicas- en el disco rígido de un computador recién comprado. Algunos ejemplos populares de Software de aplicación son:

- **Google Chrome.** Se llama así a este navegador de Internet diseñado y publicado por Google Inc., y distribuido de manera totalmente gratuita por la red. Revolucionó el ramo al ser uno de los exploradores más livianos y veloces del momento.
- **Skype.** Un software diseñado para establecer comunicaciones telefónicas o videoconferencias empleando la transmisión de datos de Internet. Permite al usuario establecer su agenda de contactos, emitir y recibir tanto señales de audio como de video en tiempo real.

- **Adobe Ilustrador.** Diseñado para la edición de gráficos y en representación virtual de un taller de arte, este programa opera en base a un tablero de dibujo llamado “mesa de trabajo”, en el que se lleva a cabo la labor de diseño artístico y creativo, ayudado por diversas herramientas digitales. Es propiedad de Adobe System y se encuentra en el paquete Adobe Creative Cloud.

**Software de sistema:** se distingue del software de aplicación en que no le ofrece al usuario la posibilidad de cumplir sus tareas, sino que se ocupa de mantener funcionando el sistema informático. Dicho de otro modo, se trata de un sistema operativo: un programa diseñado para hacer funcionar la máquina y mantenerla encendida y lista, dispuesta para la ejecución de los programas de aplicación.

Este tipo de software suele venir preinstalado en los computadores, si bien puede ser reemplazado por otros a voluntad del usuario. En ese caso el nuevo software de sistema deberá ser comprado e instalado como con cualquier software de aplicación. Algunos ejemplos de este tipo de programas son:

- **Sistemas operativos.** En realidad, muchos de sus componentes, ya que cierta parte de los SO forman parte de otros aspectos del funcionamiento del sistema.
- **Cargadores de programas (loader).** Programas encargados de la ejecución de otros programas y de garantizar la estabilidad el sistema.
- **BIOS.** Siglas en inglés de Basic Input-Output System (Sistema Básico de Entrada y Salida), es el programa elemental de control de la información del sistema, que garantiza que el computador se comunique con el mundo externo.

**Software de Programación:** se refiere a los programas que sirven para diseñar otros programas. Para ello el usuario o programador cuenta con diversas herramientas de asistencia en la escritura del código de la nueva pieza de software, tales como funciones de prueba o interfaces amigables. Todos los programas existentes provienen de un software de programación, y cada software de este tipo permite “escribir” un lenguaje o protocolo diferente. Algunos ejemplos de este tipo de software son:

- Java script
- C++
- Visual Basic

## **5.1. Requisitos del Sistema**

La determinación de requerimientos es el estudio de un sistema para conocer cómo trabaja y donde es necesario efectuar mejoras. Los estudios de sistemas dan como resultado una evaluación de la forma como trabaja los métodos empleados y si es necesario o posible realizar ajustes. Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esta puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad de la empresa o brindar soporte a la gerencia (Senn, 1992).

### **5.1.2. Requisitos mínimos java**

Para ejecutar aplicaciones javas según la página oficial de Oracle Java si se utiliza un sistema operativo Windows 10, 8.x, 7, vista, Windows server 2008, Windows server 2012. se necesita al menos 128 MB de ram para ejecutar programas de java más aparte con la que cuenta el sistema operativo para poder funcionar la computadora que se utiliza es decir 2 GB de ram mínimos en

arquitecturas de 64 bits y para 32 bits 1 GB ram. En cuanto espacio 22 MB en disco 124 MB para JRE (Java Run time Environment) 2 MB para Java Update, procesador mínimo Pentium 2 a 266 MHz y exploradores de internet explorer 9 o superior, Firefox recomendable entre otros.

### **5.1.3. Requerimientos recomendados**

Los requerimientos funcionales para un sistema refieren lo que el sistema deber hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, los requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera (Sommerville, 2011).

## **5.2. Sistemas de información.**

La información se puede definir como el conjunto de datos que, transformados o modificados, tiene un valor para aquellos usuarios que hacen uso de ellos. A la vez el sistema de información se puede definir como: “Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan elaboran y la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (toma de decisiones), que permitan desempeñar su actividad de acuerdo a su estrategia de negocio”. (Carrera, R. 2002).

El objetivo de un sistema de información es ayudar al desempeño de las actividades que desarrolla la empresa, suministrando la información adecuada, con la calidad requerida, a la persona o departamento que lo solicita, en el momento y lugar especificados con el formato más útil para el receptor. El sistema de información está al servicio de los objetivos establecidos. Para lograr dichos objetivos la empresa y sus individuos adoptan procedimientos y prácticas de trabajo que deben resultar útiles y eficaces.

La forma en que están organizados o dispuestos los elementos del sistema se le llama configuración. De modo muy similar a como ocurre con los datos, las relaciones entre elementos de un sistema se definen por medio del conocimiento.

En la mayoría de los casos, conocer el propósito o resultado que se desea obtener de un sistema es el primer paso en la definición de la manera en que se configuraran sus elementos. Por ejemplo, suponiendo que el resultado deseado de nuestro sistema fuera un auto limpio, Por experiencia sabemos que sería ilógico disponer las cosas en tal forma que el rociador de agua precediera al jabón, pues los pasos del proceso estarían invertidos (enjuagar y luego enjabonar), con lo cual el automóvil no quedaría precisamente limpio.

Tal cómo se deduce de este ejemplo, el conocimiento es necesario tanto para definir las relaciones entre las entradas a un sistema (el auto sucio y las instrucciones del operador) como para organizar los elementos del sistema que se utilizan para procesar entradas (el jabón debe preceder al rociador de agua). Los sistemas pueden clasificarse de acuerdo con numerosas dimensiones; pueden ser simples o complejos; abiertos o cerrados, estables o dinámicos, adaptables o no adaptables, permanentes o temporales, A continuación, se definen sus características:

|  |   |
|--|---|
| Simple<br><b>Poseen pocos componentes, y cuya relación o interacción entre ellos es sencilla y directa</b> | <b>Compleja</b><br>Poseen muchos elementos estrechamente relacionados o interconectados |
| Abierto<br><b>Interactúa con su entorno</b>  | <b>Cerrado</b><br>No interactúa con el entorno  |
| Estable<br><b>Sufre escasos cambios al paso del tiempo</b>   | <b>Dinámico</b><br>Sufre rápidos y constantes cambios al paso del tiempo                |
| Adaptable<br><b>Es capaz de modificarse en respuesta a cambios en el entorno</b>                           | <b>No adaptable</b><br>Es incapaz de modificarse en respuesta a cambios en el entorno   |
| Permanente<br><b>Está diseñado para existir durante un periodo relativamente largo</b>                     | <b>Temporal</b><br>Está diseñado para existir durante un periodo relativamente corto    |

Figura 1 Conceptos sistemas de información ([www.oceania.fc.unam.mx/rcastro/u7\\_TIC.pdf](http://www.oceania.fc.unam.mx/rcastro/u7_TIC.pdf))

### 5.2.1. Sistema de información basado en computadoras

Un sistema de información basado en computadoras (SIBC) está compuesto por hardware, software, bases de datos, telecomunicaciones, personas y procedimientos específicamente configurados para recolectar, manipular, almacenar y procesar datos para ser convertidos en información. Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

**Entrada de Información:** Es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfases automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales USB, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el ratón, entre otras.

**Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, unidades flash.

**Procesamiento de Información:** Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.

**Salida de Información:** Es la capacidad de un sistema de información para mostrar o sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, archivos de audio, los graficadores y en este caso los plotters de impresión en lonas o viniles, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o módulo.

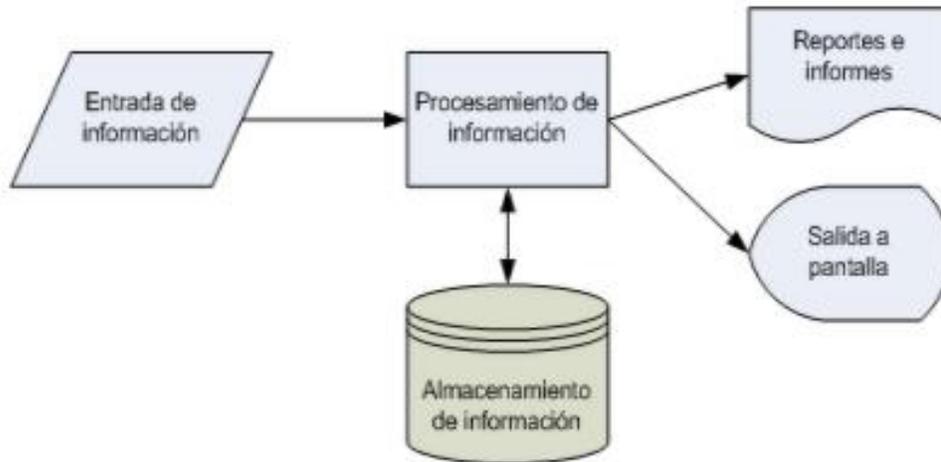


Figura 2 Diseño conceptual del funcionamiento de un sistema de información (Senn, J., 1997).

### 5.3. Tipos de sistemas de organización.

Normalmente los sistemas de información se hacen a la medida. Apenas hace unos pocos años podemos hablar de ingeniería de software como práctica y no solo en forma teórica. Sin embargo, todos los sistemas presentan algunas características comunes, puesto que ciertas actividades de administración de una empresa cuentan con una estructura de sistemas de información.

De acuerdo con los autores Kendall & Kendall (1997) describen una clasificación de sistemas muy interesante que describen en 7 tipos diferentes las cuáles describiremos a continuación.

**Sistemas de procesamiento de transacciones:** Los sistemas de procesamiento de transacciones (TPS) son sistemas de información computarizada desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios, tales como nómina e inventario. Los TPS eliminan el tedio de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se requirió para ejecutarlas manualmente, aunque la gente todavía debe alimentar datos a los sistemas computarizados.

**Sistemas de automatización de oficina y sistema de manejo de conocimiento:** dan soporte a los trabajadores de datos, quienes, por lo general, no crean un nuevo conocimiento, sino que usan la información para analizarla y transformar datos, o para manejarla en alguna forma y luego compartirla dentro de la organización. En estos sistemas se incluyen los procesadores de palabras, hojas de cálculo, editor de publicaciones, calendarización electrónica, correo de voz, correo electrónico y videoconferencias.

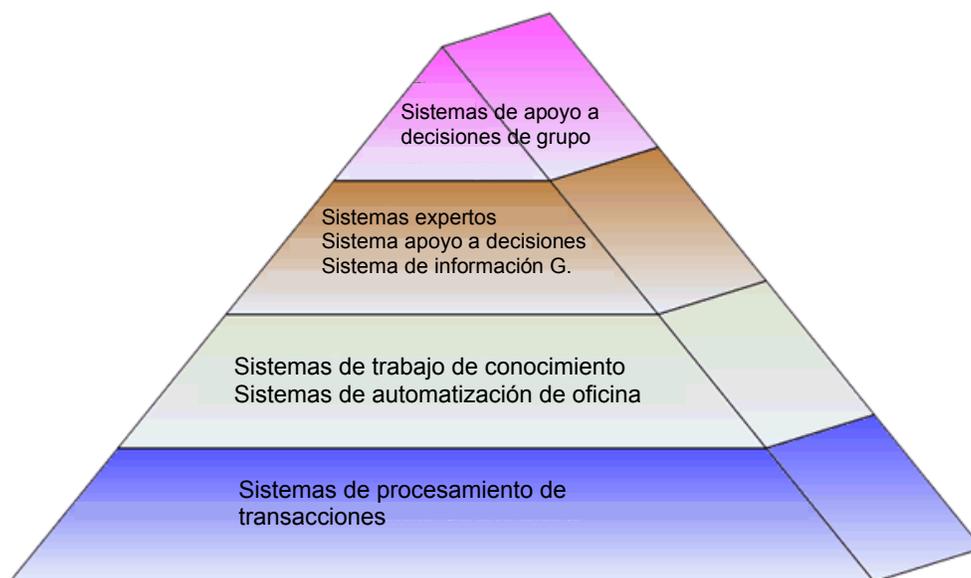
**Sistemas de información gerencial:** Los sistemas de información gerencial (MIS) no reemplazan a los sistemas de procesamiento de transacciones, sino que todos los MIS incluyen procesamiento de transacciones. Los MIS son sistemas de información que trabajan debido a la interacción resuelta entre gentes y computadoras. Requieren que las personas, el software y el hardware, trabajen juntos.

**Sistemas de apoyo a decisiones:** El DSS es similar al sistema de información gerencial tradicional en que ambos dependen de una base de datos como fuente. Un sistema de apoyo a decisiones se aparta del sistema de información gerencial tradicional en que enfatiza el apoyo a la toma de decisiones en todas sus fases, aunque la decisión final todavía es del dominio del usuario asignado.

**Sistemas expertos e inteligencia artificial:** La inteligencia artificial (AI) puede ser considerada la meta de los sistemas expertos. El empuje general de la AI ha sido desarrollar máquinas que se comporten de forma inteligente. Los sistemas expertos usan enfoques del razonamiento de la AI para resolver los problemas que les plantean los usuarios de negocios. Un sistema experto (también llamado un sistema basado en conocimiento) captura de forma efectiva y usa el conocimiento de un experto para resolver un problema particular experimentando en una organización.

**Sistema de apoyo a decisiones de grupo:** Los (GDSS) son usados en cuartos especiales, equipados en varias configuraciones diferentes, que permiten que los miembros del grupo interactúen con apoyo electrónico, frecuentemente en forma de software especializado y con una persona que da facilidades al grupo. Los sistemas de decisiones de grupo están orientados para reunir a un grupo, a fin de que resuelvan un problema con la ayuda de varios como votaciones, cuestionarios, aportación de ideas y creación de escenarios.

**Sistema de apoyo a ejecutivos:** Un (ESS) ayuda a éstos, para organizar sus interacciones con el ambiente externo, proporcionando apoyo de gráficos y comunicaciones en lugares accesibles tales como salas de juntas u oficinas personales corporativas. Aunque los ESS se apoyan en la información generada por los TPS y los MIS, los sistemas de apoyo a ejecutivos ayudan a sus usuarios a que ataquen problemas de decisión sin estructura, que no son específicos de una aplicación, creando un ambiente que les ayude a pensar acerca de los problemas estratégicos de una manera informada.



*Figura 3 Tipos de sistemas según Kendall & Kendall (Kendall, 1997).*

## 5.4. Ciclo de vida del Sistema

El ciclo de vida del desarrollo de sistemas (SDLC, Systems Development Life Cycle) es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario.” (Kendall & Kendall, 1997). Los autores mencionan que este ciclo se divide en 7 fases las cuales se describen a continuación.

**Identificación de problemas, oportunidades y objetivos:** En la primera fase del ciclo del desarrollo de sistemas el analista tiene que ver con la identificación de problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto del proyecto, debido a que nadie quiere desperdiciar el tiempo subsecuente resolviendo el problema equivocado.

**Determinación de los requerimientos de información:** Esto se hace a partir de los usuarios particularmente involucrados, para determinar los requerimientos de información dentro de una organización pueden utilizarse diversos instrumentos, los cuales incluyen: muestreo, el estudio de los datos y formas usadas para la organización, la entrevista, los cuestionarios; la observación de la conducta de quien tomó las decisiones.

**Análisis de las necesidades del sistema:** Se analizan las necesidades propias del sistema. También se analizan las decisiones estructuradas por realizar, nos podremos apoyar con el uso de diagramas de flujo donde podremos representar de manera gráfica y estructurada el proceso de entrada y salida de datos.

**Diseño del sistema recomendado:** Se usa la información recolectada con anterioridad y se elabora el diseño lógico de sistemas de información, esta

etapa también incluye el diseño de los archivos o la base de datos que los almacenara, el analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que entran al sistema de información sean correctos.

Otra parte importante en el diseño del sistema de información es diseñar la interfaz de usuario. La interfaz conecta al usuario con el sistema y es, por lo tanto, extremadamente importante. Por último, el analista debe diseñar procedimientos de control y respaldo para proteger el sistema y a los datos.

**Desarrollo y documentación del software:** Dentro de las técnicas estructuradas para el diseño y documentación del software se tienen: el método HIPO, los diagramas de flujo, los diagramas Nassi-Schneiderman, los diagramas Warnier-Orr y el pseudocódigo es aquí donde se transmite al programador los requerimientos de programación.

**Pruebas y mantenimiento del sistema:** Todo sistema de información debe probarse antes de ser utilizado, ya que el costo es menor si se detectan los problemas antes de que entre en funcionamiento. Algunas de las pruebas son realizadas por los mismos programadores, y otros por los analistas de sistemas.

**Implementación y evaluación del sistema:** Esta es la última etapa del desarrollo del sistema, esto incluye el adiestramiento que el usuario requerirá. Uno de los criterios fundamentales que debe satisfacerse, es que el futuro usuario utilice el sistema desarrollado de la mejor manera y sin complicaciones.

#### **5.4.1. Modelo en cascada.**

La versión original del modelo en cascada del ciclo de vida fue propuesta por Roycem W. (1970 citado en Piattini G.,2004) y, desde entonces, han aparecido numerosos refinamientos y variaciones de dicho modelo. El número de etapas

que se proponen en este ciclo suele variar, aunque suelen ser: análisis de requisitos, análisis, diseño, codificación, pruebas, implantación y mantenimiento.

Según el autor (Piattini G., et al., 2004). Algunas características de este modelo son:

- Cada fase empieza cuando se ha terminado la fase anterior. Es muy importante no atrasarse ya que. No puedes pasar a la siguiente etapa sino has terminado la anterior.
- Ayuda a prevenir que se sobrepasen las fechas de entrega y los costos esperados.
- Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto.
- Aunque es el ciclo de vida más antiguo y el más ampliamente utilizado, debido a las facilidades que da a los gestores para controlar el progreso de los sistemas, ha recibido numerosas críticas.

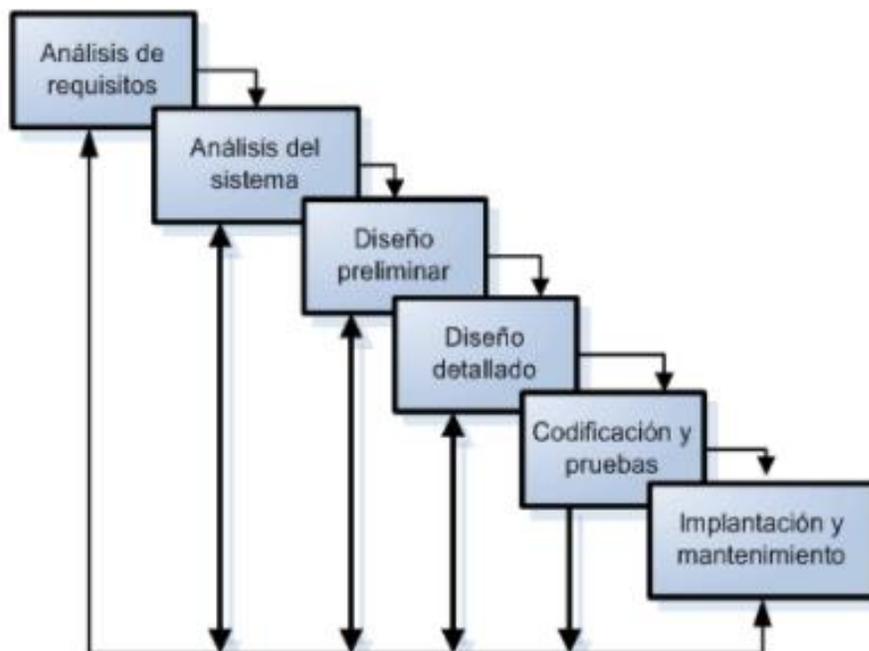


Figura 4 Modelo en cascada (Piattini, 2004)

## **5.5. Bases 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). 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 bases de datos (Luis Alberto,2005).

### **5.5.1. Evolución de la base de datos.**

Los inicios de las bases de datos se remontan al inicio de la década de 1960 donde los sistemas originales que se utilizaban para almacenar y manipular los datos eran principalmente la Base de datos jerárquica, la cual se basa en un modelo de árbol y permitía la una relación de uno a muchos. Y la Base de datos de red, este era un modelo más flexible ya que permitía relaciones múltiples.

En la década de 1980 las bases de datos relacionales se hicieron populares seguidas por las bases de datos orientadas a objetos en la década de 1990. Las bases de datos NoSQL surgieron como respuesta al presente crecimiento del internet de esa época impulsadas por la necesidad de acelerar la velocidad de procesamiento de los datos no estructurados. Hoy en día las bases de datos en la nube y las bases de datos de autogestión están abriendo nuevos horizontes en lo que respecta a la forma en la que los datos son recopilados, almacenado, gestionados y utilizados.

### **5.5.2. Tipos de bases de datos.**

Existes muchos tipos de bases de datos, Siendo la mejor aquella que se adapte mejor a las necesidades específicas de cada organización y brinde la posibilidad

de manejar y resguardar la información a utilizar de manera óptima. A continuación, en listamos los diferentes tipos de bases de datos:

**Bases de datos relacionales.** La tecnología de las bases de datos relacionales proporciona la manera más eficiente y flexible de acceder a la información estructurada. Los elementos de una base de datos relacional se organizan como un conjunto filas y columnas.

**Bases de datos orientadas a objetos.** Los datos almacenados en este tipo de base se representan como objetos, así como es representada en la programación orientada a objetos.

**Bases de datos distribuidas.** La constitución de este tipo de base consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, en la misma ubicación física o almacenarse en diferentes redes.

**Almacenes de datos.** Es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos, la información de este tipo de base de datos se almacena en un repositorio central o en Data warehouse.

**Bases de datos NoSQL.** También conocida como base de datos no relacional. Este tipo de base de datos se hizo popular a medida que las aplicaciones web se volvían más populares y complejas. Permite almacenar y manipular datos no estructurados (a diferencia de las bases de datos relacionales donde se definen como se deben componer los datos insertados a la base de datos).

**Bases de datos en la nube.** Es una recopilación de datos estructurados y no estructurados que reside en una plataforma de cloud computing privada, pública o híbrida. Existen dos tipos de modelos el modelo clásico y el modelo

BDaaS donde un proveedor de servicios realiza las tareas administrativas y de mantenimiento.

**Bases de datos de auto gestión.** También conocida como autónoma es el tipo de base más innovador ya que están basadas en la nube y utilizan el machine learning para automatizar ajustes, las copias de seguridad, actualizaciones y otras tareas rutinarias eran realizadas por los administradores.

### **5.5.3. Desafíos de las bases de datos.**

Hoy en día las bases de datos soportan consultas muy complejas y se espera que proporcionen respuestas casi inmediatas a esas consultas. En consecuencia, se pide a los administradores que empleen una gran variedad de métodos para ayudar a mejorar el rendimiento. Algunos desafíos comunes a los que se enfrentan las bases de datos y administradores incluyen.

- **Absorción de gran volumen de datos.** La gran cantidad de información proveniente de usuarios, sensores, maquinas conectadas y diversas fuentes hacen que los administradores tengan que luchar para administrar y organizar los datos de sus empresas de manera eficiente.
- **Garantía de seguridad de los datos.** El aumento en las filtraciones de datos que tenemos hoy en día y el aumento de ataques por parte de los piratas informáticos, hace que garantizar la seguridad de la información se mas importante que nunca pero también que los usuarios puedan acceder a ella fácilmente.
- **Mantenimiento al día con la demanda.** Las empresas y sus usuarios necesitan tener acceso en tiempo real a sus datos para poder tomar decisiones a tiempo y aprovechar las nuevas oportunidades generadas por el rápido crecimiento del entorno laboral de hoy en día.
- **Gestión y mantenimiento de la base de datos y su infraestructura.**

A medida que las bases de datos se hacen más y más complejas y los volúmenes de datos crecen, las empresas se enfrentan a los desafíos que esto conlleva, los administradores deben supervisar las bases de datos continuamente por si surgen problemas puedan realizar mantenimientos preventivos y correctivos, así como aplicar parches o actualizaciones.

- **Eliminación de límites de escalabilidad.** Un negocio necesita crecer si quiere sobrevivir y su administración de datos necesita crecer con él. Para un administrador resulta muy difícil predecir la capacidad que va a necesitar una empresa o usuario en especialmente con las bases de datos On-premise. Abordar todos estos problemas es de suma importancia y necesitara tiempo para poder desarrollar la base de datos de manera que cumpla de forma óptima las necesidades requeridas y en caso de requerir una actualización esta se pueda realizar sin mayor problema.

## 5.6. MySQL

Es un sistema de gestión de bases de datos relacionales de código abierto basado en SQL. Se diseño y optimizo para las aplicaciones web y puede utilizarse en cualquier plataforma. A medida que surgían nuevos y diferentes requisitos de internet, MySQL se convirtió en la aplicación favorita 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 por las empresas de comercio electrónico que necesitan gestionar múltiples transferencias de dinero. La flexibilidad On-demand es su principal función.

MySQL es el principal DBMS que se encuentra detrás de algunos de los sitios web y aplicaciones basadas en web más importantes del mundo como: Airbnb, Uber, LinkedIn, Facebook, Twitter y YouTube.

### 5.6.1. SQL.

Es el lenguaje diseñado para administrar, recuperar y gestionar los datos almacenados en una base de datos, una de sus principales características es el uso de álgebra y cálculo relacional para efectuar consultas con el fin de recuperar de forma sencilla información almacenada en la base y si así se requiere poder modificarla. Sus siglas SQL en inglés (Structured Query Language). En español significa Lenguaje de Consulta estructurada.

Según los autores (Groff & Weinberg, 2003) dicen que SQL se utiliza para controlar todas las funciones que EL SGBD ofrece a los usuarios entre las que hallan:

**Definición de datos.** Establecer la estructura y la organización de los datos almacenados y las relaciones entre los elementos de datos almacenados.

**Recuperación de datos.** El usuario o un programa de aplicación obtenga de la base de datos los datos almacenados y pueda hacer uso de ellos.

**Manipulación de datos.** Se refiere a la capacidad de modificar, actualizar o eliminar datos previamente almacenados en la base de datos.

**Control de acceso.** Hace referencia a las restricciones aplicadas a los usuarios o programas que intenten obtener información de la base de datos modificarla o añadir nueva. De esta forma se asegura la integridad de la información.

**Comportamiento de los datos.** Se utiliza para coordinar el comportamiento de datos entre usuarios recurrentes, evitando que interfieran entre sí.

**Integridad de datos.** Se refiere a los protocolos de seguridad que sigue la base de datos con la finalidad de asegurar la información que contiene, así en caso de actualizaciones inconsistentes o fallos del sistema la información estará protegida y segura en todo momento.

## **5.7. Programación orientada a objetos**

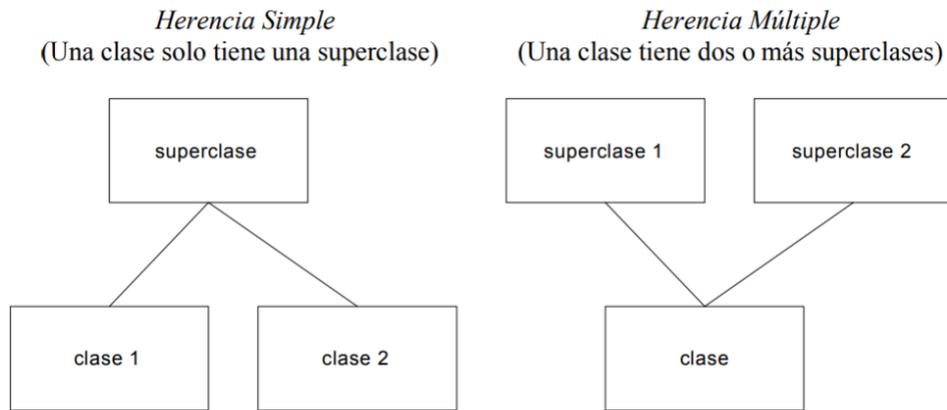
Es una metodología de diseño de software y un paradigma de programación que define los programas en términos “clases de objetos”, objetos que son entidades que combinan estados (datos) y comportamientos (procedimientos o métodos).

La programación orientada a objetos es un programa con un conjunto de objetos que se comunican entre ellos para realizar tareas, podría interpretarse como una representación de un subconjunto del mundo real, tan fielmente como sea posible, de modo fácil y natural donde los objetos van a tener características (atributos) y comportamientos (métodos). Que, a diferencia de los lenguajes procedurales, en donde los datos y procedimientos se encuentran separados y sin relación alguna. (Alan,F.1993).

Los lenguajes procedurales, utilizan funciones y después pasan los datos, mientras que los lenguajes orientados a objetos definen objetos y después envían mensajes a los objetos diciendo que realicen alguno de los métodos especificados previamente.

Una ventaja de la programación orientada a objetos son los métodos que están pensados para hacer programas y módulos sencillos de escribir, mantener y reutilizar, además de poder reutilizar parte del código en nuevos programas. A continuación, describiremos conceptos básicos de la programación orientada a objetos.

- **Objetos:** Se definen como las unidades básicas de construcción, para la conceptualización, diseño o programación, esto hace referencia a que son instancias agrupadas en clases con características en común y que son los atributos y procedimientos, conocidos como operaciones y métodos.
- **Mensajes:** Se refiere a la forma de comunicación entre objetos mediante señales específicas que le ordenan ejecutar los métodos establecidos.
- **Métodos:** también conocidos como función hace referencia a una acción que determina como debe actuar un objeto cuando recibe un mensaje. Un método también puede enviar mensajes a otros objetos, para realizar una acción o una solicitud de información.
- **Clase:** Se define como el conjunto de características (atributos) y comportamientos de todos los objetos que componen a la clase, puede ser expresada de dos maneras Superclase y Subclase.
- **Encapsulamiento:** Se refiere a un método utilizado para aislar objetos con la intención de proteger los datos que este tenga asociados y evitar ser modificados por usuarios no permitidos. De esta manera se conserva la integridad de la información y permite manejar a los objetos como unidades básicas dejando oculta su estructura interna.
- **Herencia:** Es la característica que permite compartir en forma automática los métodos y datos entre clases y subclases de los objetos.



*Figura 5 Herencia simple y múltiple (Joshep, L.2009).*

## 5.8. Java

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la web en un elemento más interesante y útil. Java no es lo mismo que JavaScript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador. Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa. Con más de 9 millones de desarrolladores en todo el mundo, Java le permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios.

Java Micro Edition (Java ME) está orientada hacia el desarrollo de aplicaciones para dispositivos inteligentes que utilizan el sistema operativo Android de Google que se utiliza en muchos teléfonos inteligentes, tabletas (pequeñas computadoras ligeras y móviles con pantallas táctiles).



*Figura 6 Logotipo Tecnología java (www.java.com/es/about/)*

- El 97% de los escritorios empresariales ejecutan Java.
- El 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java.
- 9 millones de desarrolladores de Java en todo el mundo.
- La primera opción para los desarrolladores.
- La primera plataforma de desarrollo.
- 3 mil millones de teléfonos móviles ejecutan Java.
- El 100% de los reproductores de Blu-ray incluyen Java Logo.
- 5 mil millones de Java Cards en uso.
- 125 millones de dispositivos de televisión ejecutan Java.

## **5.9. UML**

UML (Unified Modeling Lenguaje) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos. Capturar decisiones y conocimiento sobre sistemas que deben ser construidos. Se usa para comprender, diseñar, ojear, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para ser utilizado con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje

de modelado pretende unificar la experiencia pasada sobre las técnicas de modelado e incorporar las mejores prácticas de software actuales en una aproximación estándar. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo interactivo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientado a objetos exigentes (Rumbaugh, Jacobson & Booch,2007).

El objetivo principal cuando se empezó a gestar UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación semántica común.

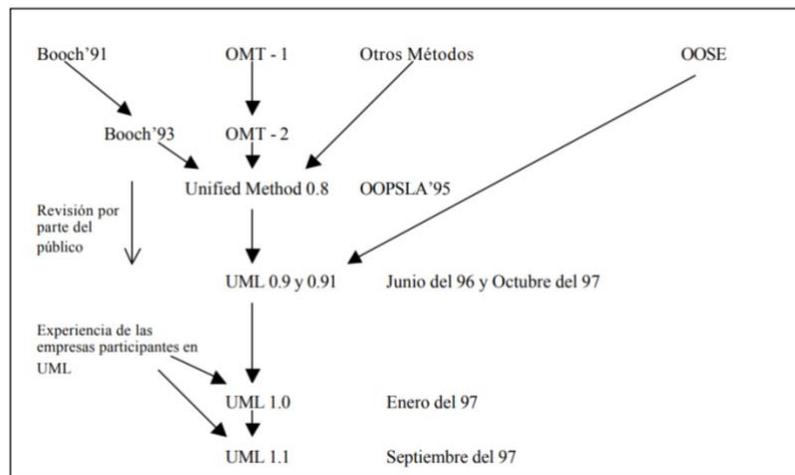


Figura 7 Historia de UML (Booch, OMT, OOSE.1993).

Esta notación ha sido ampliamente aceptada debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT Y OOSE. UML puso fin a la llamada “Guerra de métodos” que se mantuvo a lo largo de los años 90, Con UML se fusionan la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros de software que trabajaban en el desarrollo orientado a objetos.

## **5.10. Diagramas de Caso de Uso**

Un diagrama de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Un caso de uso es una unidad de funcionalidad expresada como una transacción entre los actores y el sistema. El propósito de la vista de casos de uso es enumerar los actores y casos de uso, y mostrar que actores participan en cada caso de uso. El comportamiento de los casos de uso se expresa mediante las vistas dinámicas, especialmente la vista de interacción (Rumbaugh, Jacobson, Booch,2007).

Según el autor (Larman, 2003) “ un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces de software por ejemplo, las interfaces de java en un aplicación.

## **5.11. Diagrama de Secuencia.**

Este tipo de diagrama muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo, en el cual se indicarán los módulos o clases que formaran parte del programa y las llamadas que realizan cada uno de ellos por esta razón permite observar las interacciones de manera cronológica. Es importante recordar que el diagrama de secuencia se realiza a partir de la descripción de un caso de uso.

## **5.12. Diagrama de Clases.**

Los diagramas de clases son uno de los tipos de diagrama más utilizados debido a que permiten trazar claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.

## **VI. Metodología Técnicas de Investigación**

Se aplicará el método de investigación experimental, documental y de campo para el diseño del sistema de información, ya que se van a realizar recopilaciones de datos ya existentes que se emplean en las diferentes áreas de la empresa.

El motivo de usar la investigación documental es debido a que vamos a utilizar una técnica de investigación en la que se seleccionaran y analizaran documentos que contienen datos de interés relacionado con este estudio. A si mismo pondremos en práctica todos los conocimientos teóricos fundamentales para poder diseñar nuestro sistema de información de manera correcta.

Por su parte La metodología de campo es la manipulación de valores externos y no comprobados en situaciones establecidas y controladas de este modo se quiere saber porque se presentan ciertas situaciones en particular y cómo actuar en cada una de ellas.

Otra forma de definirlo sería diciendo que es el proceso que haciendo uso del método científico permite obtener nuevos conocimientos basados en la vida real. Con el fin de obtener datos manejables y posteriormente poder aplicarlos en resolver ciertas problemáticas.

Este tipo de investigación también se conoce como “Investigación In Situ” ya que se realiza con el propio lugar donde se encuentra el objeto de estudio o la problemática que se quiere resolver, esto permite que el conocimiento no solo se limite al investigador, sino que además se pueden manejar datos con más seguridad y se podrán realizar diseños exploratorios, descriptivos y experimentales, teniendo más control sobre la manipulación de variables dependientes.

Cuando los datos se obtienen de situaciones o eventos reales se les denominan primarios, su valor radica en que permiten cerciorarse de las verdaderas condiciones en que se han obtenido los datos, por lo que cuando necesitan ser revisados, consultados y/o modificados resulta ser más fácil.

Cabe aclarar que no toda información se puede obtener siguiendo esta metodología ya que, por circunstancias especiales, de tiempo o por problemas de escasez de información podrían limitar nuestra investigación.

Una vez diseñado y estructurado el sistema se va a implementar la metodología de cascada. Esta consiste en que el inicio de cada etapa es condicionado a que finalice la anterior para poder continuar con el desarrollo.

El modelo cascada, o también llamado ciclo de vida clásico surge un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue. (Pressman,2010).

## **VII. Presentación y discusión de resultados**

En este capítulo se presentan y explican todos los procedimientos que se llevaran a cabo para el diseño y desarrollo del sistema.

### **7.1 Análisis de requerimientos.**

Para este análisis se debe realizar una investigación previa para poder identificar los requerimientos del sistema de información, además podemos realizar entrevistas directas con el encargado de Ds Wraps, de esta manera podemos conocer las necesidades principales para diseñar una base de datos que permita almacenar la información de manera óptima.

Debemos diseñar una base datos partiendo del análisis de datos duros que permita almacenar los registros del sistema, para el control y gestión de la información obtenida, de esta manera podemos implementar las interfaces graficas con el fin de que el usuario final pueda adaptarse de manera rápida y sencilla para conseguir un ritmo de trabajo armónico entre el sistema y el usuario garantizando el uso y manejo información.

Debemos realizar pruebas para asegurar que el diseño, implementación, control y gestión de información funcionen correctamente y a su vez identificar posibles fallas o mejoras que ayuden al usuario a tener mejor control del sistema.

Existirá solo 1 tipo de usuario el administrador y podrá: realizar altas de nuevos usuarios, nuevos productos o proveedores, modificar datos ya almacenados o poder eliminar productos, proveedores, usuarios o datos ya almacenados realizar búsquedas. El sistema deberá de contar con métodos de seguridad y encapsulamiento de datos para garantizar la seguridad del usuario al momento de usar el sistema.

Las ventajas que ofrecer desarrollar un sistema son muchas ya que realmente no se necesitan requerimientos especiales para poder hacer uso del sistema.

A continuación, se mencionan las ventajas:

- Mayor control sobre la administración de la empresa y sobre sus departamentos.
- Los usuarios pueden acceder al sistema de manera fácil y rápida.
- Mayor accesibilidad debido a que cada usuario podrá acceder al sistema desde la computadora en la que se encuentra el sistema.
- Acceso a información de manera inmediata dado que toda la información se almacena de manera local.

La movilidad es una de las principales ventajas que tenemos al trabajar con el sistema ya que podremos acceder al sistema en cualquier momento. La flexibilidad en sentido de que un usuario realice cambios en los datos de un pedido o realice una venta y esta se actualice.

Desarrollar el sistema en java nos brinda la posibilidad de poder tener acceso a él desde cualquier ordenador que tenga a java instalado, muchos de los softwares orientados a desarrollo son "OpenSource", esto quiere decir que no tenemos que pagar o rentar una licencia para poder hacer uso de ellos, al utilizar herramientas OpenSource se reducen costos lo que resulta muy atractivo para el usuario.

Así como tenemos ventajas al desarrollar este tipo de sistemas también tenemos algunas desventajas las cuales son:

- Seguridad escasa sobre todo si no se manejan métodos de encriptación de archivos.

- Lentitud si presenta mucha concurrencia y no se hayan utilizado métodos para evitar esta situación.
- Se requiere personal especializado para realizar tareas de mantenimiento.
- Vulnerabilidad en Bases de Datos sino se manejan protocolos de seguridad.

La seguridad es un tema muy importante por tratar ya que es de vital importancia contar con personal que se encargue de que la información viaje segura a través de la red y que los scripts desarrollados cumplan sus funciones de forma segura y no comprometan la información de la empresa.

Cada vez que se realiza una solicitud (sea el proceso que sea), desde un cliente hacia un servidor esta tiene que viajar por toda la red para darle una respuesta al cliente, sí esto sucede sin problemas el cliente recibe respuesta y puede continuar con sus tarea, pero en caso de que el tiempo establecido para ir y regresar se agota se produce un Time Out y por consiguiente genera un error en la petición del servicio dado que el servidor se encuentra recibiendo muchas peticiones y no puede atender a todas, en este caso el sistema se vuelve lento. Para solucionar los problemas que vayan surgiendo durante el uso del sistema se requiere de personal especializado. No solo para poder dar solución a los problemas presentados sino además de hacerlo de una manera rápida y eficiente.

El equipo se debe dividir en:

- Encargados de Seguridad.
- Personal de Mantenimiento.
- Administrador de Base de Datos.
- Administrador de Red.

Para evitar vulnerabilidades de seguridad en el sistema debemos evitar tener el sistema conectado directo a la base de datos, se debe realizar un filtro de seguridad para autenticar la identidad del usuario que desea establecer una conexión a la base de datos.

también debemos asegurar la integridad de los datos ya que estos viajan libres por la red local y podrían ser interceptados por un tercero, estableciendo capas de transmisión de datos entre las conexiones del sistema y los datos agregamos un nivel de seguridad consiguiendo de esta manera proteger la integridad del sistema y de la base de datos.

## **7.2. Software por utilizar**

Tomando en cuenta los requerimientos funcionales mencionados anteriormente llegamos a la conclusión de utilizar las siguientes herramientas para el desarrollo y funcionamiento óptimo del sistema.

- Lenguaje Java.
- Conector Java: jdk-8u20-nb-8.0.1
- SQL.
- Gestor de base de datos MySQL.
- Netbeans 8.0

## **7.3. Procedimiento**

Utilizaremos el MVC (Modelo Vista Controlador) para el diseño del sistema, lo que nos permitirá desarrollar todo el sistema en capas independientes unas de otras esto nos ayudará a mantener un nivel de complejidad bajo por si en un futuro se requiere realizar alguna actualización en virtud de mejorar el funcionamiento del sistema.

- Modelo: Representa los datos que el usuario está esperando ver, gestiona todo los accesos o manipulación de información.
- Vista: hace referencia a una representación gráfica del modelo para que pueda ser visualizada por el usuario ya sea un archivo de texto o una página web. El propósito de la vista es convertir los datos para que el usuario los pueda interpretar fácilmente.
- Controlador: Es la parte lógica responsable del procesamiento de las peticiones (request) del usuario, construyendo un modelo apropiado se pasa a la vista para su visualización.

A continuación, se muestra una representación gráfica del funcionamiento de un MVC y su interacción con el usuario.

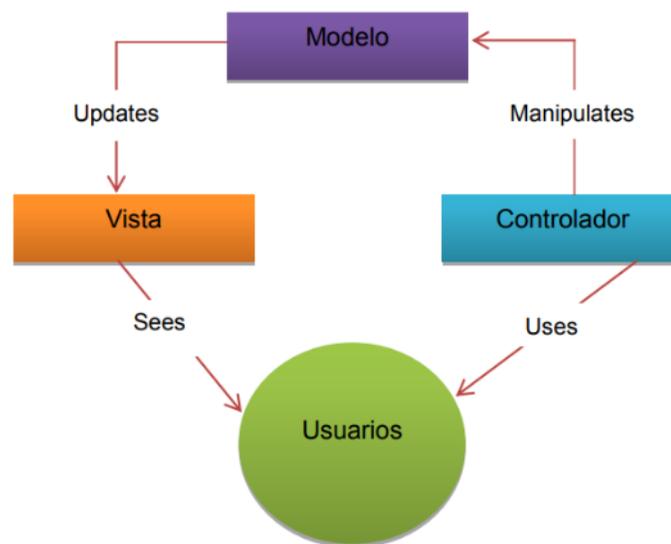
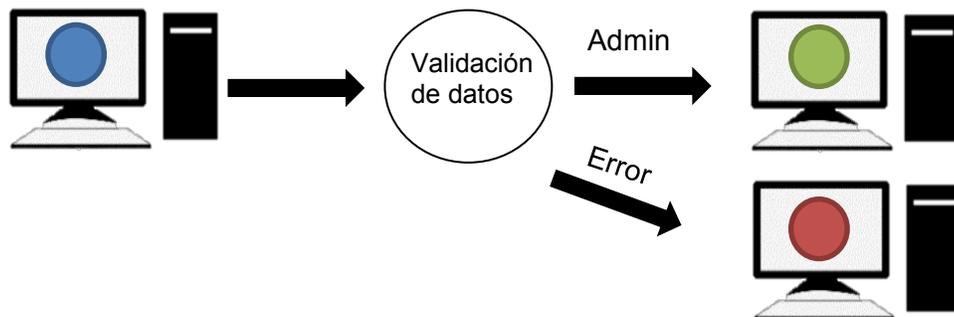


Figura 8 Modelo vista controlador (<https://wordpress/modelo-vista-controlador-mvc.pdf>)

### 7.3.1. Arquitectura y Diseño del sistema

Para acceder a los datos que contiene el sistema será a través del usuario registrado, el sistema identifica el usuario y muestra la información requerida. En la siguiente imagen se describe gráficamente este proceso de interacción del sistema con el usuario y su autenticación.



*Figura 9 Representación validación exitosa y fallida (Fuente: Propia, 2022)*

En este diagrama representaremos de manera gráfica las funciones que tiene el usuario que tendrán acceso al sistema y a las áreas respectivas a sus funciones.

El administrador tendrá acceso total al sistema, así como a las diferentes funciones como lo son alta, modificación, actualización de registro o si lo prefiere podrá eliminar datos ya ingresados, podrá llevar a cabo tareas de mantenimiento.

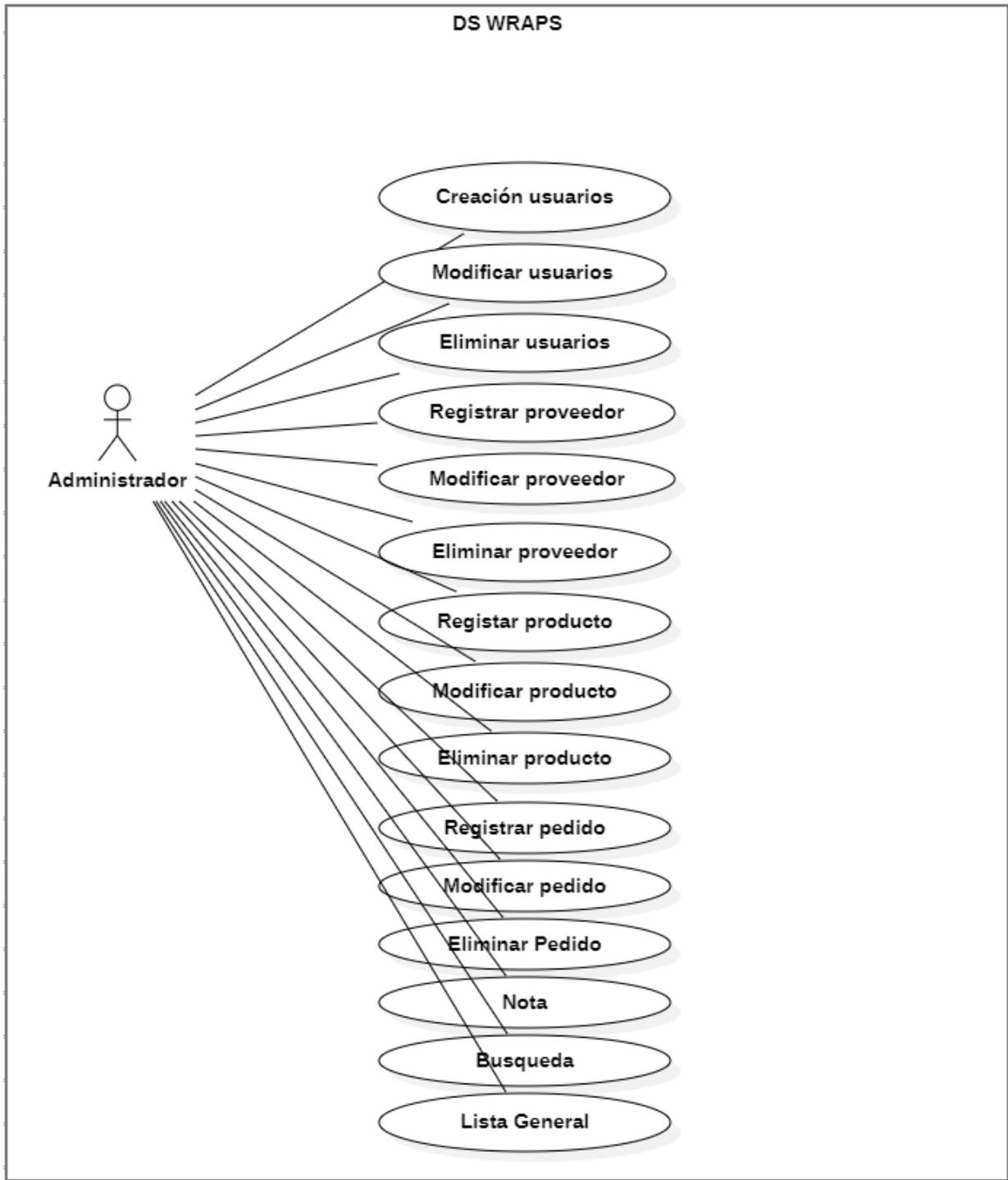


Figura 10 Diagrama de casos de uso (Figura: Propia, 2022).

### **7.4.1. Modelo de la Base de datos.**

La base de datos es el elemento principal por tomar en cuenta al desarrollar el sistema debido a que será la responsable de toda la parte operativa relacionada con el uso y manejo de la información. Antes de crear la base de datos debemos realizar un fuerte análisis en virtud de evitar posibles errores de diseño o relaciones que deriven en problemas futuros como lo son redundancias en datos, consultas lentas, pérdida de información o consumo excesivo de recursos destinados a su correcto funcionamiento. Se utilizará el Modelo Relacional para representar de manera grafica la estructura de la base de dato y poder tener una vista general.

### **7.4.2. Modelo Relacional.**

Es un modelo de datos basado en la lógica de predicados y en teoría de conjuntos. Tras ser postuladas sus bases en 1970 por Edgard Frank Codd, de los laboratorios IBM, se popularizo como nuevo paradigma en los modelos de bases de datos. Desarrolla un esquema de base de datos a partir del cual se podrá realizar el modelado físico o de implementación de DBMS.

Su idea fundamental es el uso de relaciones. La mayoría de las veces se conceptualiza de manera más fácil el imaginar las relaciones como si fueran tablas que están compuestas por registros de columnas y filas, siendo cada fila o columna un registro o tupla, en la actualidad es el modelo más utilizado para modelar problemas reales y administrar datos dinámicamente.

Este modelo está basado en que todos los datos están almacenados en tablas (Entidad/Relaciones) y cada una de estas es un conjunto de datos, por lo tanto, una base de datos es un conjunto de relaciones. La agrupación se origina en la tabla: Tabla > Fila (Tupla) > Campo (Atributo).

El modelo relacional se ocupa de:

- La estructura de los datos.
- La manipulación de los datos.
- La integridad de los datos.

Donde las relaciones están formadas por: Atributos (columnas) y Tuplas (Conjunto de filas). De acuerdo con las tablas que existen en la base de datos para poder hacer uso de la información que manejan dichas tablas debemos crear clases DTO (Data Transfer Object).

Según el autor (Blancarte, 2018) menciona que, “El patron DTO tiene como finalidad el crear un objeto plano POJO con una serie de atributos que puedan ser enviados o recuperados del servidor en una sola petición, de tal manera que un DTO puede contener información de múltiples fuentes o tablas y concentrarlas en una única clase simple.

A continuación, mostraremos una representación gráfica de cómo se compone nuestra base de datos, mostrando las tablas que la conforman y las relaciones que existen entre ellas.

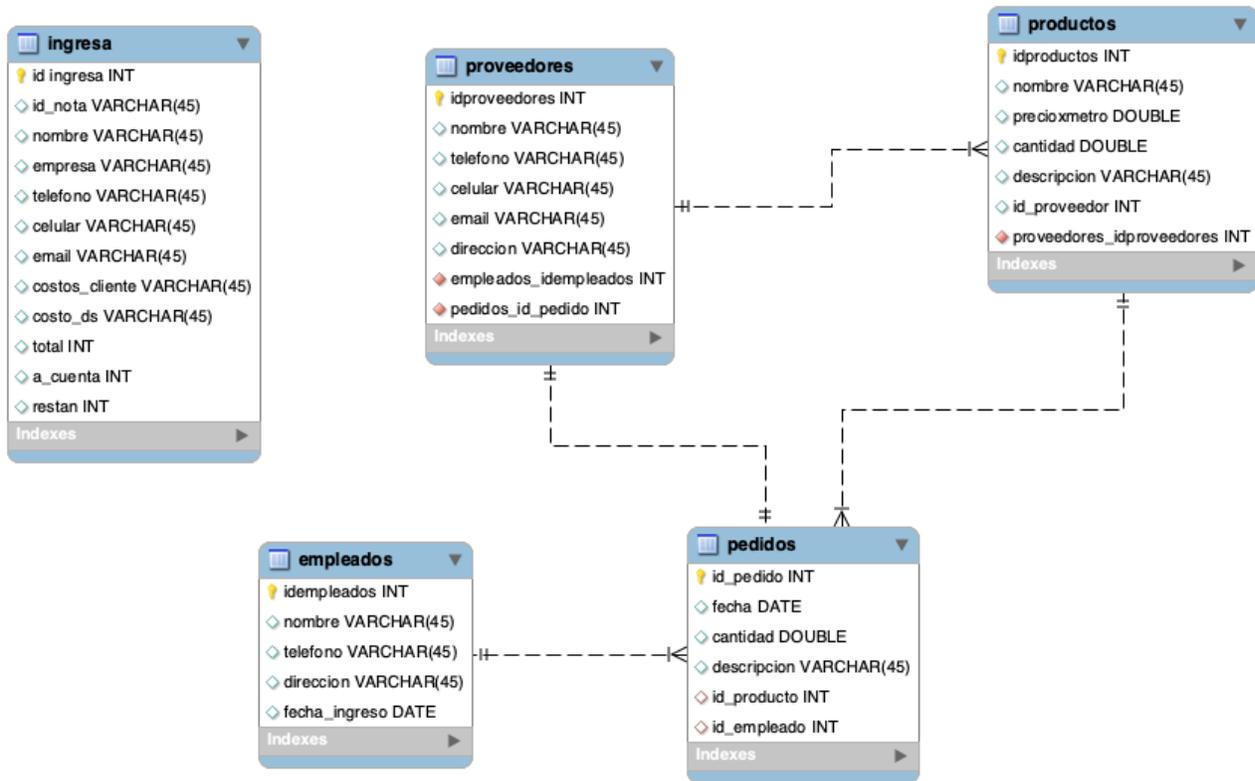


Figura 11 Modelo de base de datos del sistema Ds Wraps (Fuente: Propia,2022).

En la figura anterior podemos ver de manera grafica la composición de la base de datos y las relaciones que existen entre las diferentes tablas, podemos apreciar la relación que comparte la tabla 'proveedores' con la tabla 'productos', donde la relación es de uno a muchos debido a que un proveedor puede contener varios productos al igual que la tabla 'productos' tiene una relación uno a muchos con la tabla 'pedidos' debido a que un mismo producto puede estar en muchos pedidos por clientes.

A continuación, presentaremos cada tabla de manera individual, con la intención de explicar a detalle el porqué de cada una, además explicaremos los métodos Getters y Setters que son de suma importancia en la etapa de diseño y desarrollo del sistema ya que estos métodos son los encargados de obtener y definir los datos según sea el caso cuando el sistema los utilice.

### 7.4.2.1. Tabla Ingresa.

La tabla ingresa sirve para crear nuevas notas de servicio, para registrar las ventas hechas a los clientes o para ingresar nueva información al sistema tal como se muestra en la figura 12.

| Ingresa DTO   |
|---|
| +id_nota: integer<br>+nombre: string<br>+empresa: string<br>+telefono: double<br>+celular: double<br>+email: string<br>+costos_cliente: string<br>+costos_ds: string<br>+total: integer<br>+acuenta: integer<br>+restan: integer  |
| +set_id_nota(id_nota: integer)<br>+get_idnota(): integer<br>+set_nombre(nombre: string)<br>+get_nombre(): string<br>+set_telefono(telefono: double)<br>+get_telefono(): double<br>+set_celular(celular: double)<br>+get_celular(): double<br>+set_email(email: string)<br>+get_email(): string<br>+set_costocliente(costocliente: string)<br>+get_costocliente(): string<br>+set_costods(costods: string)<br>+get_costods(): string<br>+set_total(total: integer)<br>+get_total(): integer<br>+set_acuenta(acuenta: integer)<br>+get_acuenta(): integer<br>+set_restan(restan: integer)<br>+get_restan(): integer |

Figura 12 Ingresa DTO (Fuente: Propia,2022).

### 7.4.2.2. Tabla proveedores.

La tabla proveedores sirve para almacenar la información de cada proveedor que trabaja en conjunto con Ds Wraps, se almacenan los datos de contacto principales que serán utilizados a la hora de requerir los servicios o materias que brinda cada uno como se muestra en la figura 13.

| Proveedores DTO   |
|---|
| +id_proveedor: integer<br>+nombre: string<br>+telefono: double<br>+celular: double<br>+email: string<br>+direccion: string  |
| +set_id_proveedor(id_proveedor: integer)<br>+get_id_proveedor(): integer<br>+set_nombre(nombre: string)<br>+get_nombre(): string<br>+set_telefono(telefono: double)<br>+get_telefono(): double<br>+set_celular(celular: double)<br>+get_celular(): double<br>+set_email(email: string)<br>+get_email(): string<br>+set_direccion(direccion: string)<br>+get_direccion(): string |

Figura 13 Proveedores DTO (Fuente: Propia, 2022).

### 7.4.2.3. Tabla empleados.

En esta tabla se ingresará la información personal de cada empleado que trabaja para Ds Wraps datos básicos de cada uno de ellos como lo son nombre, teléfono su dirección de contacto en caso de alguna emergencia o algún asunto extraordinario y por último la fecha en la que ingreso a laborar en la empresa

Esto los podemos apreciar en la siguiente figura número 14.

| Empleados DTO  |
|--|
| +id_empleado: integer<br>+nombre: string<br>+telefono: double<br>+direccion: string<br>+fecha_ingreso: date  |
| +set_id_empleado(id_empleado: integer)<br>+get_id_empleador(): integer<br>+set_nombre(nombre: string)<br>+get_nombre(): string<br>+set_telefono(telefono: double)<br>+get_telefono(): double<br>+set_direccion(direccion: string)<br>+get_direccion(): string<br>+set_fecha_ingreso(fecha_ingreso: date)<br>+get_fecha_ingreso(): date |

Figura 14 Empleados DTO (Fuente: Propia,2022).

#### 7.4.2.4. Tabla productos.

En esta tabla se muestra los datos requeridos para registrar un nuevo producto que se ofrecerá en las instalaciones de Ds Wraps se agregaron con datos sencillos para poder identificarlos de manera más fácil y rápida estos datos ayudaran al personal cuando se requiera realizar un inventario de los productos existentes. Se puede apreciar en la figura 15.

| Productos DTO  |
|--|
| +id_producto: integer<br>+nombre: string<br>+precioxmetro: double<br>+cantidad: double<br>+descripcion: string   |
| +set_id_producto(id_producto: integer)<br>+get_id_producto(): integer<br>+set_nombre(nombre: string)<br>+get_nombre(): string<br>+set_precioxmetro(precioxmetro: double)<br>+get_precioxmetro(): double<br>+set_cantidad(cantidad: double)<br>+get_cantidad(): double<br>+set_descripcion(descripcion: string)<br>+get_descripcion(): string |

Figura 15 Productos DTO (Fuente: Propia, 2022).

#### 7.4.2.5. Tabla pedidos.

La tabla pedidos es de suma importancia ya que es donde se almacenarán todas las ordenes de trabajo nuevas y pasadas además aquí se podrán consultar los pedidos pendientes por entregar u ordenes próximas a trabajar. Estos lo podemos apreciar en la figura 16.

| Pedidos DTO   |
|---|
| +id_pedido: integer<br>+fecha: date<br>+cantidad: double<br>+descripcion: string  |
| +set_id_pedido(id_pedido: integer)<br>+get_id_pedido(): integer<br>+set_fecha(fecha: date)<br>+get_fecha(): date<br>+set_cantidad(cantidad: double)<br>+get_cantidad(): double<br>+set_descripcion(descripcion: string)<br>+get_descripcion(): string |

Figura 16 Pedidos DTO (Fuente: Propia, 2022).

### 7.4.3. Data Access Object (DAO).

Una de las grandes problemáticas al momento de acceder a los datos es que la implementación y el formato de la información puede variar dependiendo de la fuente. Si tenemos múltiples fuentes estas pueden cambiar y tendríamos que implementar diferentes lógicas para acceder a la información ya que podrían provenir de bases de datos relacionales, Archivos XML, Textos planos o Web services.

Cabe aclarar que con lo antes mencionado es necesario hacer mención a que se refiere cuando se habla de un Objeto de acceso de datos. Su objetivo es separar la lógica del negocio de la lógica para acceder a los datos, es decir el DAO proporciona los métodos necesarios para insertar, actualizar, borrar y consultar la información que contiene el sistema.

Los componentes que conforman un DAO son los siguientes.

- **Business Object:** representa un objeto con la lógica de negocio.
- **Data Access Object:** representa una capa de acceso a datos que oculta la fuente y los detalles técnicos para recuperar los datos.
- **Transfer Object:** este es un objeto plano que implementa el patrón Data Transfer Object (DTO), el cual sirve para transmitir la información entre el DAO y el Business Service.
- **Data Source:** representa de forma abstracta la fuente de datos, la cual puede ser una base de datos, Webservices, LDAP, archivos de texto, etc.

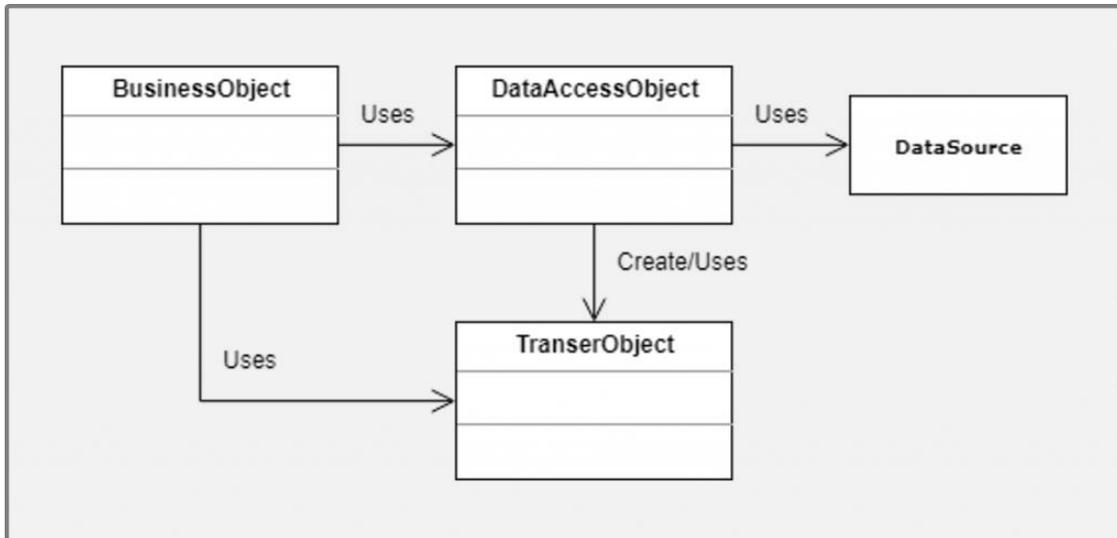


Figura 17 Diagrama Data Access Object. (Fuente: <https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/>, 2019)

## 7.5. Vistas

En este apartado podremos apreciar la vista grafica del sistema administrador, el sistema cuenta con múltiples vistas unas solo disponibles para el administrador y otras disponibles para los usuarios habituales, cada una de las vistas tiene un diseño diferente que brinda un nivel de personalización más adecuado al área a la cual hace referencia. Este proyecto cuenta con un total de 9 vistas diferentes, que incluyen la vista para ingresar al menú de administrador.

### 7.5.1. Vista menú principal de administrador.

En esta vista podemos ver el menú principal del sistema, mediante esta página podremos acceder a todas las áreas del sistema mediante botones personalizados con iconos que hacen referencia al área que pertenece, además contamos con un botón que permite salir del sistema y cierra la sesión del administrador de manera segura evitando de esta manera la posible fuga de

información o pérdida de datos ingresados.



Figura 18 Vista principal del administrador (Fuente: Propia, 2022).

### 7.5.2. Vista Nueva Nota.

En esta vista podremos ingresar nuevas notas de servicio, que se agregaran a la base de datos y podrán ser consultadas en cualquier momento que el administrador lo decida, además esta vista cuenta con una función de validación de campos vacíos para evitar posibles errores en la base de datos y en las consultas futuras, esta función muestra un mensaje de alerta al usuario indicándole que campos están vacíos y pide que se ingrese la información faltante, una vez ingresando los datos de manera correcta informa al usuario que los datos fueron guardados de manera correcta con un mensaje emergente.



Figura 19 Vista nueva nota (Fuente: Propia, 2022).

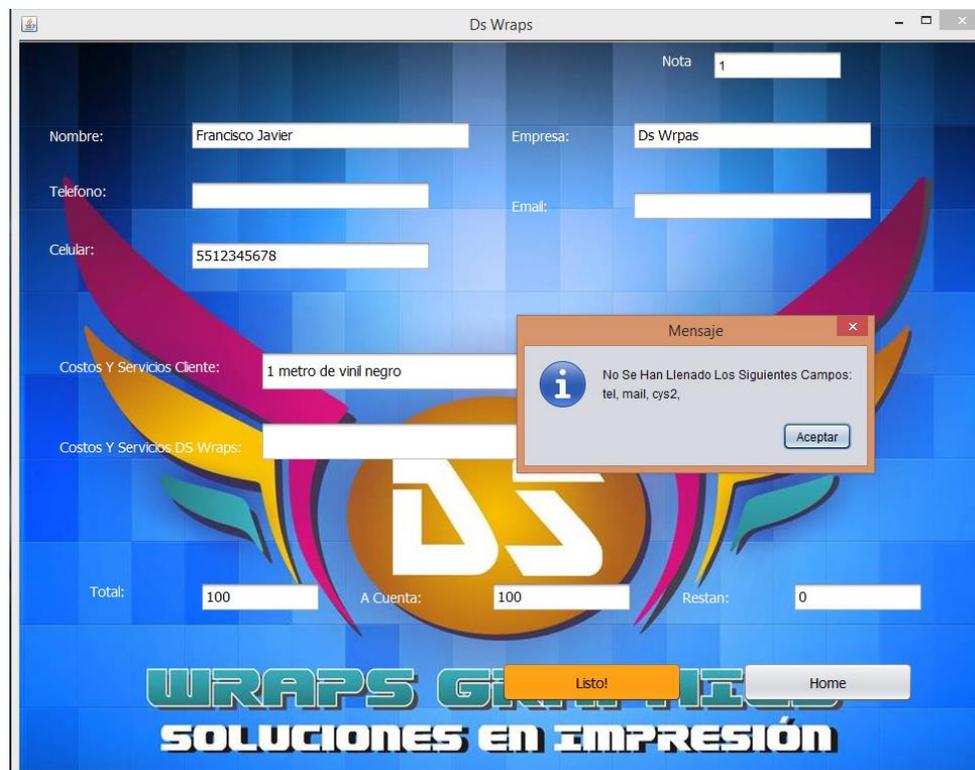
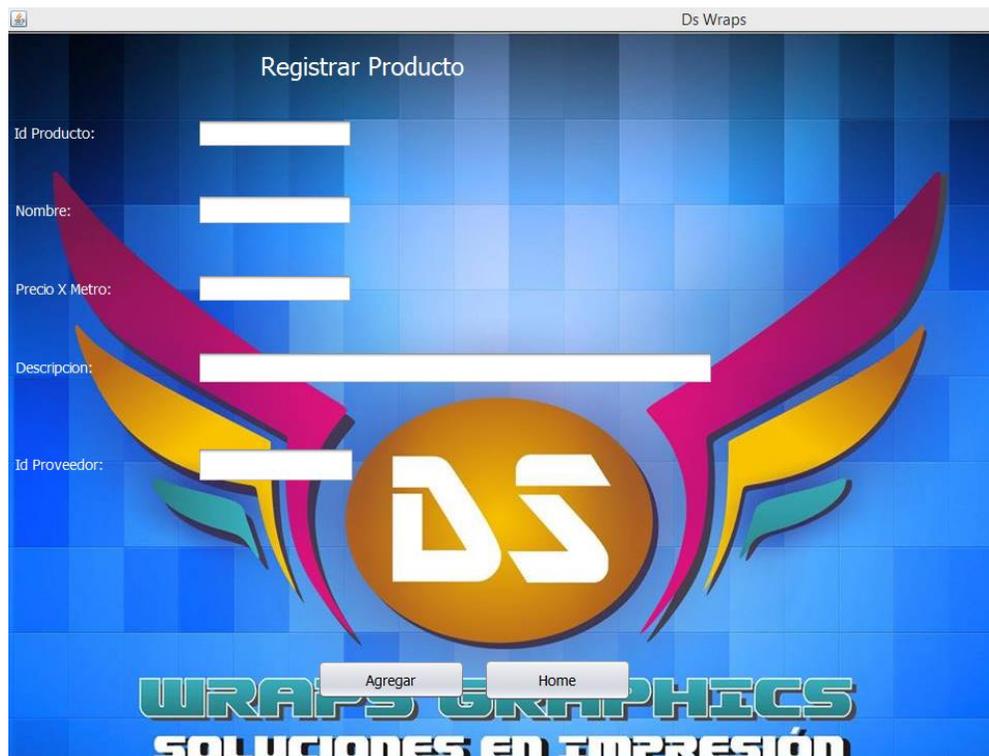


Figura 20 Vista nueva nota validación campos vacíos (Fuente: Propia, 2022)

### 7.5.3. Vista Registrar Producto.

En esta vista el administrador podrá ingresar nuevos productos a la base de datos los cuales serán ofertados en las instalaciones de Ds Wraps, cada producto se registrara con un identificador único para facilitar la búsqueda de ese producto en específico además cada producto estará relacionado con un numero identificador de su proveedor para poder tener un control más específico sobre los productos registrados y en caso de requerir volver a realizar un pedido al proveedor este ya se encuentre relacionado y se pueda realizar de manera más fácil.



The screenshot shows a web browser window titled "Ds Wraps" with a form titled "Registrar Producto". The form contains the following fields and buttons:

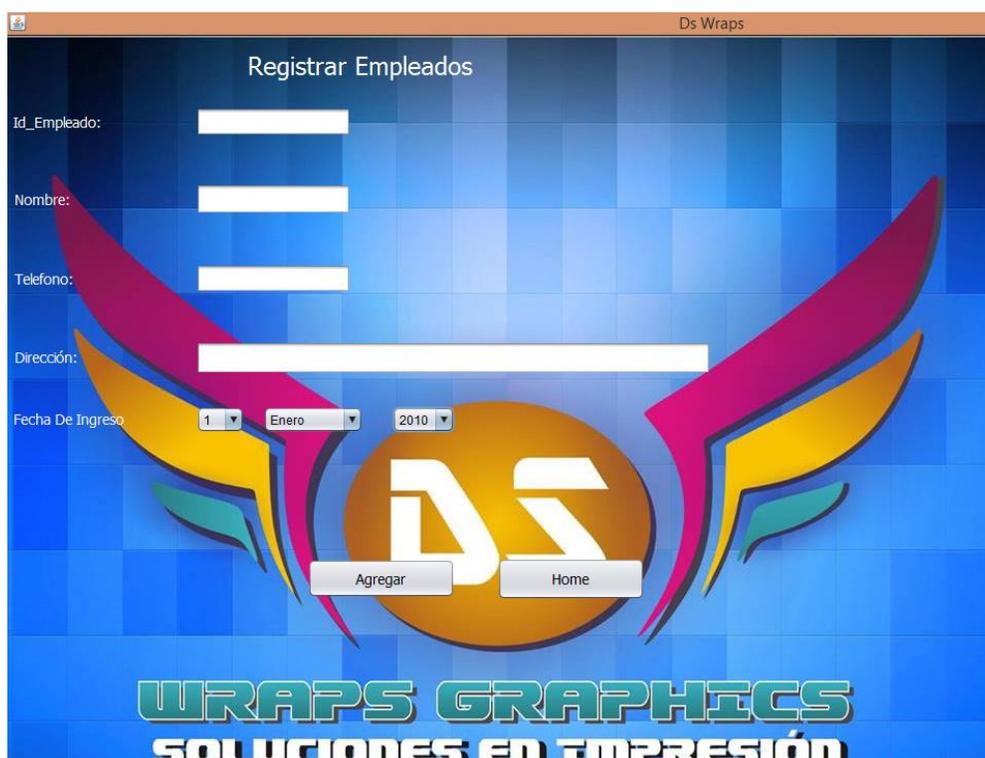
- Id Producto: [input field]
- Nombre: [input field]
- Precio X Metro: [input field]
- Descripcion: [input field]
- Id Proveedor: [input field]
- Buttons: Agregar, Home

The background of the form features a large logo with the letters "DS" in a stylized font, surrounded by colorful wings. Below the logo, the text "WRAPS GRAPHICS" and "SOLUCIONES EN IMPRESIÓN" is visible.

Figura 21 Vista registrar producto (Fuente: Propia, 2022)

#### 7.5.4. Vista Registrar Empleado

En esta vista el administrador podrá realizar el registro de empleados que ingresen a laboral en Ds Wraps. Cada empleado contara con un número de identificación el cual ayudara con el control de personal. Los datos que registraran en el sistema son datos sencillos como el nombre completo, teléfono, dirección o domicilio y la fecha en la que ingreso al laboral en Ds la cual se podrá ingresar a través de menús desplegables.



The screenshot shows a web application window titled "Ds Wraps" with a sub-header "Registrar Empleados". The form contains the following fields and controls:

- Id\_Empleado:** A text input field.
- Nombre:** A text input field.
- Telefono:** A text input field.
- Dirección:** A text input field.
- Fecha De Ingreso:** Three dropdown menus for day, month, and year, currently set to "1", "Enero", and "2010".
- Buttons:** "Agregar" and "Home".

The background features a large logo with the letters "DS" and the text "WRAPS GRAPHICS SOLUCIONES EN IMPRESIÓN".

Figura 22 Vista registrar empleados (Fuente: Propia, 2022)

#### 7.5.5. Vista Registrar Proveedor.

El administrador podrá registrar nuevos proveedores de los diferentes productos que ofrece al público en general cabe mencionar que cada proveedor tendrá un número de identificación el cual estará relacionado con los productos registrados.

Los datos que se requerirán serán: Nombre o Razón social, Número de teléfono fijo, Numero de celular, Email y Dirección de sus instalaciones.

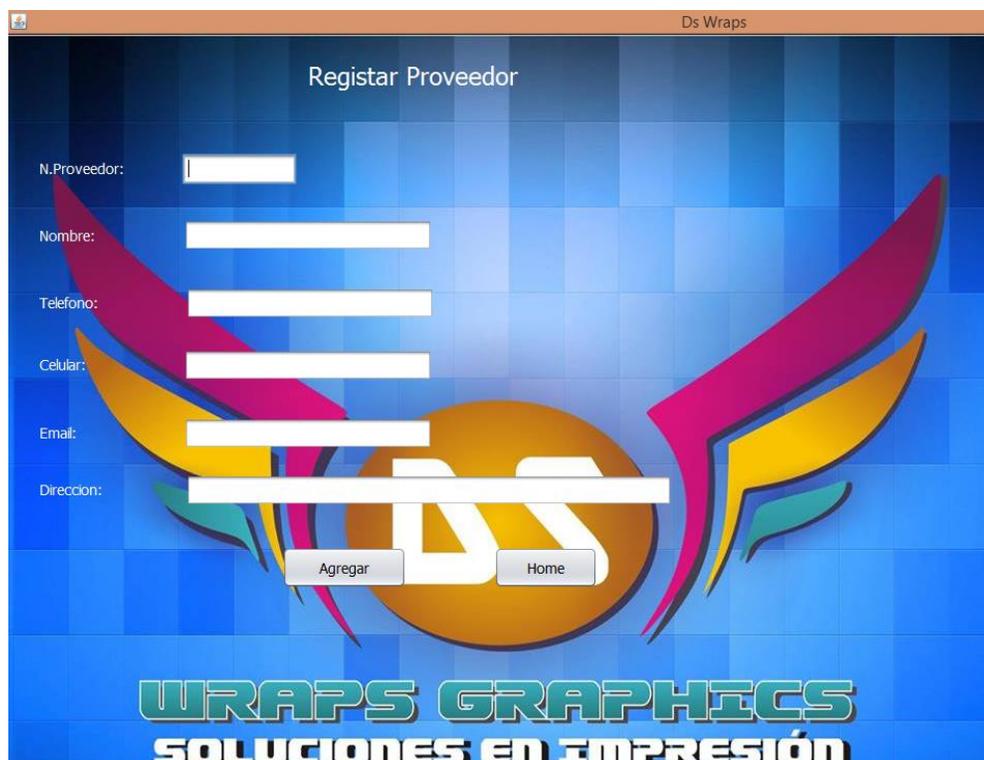
The image shows a software window titled "Ds Wraps" with a form titled "Registrar Proveedor". The form contains six input fields: "N.Proveedor:", "Nombre:", "Telefono:", "Celular:", "Email:", and "Direccion:". Below the fields are two buttons: "Agregar" and "Home". The background of the window features a large, colorful logo for "WRAPS GRAPHICS SOLUCIONES EN IMPRESIÓN".

Figura 23 Vista registrar proveedor (Fuente: Propia, 2022)

### 7.5.6. Vista Registrar Pedido.

En esta ventana el administrador podrá registrar nuevos pedidos o notas de servicio esta es una de las ventanas más importante del sistema ya que será la más utilizada en el día a día, cada nota de servicio o pedio tendrá su número de identificación como lo hemos visto en otras áreas, este número estará relacionado con el producto al que hace referencia todo con el fin de aumentar el control en la administración de los recursos materiales con los que cuenta Ds Wraps, los datos que ingresaremos en esta ventana para registrar un nuevo pedido son los siguientes: identificador del pedido, cantidad, tenemos un campo de descripción en caso de que se requieran instrucciones específicas, la fecha

en la que se realizó el pedido, el identificador del producto y por último el identificador del empleado que lo ingreso.

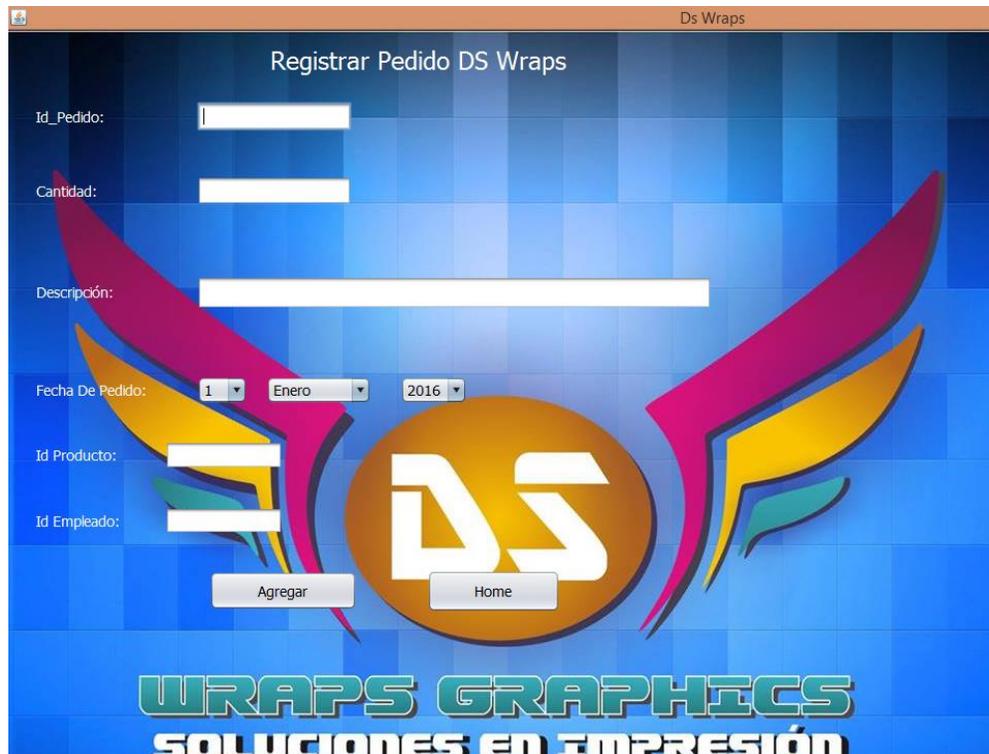


Figura 24 Vista registrar pedido (Fuente: Propia, 2022)

### 7.5.7. Vista Consultar

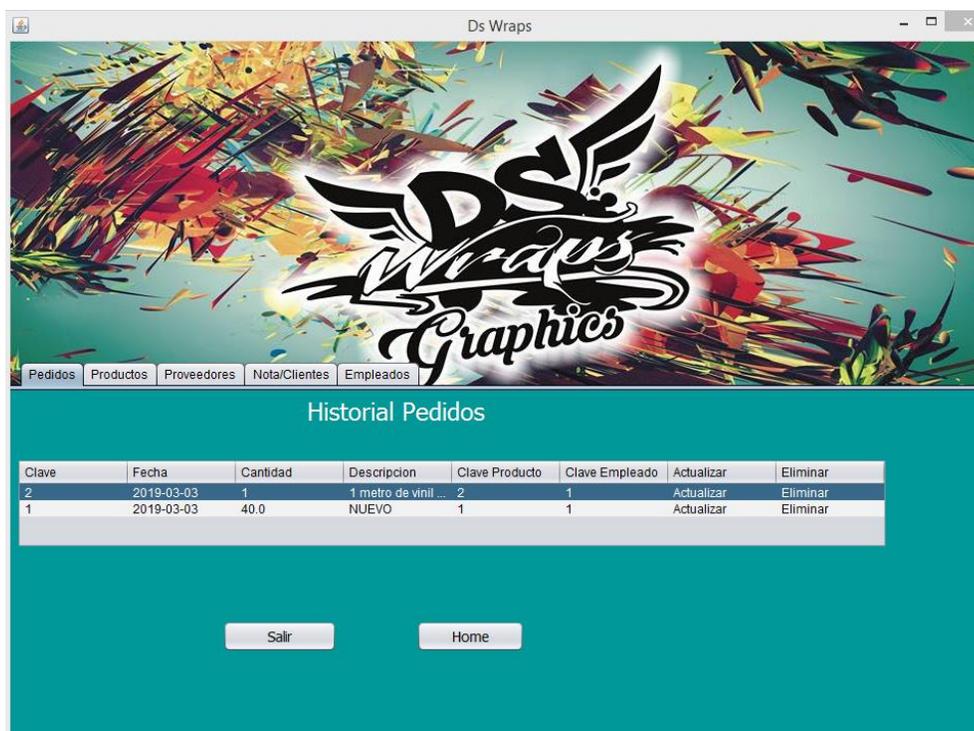
En esta parte del sistema podremos visualizar de manera general todas las operaciones registradas por el administrador, esta ventana nos muestra en formato de lista todo lo que se ha registrado en la base de datos, además aquí podremos encontrar acceso directo a toda la información previamente registrada.

En esta ventana a diferencia de otras ventanas del sistema se cuenta con un botón para poder salir del sistema de manera directa sin tener que regresar al menú principal. El administrador podrá actualizar la información existente en la base de datos o si lo prefiere también tendrá la opción de eliminar los registros

ya existentes, estas dos opciones se implementaron como parte de la tabla que despliega la información que se muestra por cada área, esto a petición del señor Alfredo ya que nos comentó que muchas veces llega a ver errores de registro y nos pidió tener estas opciones disponibles por si llegara a ocuparlas.

Con relación a la opción de eliminar la información debemos tener en cuenta que cuando se desee realizar una operación de este tipo se debe tener en cuenta la relación que puede llegar a tener con otras áreas o si posee una foreign key que pueda llegar a entorpecer el correcto funcionamiento del sistema o de la acción que se quiera realizar.

Con relación a la función actualizar es un poco más permisiva ya que se puede atribuir a una acción de actualización de datos ingresados de manera errónea por lo cual se puede realizar sin mayor problema o posibles errores de funcionamiento del sistema.



| Clave | Fecha      | Cantidad | Descripcion          | Clave Producto | Clave Empleado | Actualizar | Eliminar |
|-------|------------|----------|----------------------|----------------|----------------|------------|----------|
| 2     | 2019-03-03 | 1        | 1 metro de vinil ... | 2              | 1              | Actualizar | Eliminar |
| 1     | 2019-03-03 | 40.0     | NUEVO                | 1              | 1              | Actualizar | Eliminar |

Figura 25 Vista Consultar todo, Pedidos (Fuente: Propia, 2022)

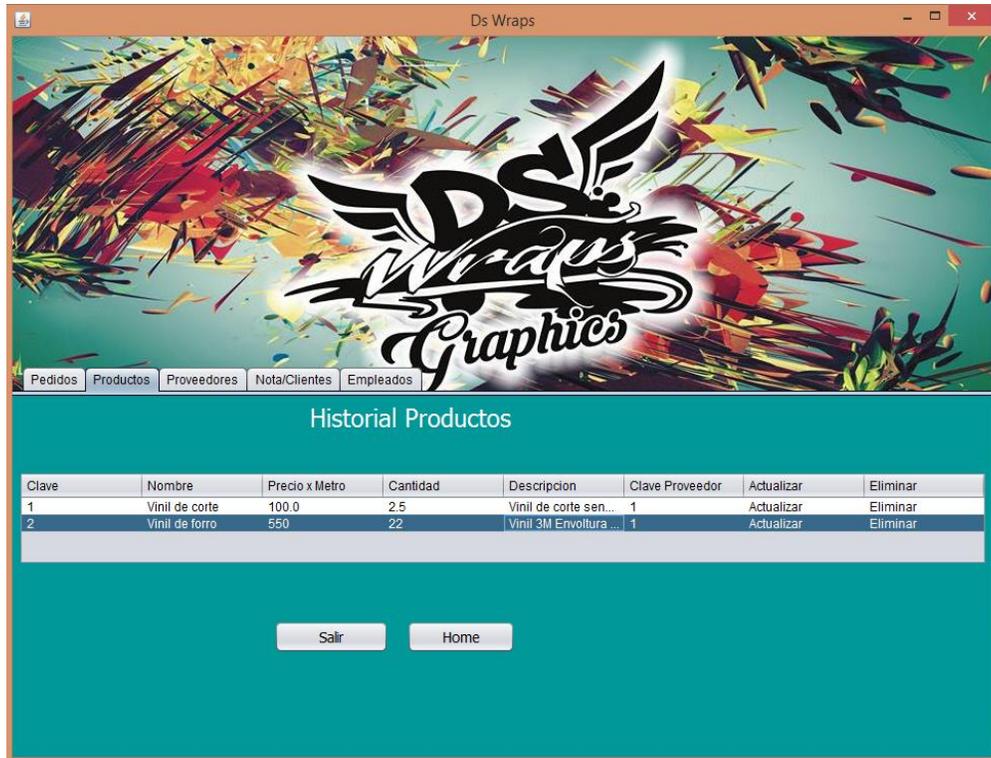


Figura 26 Vista Consultar todo. Productos (Fuente: Propia, 2022)

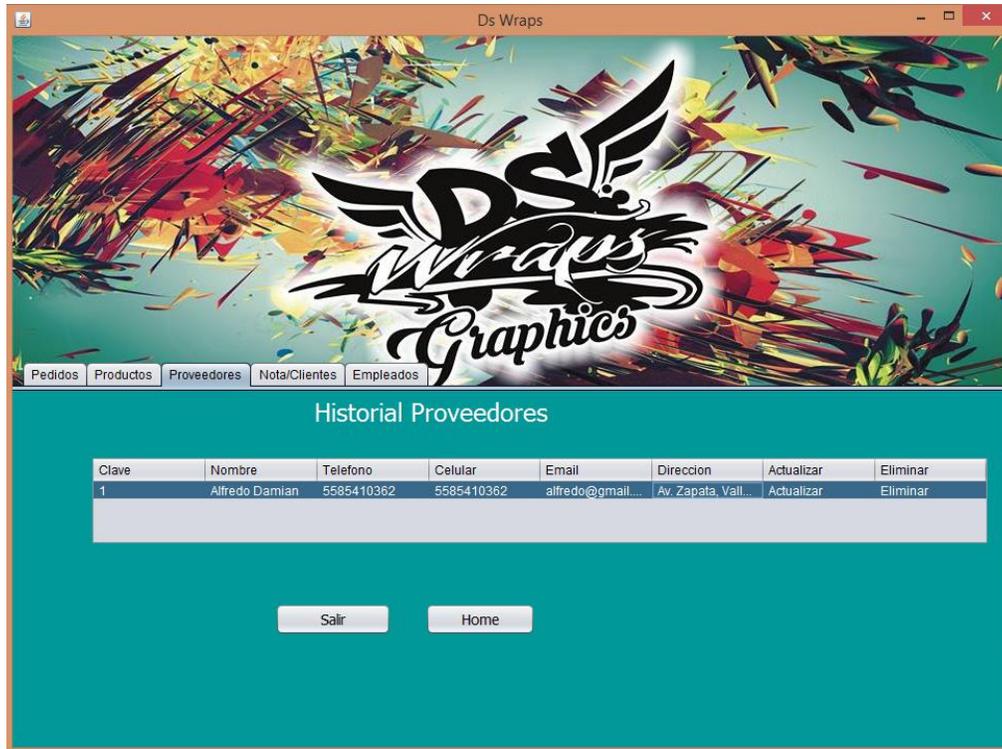


Figura 27 Vista Consultar todo, Proveedores (Fuente: Propia, 2022)



Figura 28 Vista Consultar todo, Nota / Clientes (Fuente: Propia, 2022)

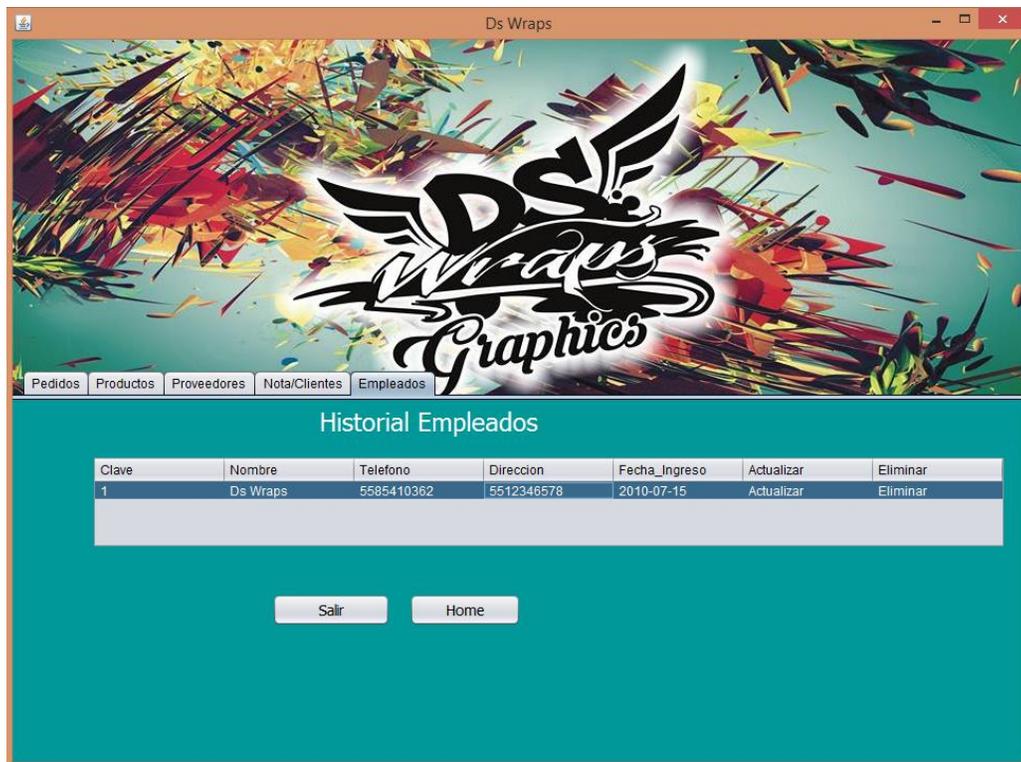


Figura 29 Vista Consultar todo, Empleados (Fuente: Propia, 2022)

### 7.5.8. Vista Búsqueda.

Esta es la última parte del sistema, en esta vista el administrador podrá realizar búsquedas de cualquier área que necesite consultar. Para poder realizar una búsqueda en el apartado de Nota/Clientes el sistema nos pedirá 3 datos los cuales son el número de identificación, nombre del cliente y empresa a la cual pertenece, es muy importante especificar que cuando se requiera realizar una búsqueda siempre se cuente con el número de identificación ya que este es un dato sensible que el sistema requiere para mostrar la información de manera correcta con respecto a los otros dos datos que el sistema requiere se puede realizar la búsqueda ingresando al menos uno de los dos.

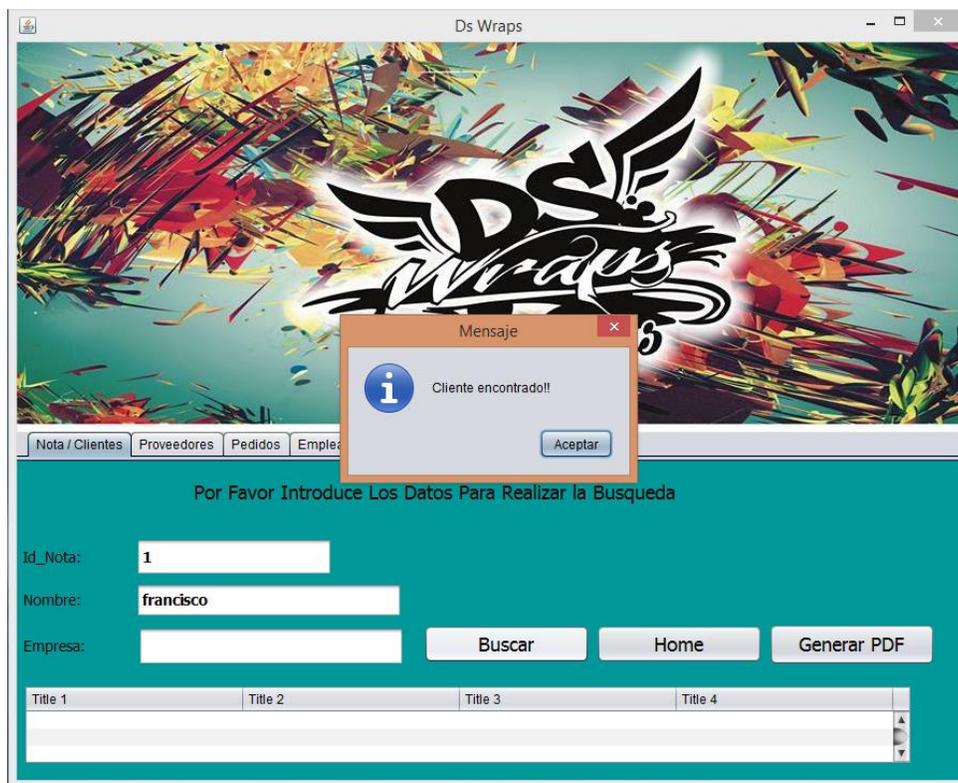


Figura 30 Vista Búsqueda, Nota / Clientes (Fuente: Propia, 2022)

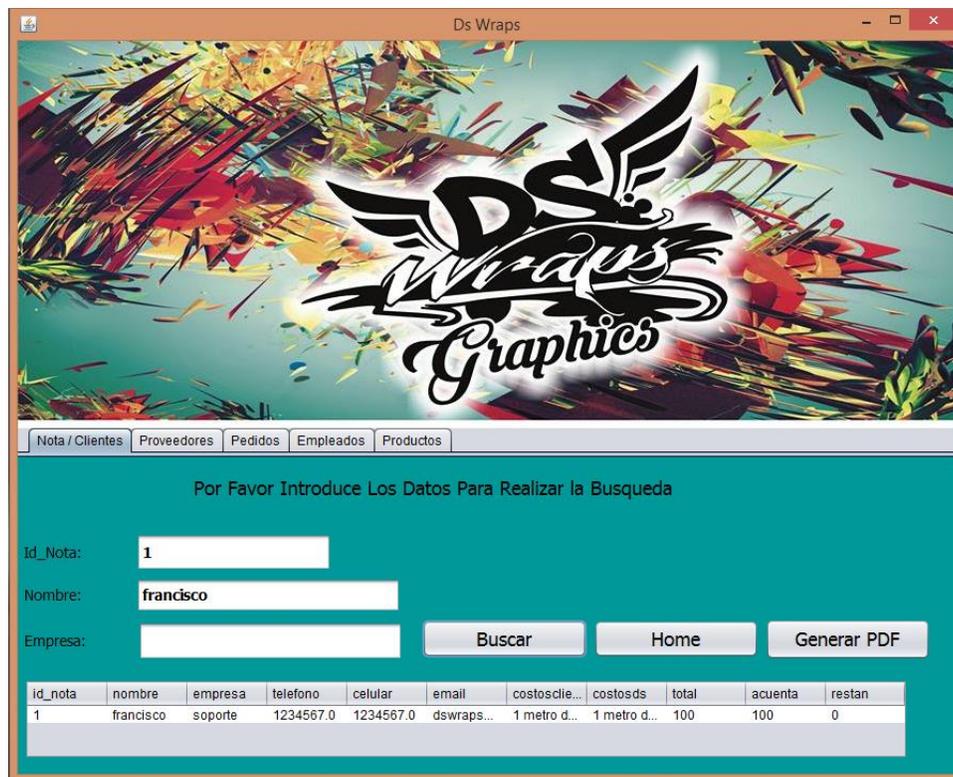


Figura 31 Vista Búsqueda, Nota / Clientes, Datos (Fuente: Propia, 2022)

En esta vista se cuenta con una función adicional que no se encuentra en las demás vistas del sistema, en esta vista tenemos la opción de generar un PDF que mostrara los datos que se consultaron y que se muestran en la tabla, además tendremos la opción de poder imprimir esa información en un formato previamente diseñado solo tenemos que dar clic en el botón “Generar PDF”, para que el sistema nos muestre la información y si el usuario así lo desea podrá imprimirlo para entregarlo al cliente como comprobante de venta o simplemente si así lo requiere.

El reporte se genera con la información consultada por el usuario, el logo de la empresa Ds Wraps, Un encabezado con el nombre de la empresa y en el pie de página podemos encontrar la fecha actual en la que se está generando el reporte. Además, contamos con las opciones generales de un visor de PDF, las cuales son poder imprimir el reporte, guardarlo en una ubicación deseada, navegar entre páginas, funciones de visualización de página y función de zoom.

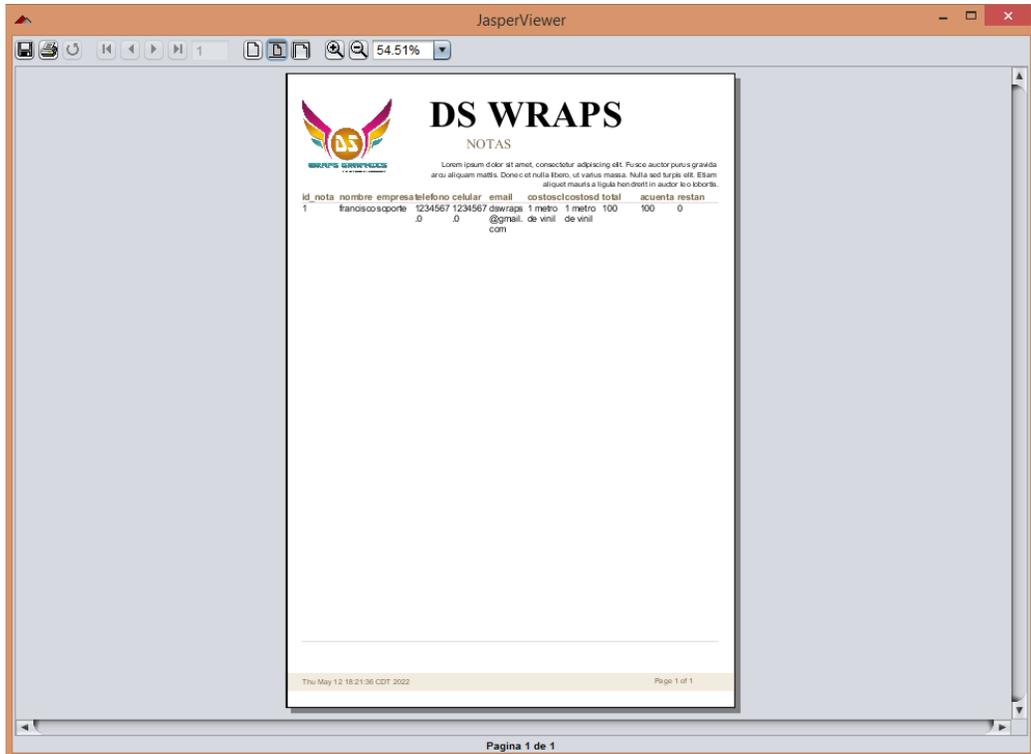


Figura 32 Vista Búsqueda, Nota / Clientes, Generar PDF (Fuente: Propia, 2022)



Figura 33 Vista Búsqueda, Nota / Clientes, Generar PDF, Guardar como (Fuente: Propia, 2022)

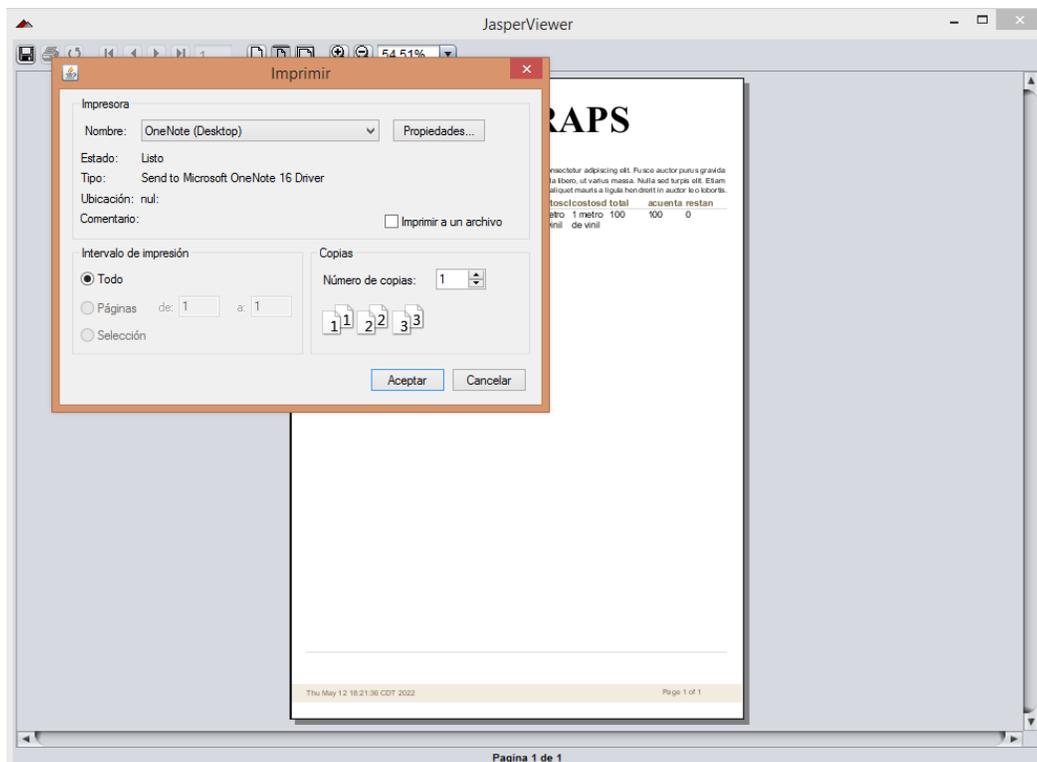


Figura 34 Vista Búsqueda, Nota / Clientes, Generar PDF, Imprimir (Fuente: Propia, 2022)

## 7.6. Controladores

Los controladores son métodos encargados de realizar las peticiones del usuario que son ingresadas desde las diferentes vistas del sistema, estas peticiones pueden ser para registrar nuevos datos, modificarlos, eliminar información ya existente en la base de datos o en listar información que se requiera consultar.

La función de cada controlador es muy importante dado que de estos depende las funciones del sistema, por ejemplo, si el usuario quisiera guardar un registro el botón guardar servirá específicamente para guardar los datos ingresados en la base de datos y tabla especificada en el código de funcionamiento. No se podrán realizar funciones de manera simultánea solo de manera secuencial esto quiere decir que si queremos guardar información actualizar o eliminar se tendrá que realizar de manera secuencial y esperar que una función termine antes de iniciar otra.

## **7.7. Instalación de software principal.**

Para la realización de este sistema se descargó el entorno de desarrollo integrado NetBeans desde la página oficial del desarrollador en su versión 8.1 completa para uso en una arquitectura de 64 bits, cabe mencionar que es un software de licencia libre y gratuito. Al momento de su instalación el software solicitaba la instalación de JDK 8 (Java Development Kit) que es un software desarrollado por java. Con esta herramienta podremos compilar y ejecutar nuestro código escrito en java.

El siguiente software será MySQL, como mencionamos anteriormente es un gestor de base de datos, desde Neatbeans realizaremos la conexión a la base de datos a través de métodos DAO y DT.

## **7.8. Desarrollo del sistema.**

En este capítulo se describe a detalle el diseño y desarrollo del sistema administrador. Preferentemente iniciamos con el diseño de la base de datos como lo mencionamos en los capítulos anteriores para tener una idea global de cómo va a funcionar el sistema y como se almacenará la información.

Usaremos el entorno grafico de mysql, puesto que es gratuito y de código abierto para poder crear nuestra base de datos iniciaremos el programa e introduciremos la contraseña de root para poder entrar, dicho proceso se muestra a continuación.

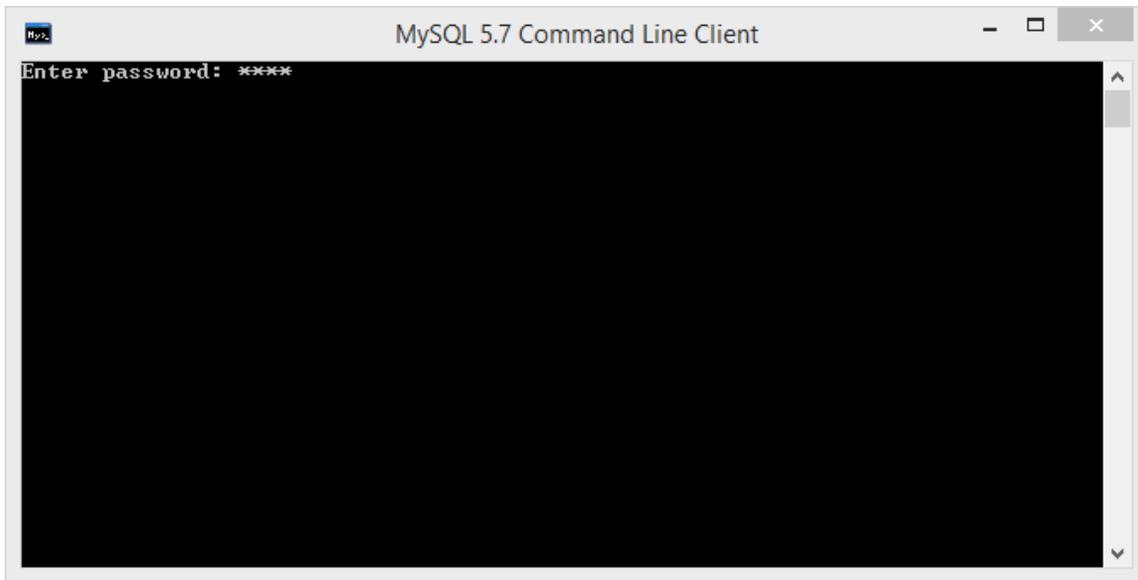


Figura 35 Inicio de Sesión MySQL (Fuente: Propia, 2022)

En la siguiente imagen podemos visualizar la pantalla de inicio de mysql y las bases de datos que tenemos disponibles, en este caso la base de datos que creamos para el proyecto se llama “d”, para crearla usamos el comando “Create data base d;” y para mostrar las bases de datos que tenemos disponibles utilizaremos el comando “show databases;” y presionamos “enter” para ejecutar el comando en la consola.

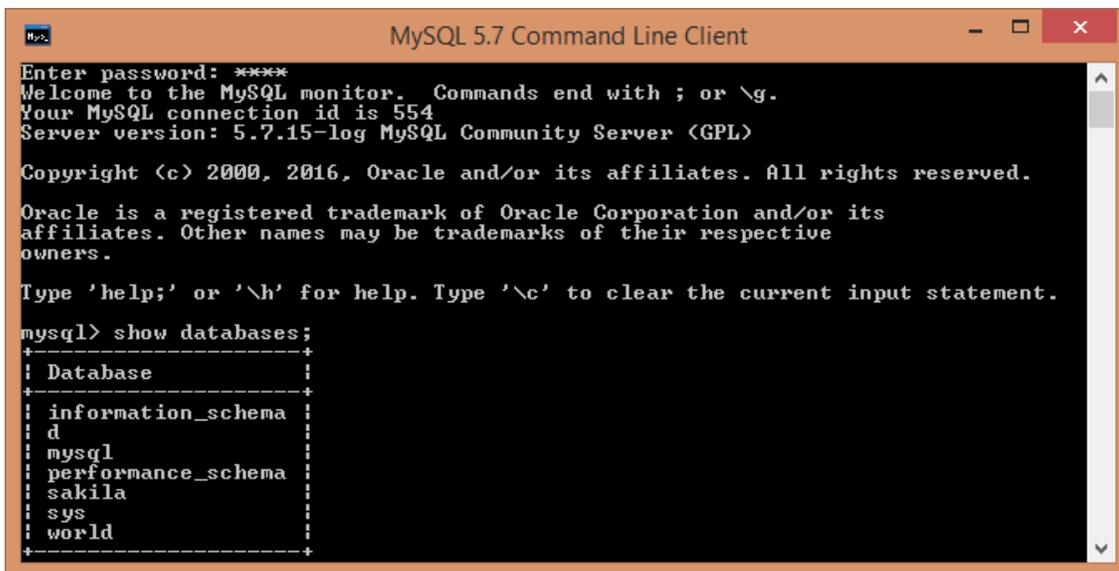
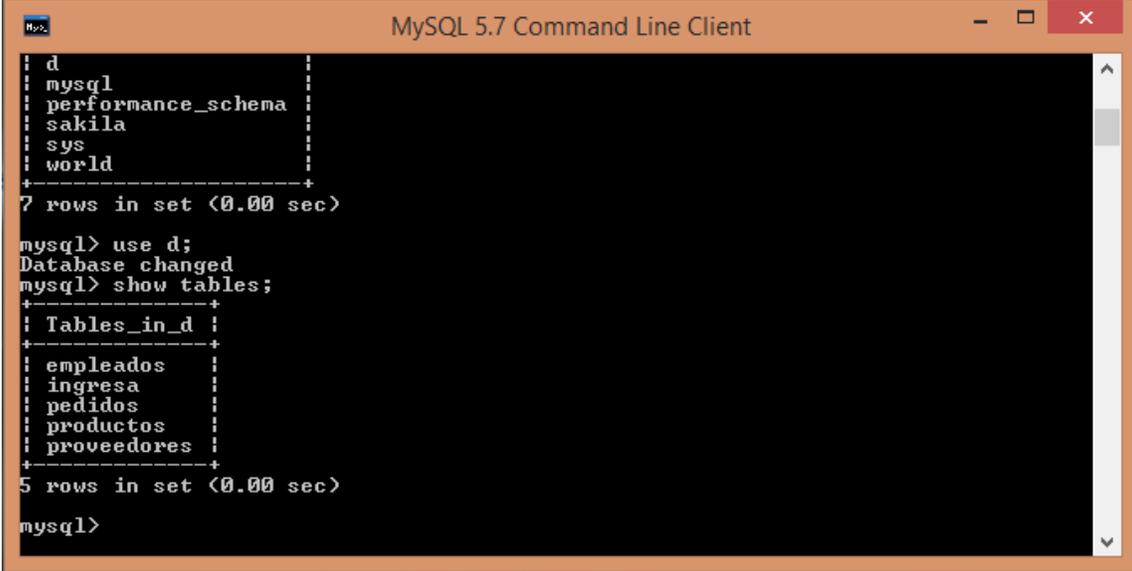


Figura 36 Bases de datos disponibles en mysql. (Fuente: Propia, 2022)

Ahora usaremos el comando “use d;” para entrar en la base de datos de nuestro sistema y el comando “ show tables; ” para visualizar las tablas que contiene y poder visualizar en donde se guardara la información que registremos a través de sistema.



```
MySQL 5.7 Command Line Client
mysql>
+-----+
| d      |
| mysql |
| performance_schema |
| sakila |
| sys   |
| world |
+-----+
7 rows in set (0.00 sec)

mysql> use d;
Database changed
mysql> show tables;
+-----+
| Tables_in_d |
+-----+
| empleados  |
| ingresa    |
| pedidos    |
| productos  |
| proveedores |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Figura 37 Descripción de comandos use y show tables en mysql (Fuente: Propia, 2022)

El código que utilizamos para crear las tablas dependerá del tipo de información que almacenará cada una. También debemos tener en cuenta que debemos establecer la longitud de cada campo de la tabla esto con el fin de evitar errores futuros o de almacenamiento insuficiente o espacios reservados grandes que en realidad contienen poca información.

Para crear las tablas se utiliza el comando “create table + nombre de la tabla (“, después agregamos todo el cuerpo de los campos que la integran como el siguiente ejemplo.



```
mysql> show tables;
+-----+
| Tables_in_d |
+-----+
| empleados  |
| ingresa    |
| pedidos    |
| productos  |
| proveedores|
+-----+
5 rows in set (0.00 sec)

mysql> create table ingresa(
  -> id_nota int,
  -> nombre varchar(40),
  -> empresa varchar(20),
  -> telefono double,
  -> celular double,
  -> email varchar(30),
  -> costoscliente varchar(200),
  -> costosds varchar(200),
  -> total int,
  -> acuenta int,
  -> restan int,
  -> primary key(id_nota));
```

Figura 38 Creación de tablas en mysql (Fuente: Propia, 2022)

Repetimos este proceso con cada tabla tomando en cuenta las relaciones que pueden tener con las demás tablas mediante foreign key. Una vez creada la base de datos y las tablas procederemos a crear el proyecto en Neatbeans, para eso iniciamos el programa y nos mostrara la pantalla de inicio del software, en la sección de nuevo proyecto seleccionamos la opción de Aplicación de java debido a la naturaleza del proyecto que será usada de manera local no necesitamos que sea del tipo de web services ya que estas funciones no serán requeridas por el usuario así que podemos optar por un tipo de proyecto más sencillo.

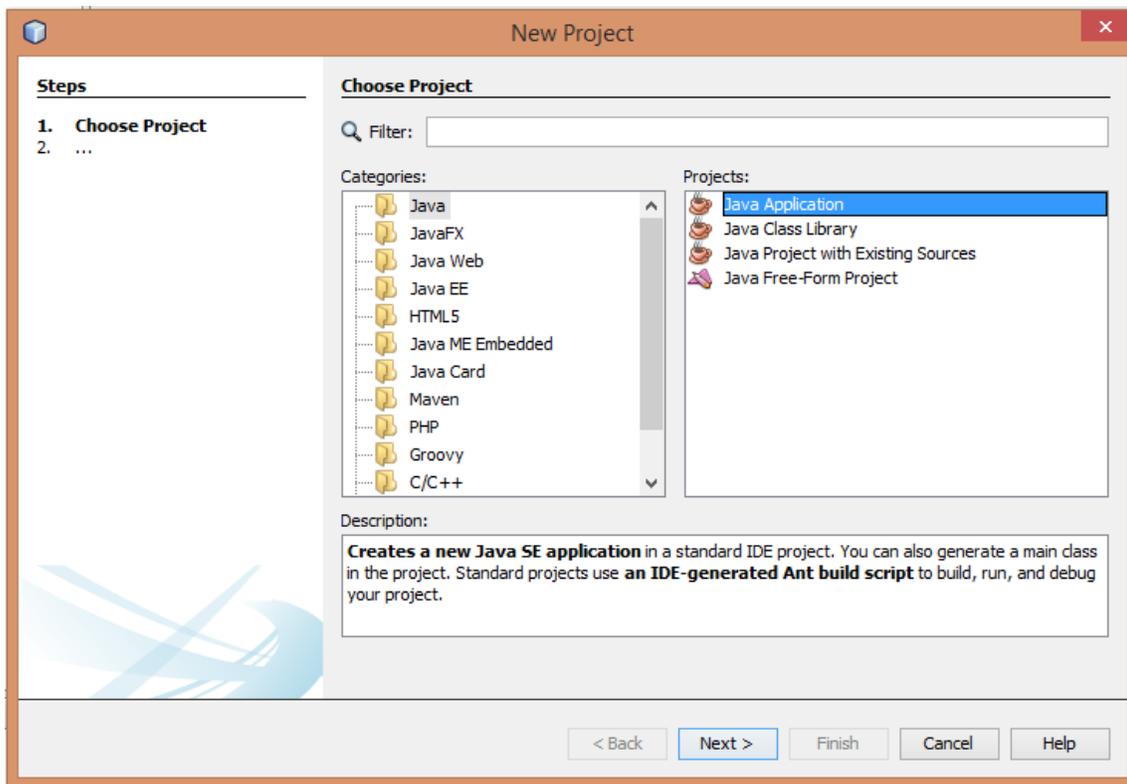


Figura 39 Creación de proyecto nuevo en Java Application (Fuente: Propia, 2022)

Establecemos el nombre del proyecto, seleccionamos la ruta donde se guardarán todos archivos que comprenderán al proyecto esta puede ser en un disco local o en alguna ruta externa tomando en cuenta que si se elige esta última opción tendremos que asegurarnos que tenemos acceso a la ruta ya que si no es así tendremos problemas de conexión, después se creara la Main Class que será la clase principal del proyecto por ultimo indicamos a que deseamos finalizar la creación del proyecto y Neatbeans en automático nos creara el proyecto para empezar a crear nuestras clases y paquetes donde desarrollaremos el proyecto.

La vista principal del proyecto se muestra a continuación, a partir de este paso procederemos a crear todas las clases que serán necesarias para el buen funcionamiento del sistema, crearemos clases de conexión, vistas de usuario para cada área del sistema, clases para modelo DAO y clases para modelo DT.

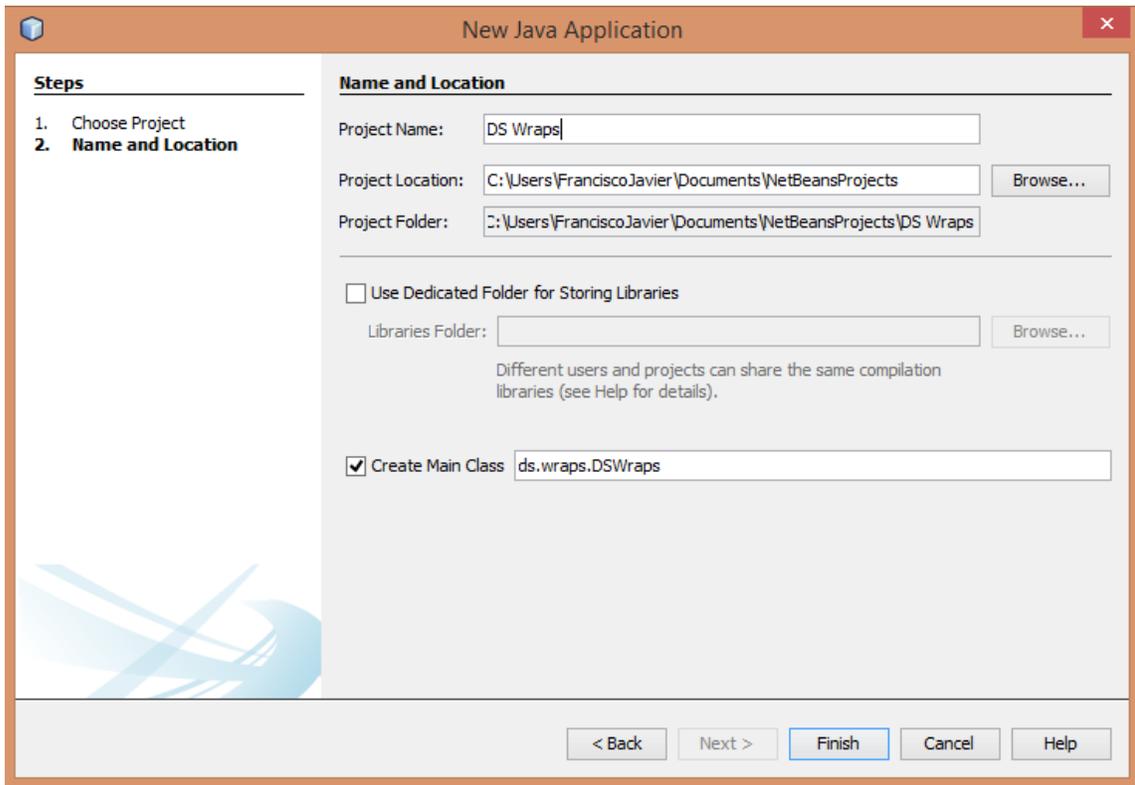


Figura 40 Creación de proyecto, Ruta de proyecto y creación de Main Class (Fuente: Propia, 2022)

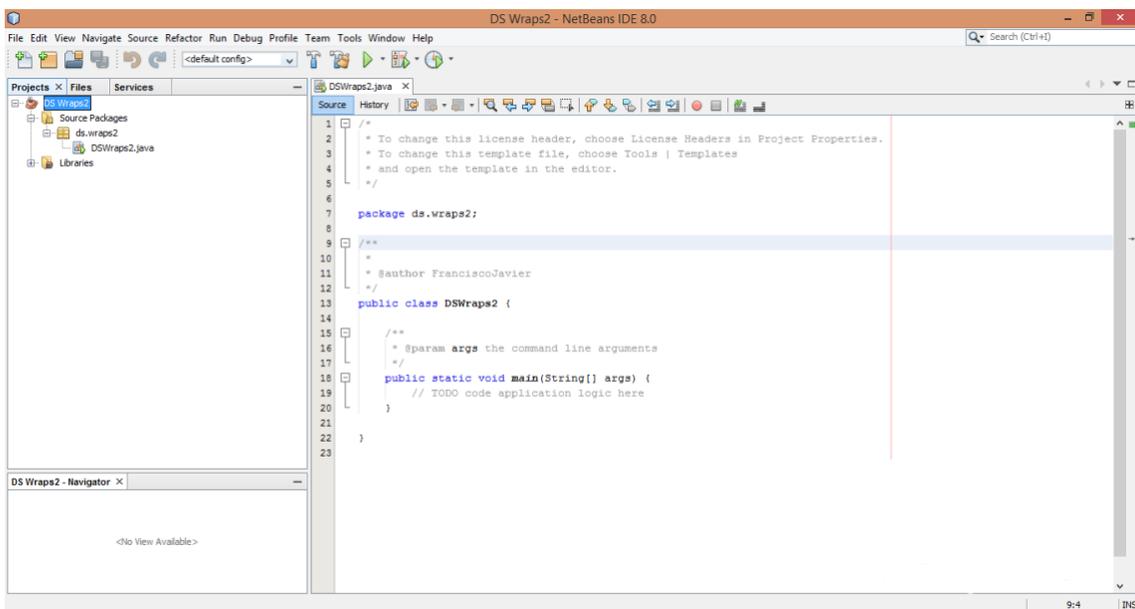


Figura 41 Main Class de DS Wraps. (Fuente: Propia, 2022)

### 7.8.1. Creación de métodos con java interface.

Una interfaz de java consiste en la creación de métodos abstractos con métodos que se declaran, pero no se implementan es decir solo llevan nombre de identificación y parámetros que desea devolver en el método que lo contiene mas no en toda la lógica que debería realizar. Para el desarrollo del proyecto se consideran los métodos DAO debido a que vamos a realizar de manera independiente una clase de tipo Java Interface uno por cada tabla que compone la base de datos con los siguientes métodos: Crear, Actualizar, Buscar, Eliminar y Listar los elementos por tabla de la base de datos.

Ahora crearemos la clase que se encargara de la conexión con la base de datos desde Neatbeans hacia nuestro gestor Mysql. Seleccionamos el proyecto y seleccionamos la opción de crear nuevo paquete dentro del proyecto estos paquetes nos ayudarán a seccionar las diferentes partes del proyecto y dentro de cada paquete crearemos las diferentes clases que comprenderán el sistema.

Crearemos un nuevo paquete al cual llamaremos “modelo.dao” y dentro de este paquete una clase que se llamara conexión en esta clase se encontrara todo el código necesario para realizar la conexión con la base de datos, ahora tendremos que importar el driver que se encargara de dicha conexión en este caso usaremos “Mysql-connector-java-5.1.6-bin.jar” este conector lo agregaremos directamente desde el paquete llamado “Libraries” el cual se crea en automático al crear el proyecto.

Una vez importado el conector dentro del paquete modelo.dao crearemos la clase conexión e importaremos las librerías necesarias, como mencionamos anteriormente esta clase será la encargada de realizar la conexión mediante el siguiente código en el que debemos especificar el usuario que se utilizó para crear la base de datos, la contraseña y el nombre de la base de datos:

```

1 package modelo.dao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8
9 /**
10  *
11  * @author FranciscoJavier
12  */
13 public class ConexionBD {
14     protected Connection conn;
15
16     protected void cerrar(PreparedStatement stm) throws
17 Exception{
18         stm.close();
19     }
20     protected void cerrar(ResultSet rst) throws Exception{
21         rst.close();
22     }
23
24     /** Creamos una instancia */
25     public ConexionBD() {
26         String driver="org.gjt.mm.mysql.Driver";
27         String user="root";
28         String pwd="1234";
29         String basedatos="d";
30         String server="jdbc:mysql://localhost/"+basedatos;
31         try{
32             Class.forName(driver);
33             conn=DriverManager.getConnection(server,user,pwd);
34         }
35         catch(SQLException e){
36             System.out.println(e);
37         }
38         catch(ClassNotFoundException e){
39             System.out.println(e);
40         }
41     }

```

*Tabla 1 Conexión hacia base de datos (Fuente Propia, 2022)*

## 7.8.2. Implementando clases interface

Dentro del mismo paquete donde creamos la clase ConexiónBD, crearemos una nueva Clase publica con extensión a la clase ConexiónBD dentro de esta crearemos los métodos para poder ingresar, buscar, listar actualizar o eliminar datos dependiendo de la tabla a la cual estemos haciendo referencia.

```
/**
 *
 1  * @author FranciscoJavier
 */
2  public class DSDAO extends ConexionBD {
3      private final static String SQL_INSERT="insert into
4      ingresa(id_nota,nombre,empresa,telefono,celular,email,ostoscliente,ostosds,total,acuenta,
      restan)values(?,?,?,?,?,?,?,?,?)";
5      private final static String SQL_SELECT="select * from ingresa where id_nota=? or
6      nombre=? or empresa =?";
7      private final static String SQL_SELECTALL="select * from ingresa";
8      private final static String SQL_DELETE = "delete from ingresa where id_nota=?";
9      private final static String SQL_UPDATE = "update ingresa set
      nombre=?, empresa=?, telefono=?, celular=?, email=?,ostoscliente=?,ostosds=?,total=?,acu
      enta=?,restan=? where id_nota =?";
1     public DSDAO() {
4         super();
     }
}
```

Tabla 2 Creación de métodos para manejo de datos (Fuente Propia, 2022)

El método SQL\_INSERT se encargará de enviar la información a la base de datos a través de un formulario donde tendremos que especificar todos los datos que requiere la tabla a la cual hacemos mención o queremos ingresar datos nuevos.

```
1 public void create(ClienteDt dto) throws Exception{
2     PreparedStatement ps = null;
3     ps=conn.prepareStatement ( SQL_INSERT);
4     ps.setInt (1, dto.getId_nota ());
5     ps.setString (2, dto.getNombre ());
6     ps.setString (3, dto.getEmpresa ());
```

```

7      ps.setDouble (4, dto.getTelefono ());
8      ps.setDouble (5, dto.getCelular ());
9      ps.setString (6, dto.getEmail ());
10     ps.setString (7, dto.getCostoscliente ());
11     ps.setString (8, dto.getCostosds ());
12     ps.setInt (9, dto.getTotal ());
13     ps.setInt (10, dto.getAcuenta ());
14     ps.setInt (11, dto.getRestan ());
15     ps.executeUpdate ();
16     cerrar (ps);
17 }

```

*Tabla 3 Método Insert para ingresar nuevos datos (Fuente Propia, 2022)*

El método SQL\_SELECT se encargará de buscar información almacenada, en la base para realizar esta función tendremos que proporcionarle al sistema el número de identificación y el nombre o empresa del cliente, proveedor o empleado que se requiera consultar.

```

1 public ClienteDt select (ClienteDt dto) throws Exception{
2     PreparedStatement ps=null;
3     ResultSet rs=null;
4     ps = conn.prepareStatement (SQL_SELECT);
5     ps.setInt (1, dto.getId_nota ());
6     ps.setString (2, dto.getNombre ());
7     ps.setString (3, dto.getEmpresa ());
8
9     rs=ps.executeQuery ();
10    if (rs.next ()) {
11        dto.setId_nota (rs.getInt ("Id_Nota"));
12        dto.setNombre (rs.getString ("Nombre"));
13        dto.setEmpresa (rs.getString ("Empresa"));
14        dto.setTelefono (rs.getDouble ("Telefono"));
15        dto.setCelular (rs.getDouble ("Celular"));
16        dto.setEmail (rs.getString ("Email"));
17        dto.setCostoscliente (rs.getString ("CostosCliente"));
18        dto.setCostosds (rs.getString ("CostosDs"));
19        dto.setTotal (rs.getInt ("Total"));
20        dto.setAcuenta (rs.getInt ("ACuenta"));
21        dto.setRestan (rs.getInt ("Restan"));
22    }
23    else
24        return null;
25    return dto;
26 }
27

```

*Tabla 4 Método Select para realizar búsqueda de información (Fuente Propia, 2022)*

El método SQL\_SELECT\_ALL se encargará de mostrar una lista con toda la información que contenga la tabla a la cual estemos haciendo referencia.

```
1     public List selectall() throws Exception{
2         List lista = new ArrayList();
3         PreparedStatement ps=null;
4         ResultSet rs=null;
5         ClienteDt dto;
6         ps=conn.prepareStatement (SQL_SELECTALL);
7
8         rs=ps.executeQuery ();
9         while (rs.next ()) {
10            dto = new ClienteDt ();
11            dto.setId_nota (rs.getInt ("Id_Nota"));
12            dto.setNombre (rs.getString ("Nombre"));
13            dto.setEmpresa (rs.getString ("Empresa"));
14            dto.setTelefono (rs.getDouble ("Telefono"));
15            dto.setCelular (rs.getDouble ("Celular"));
16            dto.setEmail (rs.getString ("Email"));
17            dto.setCostoscliente (rs.getString ("CostosCliente"));
18            dto.setCostosds (rs.getString ("CostosDs"));
19            dto.setTotal (rs.getInt ("Total"));
20            dto.setAcuenta (rs.getInt ("ACuenta"));
21            dto.setRestan (rs.getInt ("Restan"));
22            lista.add (dto);
23        }
24        return lista;
25    }
26 }
```

*Tabla 5 Método Select\_All muestra una lista con la información almacenada (Fuente Propia, 2022)*

El método SQL\_DELETE se encargará de eliminar los datos que el administrador necesite eliminar.

```
1     public void delete (ClienteDt dto) throws Exception{
2         PreparedStatement ps=null;
3         ps=conn.prepareStatement (SQL_DELETE);
4         ps.setInt (1, dto.getId_nota ());
5         ps.executeUpdate ();
6         cerrar (ps);
7     }
```

*Tabla 6 Método Delete elimina registros de la base de datos (Fuente Propia, 2022)*

El método SQL\_UPDATE se encargará de actualizar los datos previamente ingresados a la base ya sea por error de captura o simplemente por actualización o cambios en los datos previamente ingresados.

```
1      public void update (ClienteDt dto) throws Exception{
2          PreparedStatement ps=null;
3          ps=conn.prepareStatement (SQL_UPDATE);
4
5          ps.setString (1, dto.getNombre () );
6          ps.setString (2, dto.getEmpresa () );
7          ps.setDouble (3, dto.getTelefono () );
8          ps.setDouble (4, dto.getCelular () );
9          ps.setString (5, dto.getEmail () );
10         ps.setString (6, dto.getCostoscliente () );
11         ps.setString (7, dto.getCostosds () );
12         ps.setInt (8, dto.getTotal () );
13         ps.setInt (9, dto.getAcuenta () );
14         ps.setInt (10, dto.getRestan () );
15         ps.setInt (11, dto.getId_nota () );
16         ps.executeUpdate ();
17         cerrar (ps);
18     }
19
20 }
```

*Tabla 7 Método Update actualiza datos en la base de datos (Fuente Propia, 2022)*

### 7.8.3. Creación Modelos DT

En esta clase se crearán los métodos Getters y Setters. Estos métodos como su nombre lo indica el método getter indica que podemos obtener algún valor de un atributo, y el método setter sirve para definir o guardar algún valor sobre el atributo. Estos métodos son importantes al momento de crear una clase objeto ya que de ellos depende el valor que pueden tomar los atributos o a su vez para modificar algún otro atributo.

En los métodos Getters siempre nos retornaran el valor del atributo sin necesidad de pasar por ningún parámetro, mientras que los métodos Setters

siempre nos pedirá algún valor como parámetro para guardarlo al atributo de la clase y este nunca retornara ningún valor, a continuación, mostraremos lo métodos creados en la clase “Cliente Dt.”

```
1 package modelo.dt;
2
3 /**
4  *
5  * @author FranciscoJavier
6  */
7 public class ClienteDt {
8
9     private int id_nota;
10    private String nombre;
11    private String empresa;
12    private double telefono;
13    private double celular;
14    private String email;
15    private String costoscliente;
16    private String costosds;
17    private int total;
18    private int acuenta;
19    private int restan;
20
21    /**
22     * @return the id_nota
23     */
24    public int getId_nota() {
25        return id_nota;
26    }
27
28    /**
29     * @param id_nota the id_nota to set
30     */
31    public void setId_nota(int id_nota) {
32        this.id_nota = id_nota;
33    }
34
35    /**
36     * @return the nombre
37     */
38    public String getNombre() {
39        return nombre;
40    }
41
42    /**
43     * @param nombre the nombre to set
44     */
```

```

45     public void setNombre(String nombre) {
46         this.nombre = nombre;
47     }
48
49     /**
50      * @return the empresa
51      */
52     public String getEmpresa() {
53         return empresa;
54     }
55
56     /**
57      * @param empresa the empresa to set
58      */
59     public void setEmpresa(String empresa) {
60         this.empresa = empresa;
61     }
62
63     /**
64      * @return the telefono
65      */
66     public double getTelefono() {
67         return telefono;
68     }
69
70     /**
71      * @param telefono the telefono to set
72      */
73     public void setTelefono(double telefono) {
74         this.telefono = telefono;
75     }
76
77     /**
78      * @return the celular
79      */
80     public double getCelular() {
81         return celular;
82     }
83
84     /**
85      * @param celular the celular to set
86      */
87     public void setCelular(double celular) {
88         this.celular = celular;
89     }
90
91     /**
92      * @return the email
93      */
94     public String getEmail() {
95         return email;

```

```

96     }
97
98     /**
99     * @param email the email to set
100    */
101    public void setEmail(String email) {
102        this.email = email;
103    }
104
105    /**
106     * @return the costoscliente
107     */
108    public String getCostoscliente() {
109        return costoscliente;
110    }
111
112    /**
113     * @param costoscliente the costoscliente to set
114     */
115    public void setCostoscliente(String costoscliente) {
116        this.costoscliente = costoscliente;
117    }
118
119    /**
120     * @return the costosds
121     */
122    public String getCostosds() {
123        return costosds;
124    }
125
126    /**
127     * @param costosds the costosds to set
128     */
129    public void setCostosds(String costosds) {
130        this.costosds = costosds;
131    }
132
133    /**
134     * @return the total
135     */
136    public int getTotal() {
137        return total;
138    }
139
140    /**
141     * @param total the total to set
142     */
143    public void setTotal(int total) {
144        this.total = total;
145    }
146

```

```

147     /**
148      * @return the acuenta
149      */
150     public int getAcuenta() {
151         return acuenta;
152     }
153
154     /**
155      * @param acuenta the acuenta to set
156      */
157     public void setAcuenta(int acuenta) {
158         this.acuenta = acuenta;
159     }
160
161     /**
162      * @return the restan
163      */
164     public int getRestan() {
165         return restan;
166     }
167
168     /**
169      * @param restan the restan to set
170      */
171     public void setRestan(int restan) {
172         this.restan = restan;
173     }
174
175 }

```

*Tabla 8 Creación de clase Cliente Dt (Fuente Propia, 2022)*

#### 7.8.4. Modelo Facade

El modelo facade es un patrón de diseño estructural que proporciona una interfaz simplificada a un sistema complejo de clases, bibliotecas o frameworks. El patrón facade disminuye la complejidad general de la aplicación, al mismo tiempo que ayuda a mover dependencias no deseadas a un solo lugar. Su aplicación se puede utilizar en los siguientes casos:

- Tener una interfaz simple de un subsistema complejo.
- Estructurar varios subsistemas en capas, debido a que las fachadas serían los puntos de entrada a cada nivel.

- Desacoplar un sistema de clientes, reduciendo dependencias entre los subsistemas y los clientes.

En esta clase se importarán los modelos de las clases Dao y Dt, después se crearán métodos que hagan referencia a la clase y a la función que se requiera usar ya sea crear, buscar o listar clientes, empleado, notas, pedidos, productos o proveedores.

```

1 package modelo.facade;
2
3 /**
4  *
5  * @author FranciscoJavier
6  */
7 import modelo.dao.DSDAO;
8 import modelo.dt.ClienteDt;
9 import java.util.List;
10 import modelo.dt.EmpleadosDT;
11 import modelo.dt.PedidosDT;
12 import modelo.dt.ProductosDT;
13 import modelo.dt.ProveedoresDT;
14 import modelo.dao.EmpleadosDAO;
15 import modelo.dao.PedidosDAO;
16 import modelo.dao.ProductosDAO;
17 import modelo.dao.ProveedoresDAO;
18
19
20 public class facade {
21
22     public void nuevoCliente(ClienteDt dt) throws Exception{
23         DSDAO dao = new DSDAO();
24         dao.create(dt);
25     }
26
27     public ClienteDt buscaClienteDT(ClienteDt dt) throws
28 Exception{
29         DSDAO dao = new DSDAO();
30         return dao.select(dt);
31     }
32
33     public List listaCliente() throws Exception{
34         DSDAO dao = new DSDAO();
35         return dao.selectAll();
36     }

```

```

37
38 public Object[] numeros(int inicio,int fin){
39     Object[] num =new Object[(fin-inicio+1)];
40     int y=0;
41     for(int x=inicio; x<=fin; x++){
42         num[y]=x;
43         y++;
44     }
45     return num;
46 }
47
48
49 // Empleados
50
51 public void nuevoEmpleado(EmpleadosDT dt) throws Exception{
52     EmpleadosDAO dao = new EmpleadosDAO();
53     dao.create(dt);
54 }
55     public EmpleadosDT buscaEmpleadosDT(EmpleadosDT dt) throws
56 Exception{
57     EmpleadosDAO dao = new EmpleadosDAO();
58     return dao.select(dt);
59 }
60
61 public List listaEmpleado() throws Exception{
62     EmpleadosDAO dao = new EmpleadosDAO();
63     return dao.selectall();
64 }
65
66 // Pedidos
67
68 public void nuevoPedido(PedidosDT dt) throws Exception{
69     PedidosDAO dao = new PedidosDAO();
70     dao.create(dt);
71 }
72
73     public PedidosDT buscaPedidosDT(PedidosDT dt) throws Exception{
74     PedidosDAO dao = new PedidosDAO();
75     return dao.select(dt);
76 }
77
78 public List listaPedido() throws Exception{
79     PedidosDAO dao = new PedidosDAO();
80     return dao.selectall();
81 }
82
83 // Productos
84
85 public void nuevoProducto(ProductosDT dt) throws Exception{
86     ProductosDAO dao = new ProductosDAO();
87     dao.create(dt);

```

```

88     }
89     public ProductosDT buscaProductosDT(ProductosDT dt) throws
90 Exception{
91         ProductosDAO dao = new ProductosDAO();
92         return dao.select(dt);
93     }
94     public List listaProducto() throws Exception{
95         ProductosDAO dao = new ProductosDAO();
96         return dao.selectall();
97     }
98
99     // Proveedores
100
101     public void nuevoProveedor(ProveedoresDT dt) throws Exception{
102         ProveedoresDAO dao = new ProveedoresDAO();
103         dao.create(dt);
104     }
105
106     public ProveedoresDT buscaProveedorDT(ProveedoresDT dt) throws
107 Exception{
108         ProveedoresDAO dao = new ProveedoresDAO();
109         return dao.select(dt);
110     }
111
112     public List listaProveedor() throws Exception{
113         ProveedoresDAO dao = new ProveedoresDAO();
114         return dao.selectall();
115     }
116 }

```

*Tabla 9 Creación de clase facade e instanciación de todas las clases que integran el sistema (Fuente Propia, 2022)*

## **VIII. Conclusiones y sugerencias.**

Contar con un sistema administrador que facilite las tareas es de gran utilidad cuando lo que se busca es aumentar el control sobre la información que maneja un negocio o empresa, al contar con un sistema hecho a la medida esta tarea se cumple de manera más rápida y segura debido a que el diseño y desarrollo de este se realizó en virtud de resolver los problemas a los que nos enfrentábamos.

De esta manera podremos almacenar y dotar la información de manera adecuada a los usuarios y administradores, ofreciendo mayor seguridad y tranquilidad, en que los problemas anteriores se verán notablemente disminuidos al cambiar de un sistema de notas de remisión en papel a un control digital de toda la información y datos relacionados a la empresa en cuestión.

Terminamos este proyecto con una gran satisfacción cumpliendo con las expectativas de nuestro cliente y los objetivos planteados al inicio de este proyecto. Las Dificultades encontradas durante el desarrollo de nuestro trabajo nos demostraron que si trabajamos centrados en nuestros objetivos sin olvidar las necesidades principales de nuestro cliente podemos concluir cualquier proyecto con los mejores resultados.

Algunas sugerencias que podríamos brindar para terminar son:

- tener en cuenta siempre las necesidades del sistema en virtud de que este pueda realizar su trabajo de la mejor manera.
- Cuidar la integridad del equipo de cómputo principal para asegurar el buen funcionamiento del sistema.

- Debido a la arquitectura del sistema no es necesario realizar muchas tareas, solo mantener actualizado el sistema operativo y las aplicaciones con las que se desarrolló.
- Como trabajo a futuro se podría sugerir que el sistema sea más intuitivo con los usuarios de mayor interacción.
- Por último, debido a los cambios constantes en la tecnología se podría llegar a implementar otro tipo de función particular que ayude a mejorar el sistema o el desarrollo de un nuevo sistema en otro tipo de arquitectura con nuevas funciones y características.

## IX. Referencias de consulta

Alberto Barran. (2012). Seminario-IEE. México: UNAM.

Blancarte, O. (12 de 02 de 2021). Data Transfer Object (DTO) – Patrón de diseño. Obtenido de Data Transfer Object (DTO) – Patrón de diseño: <https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dtopatron-diseno/>

Concepto de Software de aplicación (15 de 05 de 2018): <https://concepto.de/software-de-aplicacion/>

Desconocido. (2013). MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN. México: UNAP.

DTTEC. (3 de 04 de 2011). Viniles. 08/10/2017, de DTTEC IMPRESORAS E INSUMOS Sitio web: <http://dttec.com.mx/55-viniles>

Ester Maya. (09 de 10 de 2017). MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN. de UNAM Sitio web: [http://arquitectura.unam.mx/uploads/8/1/1/0/8110907/metodos\\_y\\_tecnicas.pdf](http://arquitectura.unam.mx/uploads/8/1/1/0/8110907/metodos_y_tecnicas.pdf)

Freedman Alan, Diccionario de computación, (5 edición), México, McGraw-Hill, 1993, pag 581.

Jorge Martínez Ladrón de Guevara (2006), Fundamentos de programación java, Editorial Eme.

Jordi Sánchez (22 de 01 de 2012). ¿Qué es un framework ? 08102017, de jorsysan.net Sitio <https://jordisan.net/blog/2006/que-es-un-framework/>

Larman, C. (2003). UML Y PATRONES Una introducción al análisis y diseño orientado a objetos y al proceso unificado . Madrid: Pearson.

Laudon, K. C., & Laudon, J. P. (2012). SISTEMAS DE INFORMACIÓN GERENCIAL. México: Pearson.

Luis Alberto Casillas (11 de 11 de 2005), Software Libre Recuperado de: <https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>

López Román, L. (2013). *Metodología de la programación orientada a objetos*. México: Alfaomega.

María, E. Raffino. (03 de mayo 2011). Software de Aplicación. Argentina.

Ortiz Uribe, Frida (2011). Diccionario de metodología de la investigación científica. México: Limusa. Rojas Soriano, Raúl (2010). Guía para realizar investigaciones sociales: México: Laza y Valdés editores.

PRINTIKA. (2016). Impresión Vinil Adhesivo. 08/10/2017, de PRINTIKA Sitio web: <https://ptk.mx/online/impresion/impresion-en-flexibles/impresion-vinil-autoadhesivo/>

Pressman, R. S. (2010). Ingeniería del Software un enfoque práctico séptima edición. México: McGraw-Hill.

Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *EL LENGUAJE UNIFICADO DE MODELADO MANUAL DE REFERENCIA*. España: Pearson.

Senn, J. A. (1992). *ANÁLISIS y DISEÑO de SISTEMAS DE INFORMACIÓN*. México: McGrawHill.

Sommerville, I. (2005). *Ingeniería de Software Séptima edición*. Madrid: Pearson Addison Wesley.

Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE Novena edición*. México: Pearson.

Victorio Muñoz Rosales. (2002). *Técnicas De Investigación De Campo I*. México: E.N.B.A.