



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM TEXCOCO

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE
INFORMACIÓN “INTELESIS 13” PARA EL TRATAMIENTO DE
SCRAP (MERMA) EN EL DEPARTAMENTO DE MATERIA PRIMA EN
“INDUSTRIAL DE HERRAMIENTAS GONI”.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN INFORMÁTICA ADMINISTRATIVA

PRESENTA

RODRÍGUEZ GARCÍA ROBERTO FAUSTO

ASESOR

M. EN C.C. JUAN CARLOS MORENO SÁNCHEZ

REVISORES

DR. EN I. DE S. JOSÉ SERGIO RUIZ CASTILLA

DR. EN C. ADRIÁN TRUEBA ESPINOSA

TEXCOCO, ESTADO DE MÉXICO, MARZO 2017.

OFICIO DE DIGITALIZACION

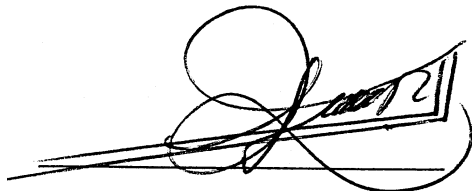
Texcoco, México a 14 de febrero de 2017

M. EN C. E. VIRIDIANA BANDA ARZATE
SUBDIRECTORA ACADÉMICA DEL
CENTRO UNIVERSITARIO UAEM TEXCOCO,
PRESENTE:

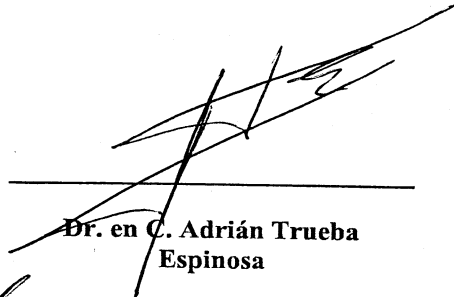
AT'N L. EN D. MARCO RODRIGO LÓPEZ GONZÁLEZ
RESPONSABLE DEL DEPARTAMENTO DE TITULACIÓN.

Con base en las revisiones efectuadas al trabajo escrito titulado **Desarrollo e implementación del sistema de información "INTELEISIS 13" para el tratamiento de SCRAP (merma) en el departamento de materia prima en "industrial de herramientas Goni"**. Que para obtener el título de **Licenciado en informática administrativa** presenta el sustentante **Rodríguez García Roberto Fausto**, con número de cuenta **1124713** respectivamente, se concluye que cumple con los requisitos teórico-metodológicos por lo que se le otorga el voto aprobatorio para su sustentación, pudiendo **continuar con la etapa de digitalización** del trabajo escrito.

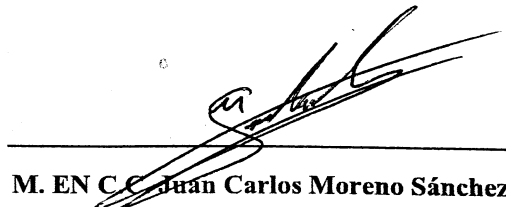
ATENTAMENTE



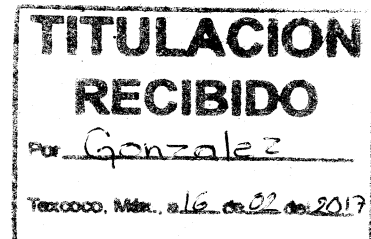
Dr. en I. de S. José Sergio Ruiz
Castilla



Dr. en C. Adrián Trueba
Espinosa



M. EN C.C. Juan Carlos Moreno Sánchez



C.C.P. (Rodríguez García Roberto Fausto)
C.C.P. (M. EN C.C. Juan Carlos Moreno Sánchez)
C.C.P. (L. EN D. Marco Rodrigo López González)

DEDICATORIAS

A mis padres.

Porque siempre han creído en mí y han sabido impulsarme y guiarme en alcanzar una meta más. Gracias por haber motivado en mí el deseo de superación y anhelo de triunfo y enseñarme que todo se puede lograr.

A mi hermano y hermana.

A mi hermana y hermano Por apoyarme y que este logro sea una motivación extra en su vida.

A mi tía.

Porque siempre me apoyo en todo momento y cuido de mi durante mi infancia.

AGRADECIMIENTOS

A mi asesor de tesis

Por guiarme en el camino de la investigación y en la realización de esta tesis.

A mis revisores

Por brindarme su tiempo y amabilidad para hacerme las observaciones oportunas.

A mi universidad

Porque en sus aulas alberga los sueños de los estudiantes y les brinda la oportunidad de hacerlos realidad al egresar como profesionales.

RESUMEN

La empresa industrial de herramientas Goni S.A de C.V, ubicada en Azcapotzalco, México DF. Es una empresa dedicada al diseño, fabricación así como comercialización de equipos de aire, aplicaciones de pintura, accesorios y consumibles. Goni se destaca de su competencia debido a la calidad de sus productos, sin embargo dentro la empresa existen problemas en el proceso productivo específicamente con el *scrap* (merma del proceso productivo) la cual es una pérdida monetaria para cualquier empresa debido a esto, se analizó el proceso productivo en el almacén de materia prima de la empresa Goni. Ahora bien, dentro de este proceso no hay un control específico para el tratamiento de información de *scrap* que se genera en los diferentes departamentos de la empresa y que posteriormente se envían al almacén de materia prima. Así mismo, cabe mencionar que se elabora un informe manual de cada pieza de *scrap* generada con los siguientes atributos: código del producto, nombre y cantidad. Por lo mencionado, anteriormente se desarrolló un sistema de información llamado INTELESIS 13 el cual lleva el control de toda la información de la merma. La metodología empleada para el desarrollo del sistema de información fue el modelo incremental o evolutivo el cual satisface las necesidades más importantes del cliente con el primer incremento. Tomando en cuenta que el almacén de materia prima no tiene un sistema para el tratamiento de información es una prioridad la implementación de un sistema de información y la metodología incremental permite hacerlo desde su primer incremento. En definitiva, con la implementación del sistema de información se ha generado un mayor control así como protección de la información dentro del almacén de materia prima, debido al fácil manejo de información que ahora se lleva así mismo, los reportes de las entradas y salidas de *scrap* son más seguros y veraces ayudando a que el supervisor del departamento tenga pruebas contundentes para las auditorías realizadas por la empresa.

Tabla de contenido

Resumen	iv
II Introducción	1
2.1 Antecedentes de la empresa	2
2.1.1 Reseña histórica de industrial de herramientas Goni	2
2.1.2 Productos	3
III Planteamiento del problema.	4
IV Justificación	5
V Objetivo	6
5.1 Objetivos particulares	6
VI Marco teórico	7
6.1 Sistemas de información	7
6.1.1 Características de los sistemas de información	7
6.1.2 Actividades básicas de un sistema de información	8
6.1.3 Tipos de sistemas de información	10
6.2 Metodologías de desarrollo de software.	12
6.2.1 Modelos de desarrollo de software.....	14
6.3 Base de datos	16
6.3.1 Sistema de Gestión de Base de Datos (SGBD)	16
6.3.2 Motores de Base de Datos	18
6.3.3 SQL y MySQL	18
6.3.4 Modelo de base de datos.	19
6.3.5 Modelo relacional.....	19
6.3.6 Normalización	23
6.4 Lenguaje de programación Java.....	25
6.4.1 Historia de java.....	25
6.4.2 Características de java.....	25
6.4.3 Plataformas.....	29
6.4.4 JDBC API.....	30

6.5 Patrones de diseño	32
6.5.1 Patrón de diseño DTO	32
6.5.2 Patrón de diseño DAO.....	32
6.5.3 Patrón de diseño FACADE	33
6.5.4 Patrón de diseño DELEGATE	33
6.5.5 Framework	34
6.6 Ubuntu	35
6.6.1 Ventajas de Ubuntu	35
6.6.2 Desventajas de Ubuntu.....	36
VII Metodología.	37
7.1 Análisis y diseño del sistema	37
7.1.1 Cuestionario de requerimientos de software	37
7.1.2 Control de mermas en el almacén de materia prima.	38
7.1.3 Organización y depuración de la información	40
7.1.4 Diagrama de casos de uso	41
7.1.5 Diagramas de actividades.....	43
7.1.6 Diseño de interfaces (Prototipo).....	55
7.2 Implementación del sistema.....	63
7.2.1 Creación de la base de datos relacional.....	63
7.2.2 Creación de las vistas (interfaces).....	65
7.2.3 Codificación de aplicación (clase controlador).....	73
7.2.4 Implementación de políticas de seguridad para INTELESIS 13.....	78
7.3 Pruebas y ajustes	80
7.3.1 Registros de ejecución de pruebas	80
7.3.2 Prueba unitaria.....	80
7.3.3 Pruebas funcionales.....	92
7.3.4 Pruebas de integración	92
7.3.5 Pruebas de validación.....	92
7.3.6 Prueba de aceptación.....	92
7.4 Entrega	93
7.5 Capacitación.....	94

VIII Resultados.....	95
8.1 Presentación de resultados	95
IX Discusión.....	96
X Conclusiones.....	97
XI Bibliografía.....	98
Anexos	101
Anexo (A) Cuestionario de requerimientos de software	101

Lista de tablas

Tabla 1. Tipos de sistemas de información. (Pressman, 2010).	11
Tabla 2. Modelos de desarrollo de software. (Pressman, 2010).....	14
Tabla 3. Ventajas y desventajas de un SGBD. (Pressman, 2010).	17
Tabla 4. Motores de base de datos. (González, 2013).....	18
Tabla 5. Estructura de una base de datos relacional. (Sánchez, 2016).	20
Tabla 6. Restricciones semánticas del modelo relacional. (Sánchez, 2016).	21
Tabla 7. Plataformas de java. (Sánchez, 2009).	29
Tabla 8. Funciones de los botones de la interfaz menú, (Autoría propia).....	56

Lista de figuras

Figura 1. Actividades de un sistema de información. (Zamora, 2016).	8
Figura 2. Tipos de sistemas de información. (Guevara, 2013).....	10
Figura 3. Ciclo de vida del sistema. (Pluke. 2011).....	12
Figura 4. Compilación de un programa C en Windows y Linux. (Sánchez 2009).....	26
Figura 5. Proceso de compilación de un programa Java. (Sánchez 2009).	27
Figura 6. JDBC API. (Valdez, 2011).....	30
Figura 7. Catálogo de multi-almacén. (Industrial de herramientas Goni)	38
Figura 8. Reporte de SCRAP de producción. (Industrial de herramientas goni).	39
Figura 9. Diagrama de casos de uso de INTELESIS 13. (Autoría propia).....	42
Figura 10. Diagrama de actividades, manipulación de la base de datos. (Autoría propia). .	43
Figura 11. Ingresa información del producto. (Autoría propia).	44
Figura 12. Ingresa información del empleado. (Autoría propia).....	45
Figura 13. Ingresa información del estatus. (Autoría propia).	45
Figura 14. Ingresa información de la orden. (Autoría propia).....	46
Figura 15. Actualización y eliminación de información, Empleado. (Autoría propia)	47
Figura 16. Actualización y eliminación de información Producto. (Autoría propia).....	48
Figura 17. Actualización y eliminación de información Estatus. (Autoría propia).....	48
Figura 18. Actualización y eliminación de información Orden. (Autoría propia).	49
Figura 19. Consulta de información Empleado. (Autoría propia).....	50
Figura 20. Consulta de información Producto, (Autoría propia).....	51
Figura 21. Consulta de información Estatus. (Autoría propia).....	51
Figura 22. Consulta de información Orden. (Autoría propia).	52
Figura 23. Impresión de órdenes. (Autoría propia).	53
Figura 24. Generar órdenes. (Autoría propia).	54
Figura 25. Interfaz login. (Autoría propia).....	55
Figura 26. Interfaz menú, (Autoría propia).	56
Figura 27. Interfaz empleado. (Autoría propia).....	57
Figura 28. Interfaz producto. (Autoría propia).....	58
Figura 29. Interfaz diagnóstico. (Autoría propia).....	59
Figura 30. Interfaz estatus. (Autoría propia).	60
Figura 31. Interfaz orden. (Autoría propia).	61
Figura 32. Interfaz consultas. (Autoría propia).	62
Figura 33. Modelo relacional SCRAP. (Autoría propia).....	64
Figura 34. Interfaz LOGIN. (Autoría propia).....	65
Figura 35. Interfaz menú. (Autoría propia)	66
Figura 36. Interfaz empleado. (Autoría propia).....	67
Figura 37. Interfaz producto. (Autoría propia).....	68
Figura 38. Interfaz estatus. (Autoría propia)	69
Figura 39. Interfaz generar orden. (Autoría propia)	70
Figura 40. Interfaz consultas. (Autoría propia)	71
Figura 41. Interfaz consultas Opción exportar. (Autoría propia)	72

Figura 42. Patrones de diseño del sistema INTELESIS 13. (Autoría propia).....	73
Figura 43. Librerías del sistema INTELESIS 13. (Autoría propia).....	74
Figura 44. Librerías de la clase DAO. (Autoría propia).....	76
Figura 45. Librerías de la clase FACADE. (Autoría propia).....	77
Figura 46. Librerías de la clase DELEGATE. (Autoría propia).....	77
Figura 47. Mensaje de validación errónea. (Autoría propia).....	81
Figura 48. Código usuario y contraseña de acceso al menú. (Autoría propia).....	81
Figura 49. Código de acceso a las diferentes interfaces del sistema INTELESIS13. (Autoría propia)	82
Figura 50. Entidad empleado con conexión a SQL. (Autoría propia).....	83
Figura 51. Validación de inserción de datos ha empleado. (Autoría propia)	83
Figura 52. Validación de actualización de datos ha empleado. (Autoría propia).....	84
Figura 53. Validación de consulta de datos a empleado. (Autoría propia)	84
Figura 54. Entidad producto con conexión a SQL. (Autoría propia)	85
Figura 55. Validación de inserción de datos a producto. (Autoría propia)	85
Figura 56. Validación de consulta de datos a producto. (Autoría propia).....	86
Figura 57. Validación de actualización de datos a producto. (Autoría propia).....	86
Figura 58. Entidad estatus con conexión a SQL. (Autoría propia).....	87
Figura 59. Validación de inserción de datos a estatus. (Autoría propia).....	87
Figura 60. Validación de consulta de datos a estatus. (Autoría propia)	88
Figura 61. Validación de actualización de datos a estatus. (Autoría propia)	88
Figura 62. Validación de inserción de datos a orden. (Autoría propia).....	89
Figura 63. Entidad ordenes con conexión a SQL. (Autoría propia).....	90
Figura 64. Validación de métodos de búsqueda de datos a órdenes. (Autoría propia)	90
Figura 65. Validación de exportación de datos a Excel. (Autoría propia)	91
Figura 66. Mensaje de validación de exportación de datos. (Autoría propia).....	91

II INTRODUCCIÓN

Durante mucho tiempo, las organizaciones han reconocido la importancia de administrar recursos clave como las personas y la materia prima. Actualmente, la información ha encontrado su lugar apropiado como recurso clave. Por consiguiente los responsables de la toma de decisiones por fin comprenden que la información no es sólo un producto derivado de las operaciones comerciales, sino que además provee impulso a las empresas y puede constituir el factor decisivo para determinar el éxito o el fracaso de un negocio (Kendall, 2005).

La competitividad de las empresas depende de su capacidad para mantener ventajas que le permitan mejorar su posición en el entorno socioeconómico. Ahora bien, un control de mermas y desperdicios aumenta la eficiencia de la producción debido a que los recursos son mejor aprovechados al reducir los costos de producción por unidad de producto.

No obstante, las entrevistas así como el cuestionario realizados previamente en el almacén de materia prima revelan que existen mermas y desperdicios dentro del proceso productivo que no han sido tratados con prioridad muchos de ellos se han considerado inherentes al proceso, debido al incorrecto control del proceso productivo.

Sin embargo, actualmente el almacén de materia prima no cuenta con ningún sistema para el tratamiento de *scrap*, por lo que se pretende crear un sistema de información lleve el control de entradas y salidas de la merma generada dentro de los departamentos, además de implementar una base de datos para la integridad de la información, es precisamente esa necesidad la que impulsa el desarrollo de este proyecto, la necesidad de generar un aumento de eficiencia y eficacia en el control del proceso productivo de los departamentos de la empresa industrial de herramientas Goni.

Una vez puesto bajo control el tratamiento de *scrap*, se está en posición de determinar el producto y el momento en que se genera una merma o desperdicio, además de definir las acciones correctivas que darán como resultado una mejora en la eficiencia del proceso productivo e inherente a esta, un aumento en la competitividad así como el control en el almacén de materia prima.

2.1 Antecedentes de la empresa.

2.1.1 Reseña histórica de industrial de herramientas Goni

Industrial de herramientas Goni es una compañía mexicana, con 30 años de dedicarse al diseño, fabricación y comercialización de equipos de aire, aplicaciones de pintura, accesorios así como consumibles para el Hojalatero, Pintor, Herrero e Carpintero. Les brindan a los usuarios herramientas y equipos de alta productividad a precios muy competitivos con todas las refacciones necesarias. Asimismo cuentan con centros de servicio a nivel nacional e internacional a través de los cuales se respaldan la garantía y el mantenimiento de los productos.

Así pues, se caracterizan por desarrollar tecnología propia, estando a la vanguardia en líneas de productos en las que son líderes del mercado. A sus distribuidores les dan un alto nivel de servicio en las entregas de producto procurando que ellos requieran el menor inventario posible. Tienen operaciones comerciales principalmente con los Estados Unidos de Norteamérica, Centroamérica y Sudamérica (Thesus de Mexico, s.f.).

Misión

- 1) Diseñar y comercializar soluciones integrales en equipos de aire y aplicación de pintura, accesorios y consumibles que satisfagan las necesidades del Hojalatero, el Pintor, el Herrero y Carpintero.
- 2) Brindarle a nuestros distribuidores el mayor rendimiento de su inversión a través de la comercialización de nuestro portafolio de productos.
- 3) Generar el desarrollo profesional y personal de nuestro equipo de trabajo en un ambiente de confianza, entusiasta y creativo de mejora continua, siempre enfocado y logrando que las cosas sucedan.
- 4) Otorgarles a los inversionistas una alta rentabilidad por su inversión con estabilidad y crecimiento.

Visión

1. Ser la solución preferida y más completa en los segmentos de mercado donde participamos, posicionados como símbolos de alta calidad y servicio en la mente del consumidor; siendo los líderes en nuestros principales mercados.
2. Ser la opción más rentable para nuestros distribuidores, con bajos requerimientos de inversión a través de entregas frecuentes, un alto porcentaje de surtimiento, una alta rotación de productos con el menor inventario posible y con eficientes soluciones de comercialización (Thesus de Mexico, s.f.).

2.1.2 Productos

La empresa industrial de herramientas GONI provee a sus clientes una gran variedad de productos tanto de línea domestica como de línea industrial además, de accesorios y refacciones de alta calidad que cumplen con los requerimientos de los clientes (Thesus de Mexico, s.f.).

Algunos de los productos más destacados son:

- Pistolas para pintar
- Compresores
- Mangueras
- Pulidoras
- Tanques para aire
- Filtro regulador
- Engrasadora neumática
- Pistola eléctrica
- Pintapad para esquinero
- Hidrolavadoras

III PLANTEAMIENTO DEL PROBLEMA

El almacén de materia prima tiene problemas de organización y control en el proceso del tratamiento de información relacionada con el *scrap* ya que no cuentan con un sistema destinado para el control de la merma.

Algunos de los principales problemas del almacén de materia prima son los siguientes:

Perdidas de material.

Las pérdidas de material son causadas por el control impreciso que se maneja, los reportes manuales que se hacen no son de gran ayuda provocando pérdidas monetarias para la empresa.

La clasificación actual de la información es imprecisa.

Debido a que no cuentan con un sistema de información para el control de la merma, se hacen en reportes bastante sencillos, perdiendo credibilidad y poniendo en riesgo la información.

Demoras en los procesos manuales.

La captura de información es demasiado lenta debido a que el usuario debe buscar manualmente los ítems de cada producto que se manda a *scrap*, generando lentitud en el proceso.

Integridad de la información.

La información no es nada segura ya que los reportes manuales que se hacen no cuentan con seguridad. Debido a esto cualquier persona con intenciones de alterar la información puede hacerlo poniendo en riesgo la producción de la empresa.

Por consiguiente, los problemas comentados anteriormente ocasionan una pérdida de producción, vulnerabilidad en la información así como una derrama económica en la empresa que a su vez origina una desestabilidad y disminución del capital constante el cual equivale a lo invertido en materia prima, maquinaria e inmuebles para llevar a cabo el proceso productivo.

IV JUSTIFICACIÓN

Con la implementación del sistema de información en el área de materia prima se evitarán las pérdidas de material en la empresa industrial de herramientas Goni, debido a que el sistema llevará el control de material desperdiciado ayudando a los jefes de cada área tomar decisiones favorables además, permitirá saber cuánto material se recicla en el proceso productivo de cada departamento.

Mientras que al llevar un control y contabilización de material desperdiciado se brindara la posibilidad de que el almacén de materia prima presente un ahorro al comprar las cantidades exactas de material para la fabricación los productos, esto debido a que con el control de *scrap* es posible reutilizar las mermas para otras piezas. Ahora bien, los reportes de la merma producida ayudarán a darle credibilidad a la información que presentará el sistema.

Por lo tanto, cuando se realicen auditorias dentro del almacén de materia prima el supervisor presentará verazmente la información y de esta manera Goni transformará su proceso productivo en una nueva ventaja competitiva para la empresa, a raíz de los cambios realizados con el sistema de información los datos relativos a todo el proceso productivo de la compañía se volverán uno de los elementos fundamentales para lograr el éxito comercial por ello es fundamental implementar así como mantener el sistema de información.

En definitiva, el sistema de información ofrecerá una importante y notable satisfacción en los usuarios que lo operan, debido a su interfaz amigable, clasificación organizada e integridad de la información para que de esta manera, los empleados puedan alcanzar, los objetivos planteados por el almacén.

V OBJETIVO

Desarrollar e Implementar un sistema de información que se encargue del tratamiento de *scrap* que se genera en el proceso productivo de la empresa.

5.1 Objetivos particulares

1. Diseñar y crear una base de datos para el sistema.
2. Crear usuarios y roles para restringir el acceso a datos.
3. Crear una interfaz amigable.
4. Generar un control en la información.
5. Hacer las pruebas para descartar cualquier error que limite la eficiencia del sistema.
6. Implementar el sistema.

VI MARCO TEÓRICO

6.1 Sistemas de información

Los sistemas de información contienen datos sobre personas, lugares y cosas importantes dentro de una organización o en el entorno que la rodea. Por consiguiente, los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar (Castro, 2010).

Los datos aisladamente pueden no contener información humanamente relevante. Solo cuando un conjunto de datos se examina conjuntamente a la luz de un enfoque, hipótesis o teoría se puede apreciar la información contenida en dichos datos. Por consiguiente, además de apoyar la toma de decisiones, la coordinación así como el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos (Laudon & Laudon, 2012).

En definitiva, la definición técnica de un sistema de información es “un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización” (Kendall, 2005).

6.1.1 Características de los sistemas de información.

Las características que debe tener un sistema de información según (Gomez & Suarez, 2009) son las siguientes:

- 1) Contener información interna y externa a la organización.
- 2) Consistencia e integración.
- 3) Asegurar una única fuente de información de gestión para todas las áreas de la empresa.
- 4) Facilitar la comprensión de la información mediante una organización adecuada de las ideas.
- 5) Proporcionar la información al ritmo que la empresa lo requiera.
- 6) Rápido acceso a la información actual e histórica.

6.1.2 Actividades básicas de un sistema de información

Un sistema de información realiza cuatro actividades básicas que son: entrada, almacenamiento, procesamiento y salida de información (Díaz, 2009), a continuación se muestran las actividades de un sistema de información (ver figura 1).

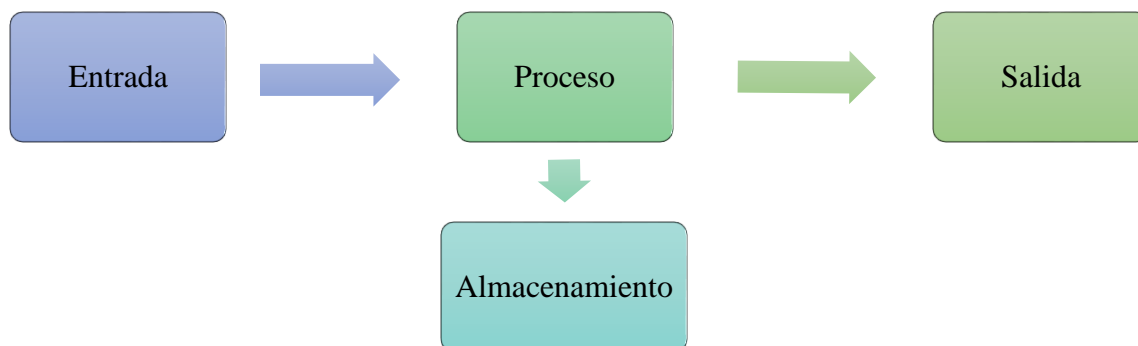


Figura 1. Actividades de un sistema de información. (Zamora, 2016).

Recuperado de <http://izamorar.com/actividades-basicas-de-un-sistema-de-informacion/>

(Zamora, 2016) Define la entrada, almacenamiento, proceso y salida de información como:

6.1.2.1 Entrada de información.

La entrada 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 informaciones que provienen o son tomados de otros sistemas o módulos.

6.1.2.2 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 sesión 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, los discos USB y los discos compactos (CD-ROM).

6.1.2.3 Procesamiento de información.

Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que estén almacenados. Esta característica de los sistemas permite la transformación de datos, fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

6.1.2.4 Salida de información.

La salida es la capacidad de un sistema de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, voz, plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o modulo.

6.1.3 Tipos de sistemas de información

Una empresa común también tiene distintos sistemas que dan soporte a las necesidades de toma de decisiones de cada uno de los principales grupos administrativos. La gerencia operacional, la gerencia de nivel medio y la gerencia de nivel superior utilizan sistemas para dar soporte a las decisiones que deben tomar para operar la compañía (Kendall, 2005).

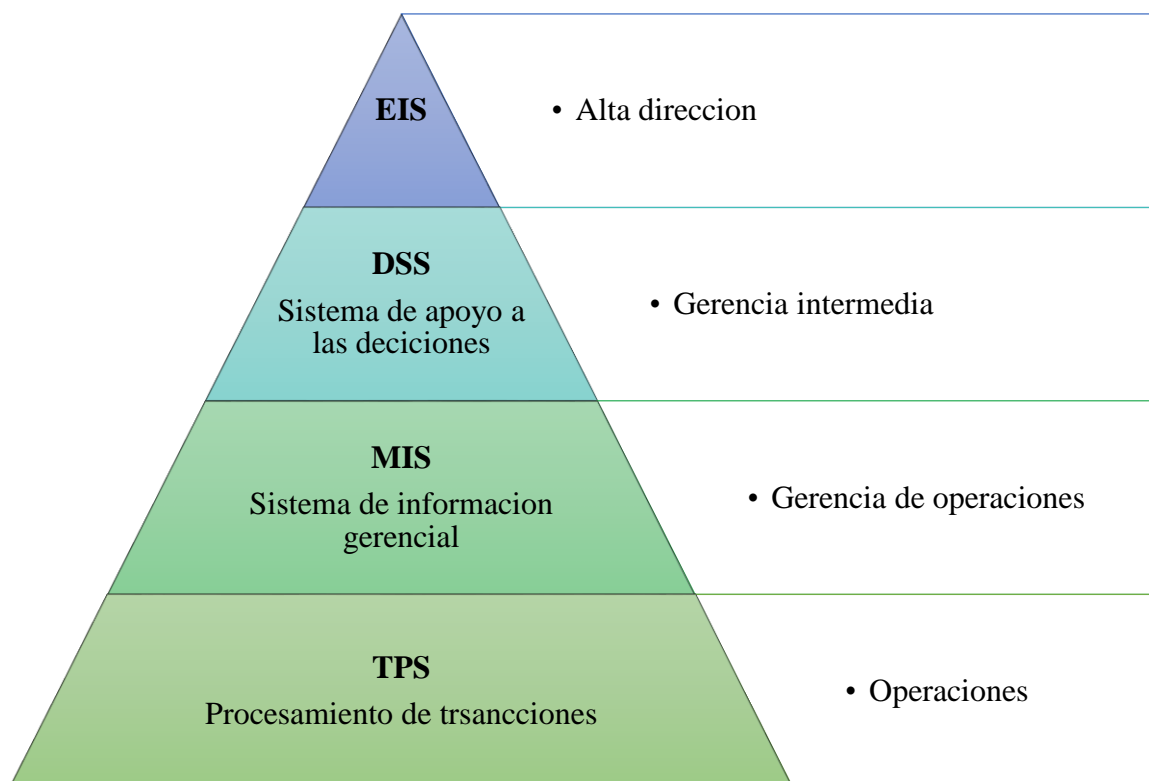


Figura 2. Tipos de sistemas de información. (Guevara, 2013).

Recuperado de <http://si-elohinaguevara.blogspot.mx/>

En la figura 2 se presentan en orden jerárquico los tipos de información que se clasifican en función al apoyo que le brindan a una organización, a continuación se muestra la tabla 1 con los tipos de sistemas de información.

Tabla 1. Tipos de sistemas de información. (Pressman, 2010).

Sistema	Funcionalidad	Ventajas/Desventajas
TPS (Sistemas de procesamiento de transacciones)	Los gerentes operacionales necesitan sistemas que lleven el registro de las actividades y transacciones elementales de la organización, como ventas, recibos, depósitos en efectivo, nóminas, decisiones de créditos y el flujo de materiales en una fábrica. Los Sistemas de Procesamiento de Transacciones (TPS) proveen este tipo de información	<ul style="list-style-type: none"> • Capaz de controlar y administrar múltiples transacciones. • Automatizan tareas operativas en la organización • Sus cálculos y procesos son poco sofisticados
MIS(Sistemas de información gerencial)	Los sistemas de información gerencial dan soporte a un espectro más amplio de tareas organizacionales, a comparación de los sistemas de procesamiento de transacciones, los sistemas de información gerencial incluyen el análisis de decisiones y la toma de decisiones.	<ul style="list-style-type: none"> • Puede manejar un enorme flujo de datos y realizar una amplia gama de procedimientos que son útiles para desarrollar. • Rápido y eficiente • Puede llegar a ser costoso
DSS(Sistema de soporte de decisiones)	Los sistemas de apoyo a decisiones son hechos de acuerdo a las características y necesidades específicas de la persona o grupo que los usa a diferencia de los sistemas de información gerencial tradicionales.	<ul style="list-style-type: none"> • Ahorro de tiempo • Mejora de la eficiencia • Mayor control organizacional • Exageración en la toma de decisiones • Asunción de relevancia
ESS(Sistema de soporte a ejecutivos)	Los sistema ESS se basa en ciertos requerimientos estratégicos de la empresa u organización, los altos ejecutivos deben manejar herramientas que le proporcionen la solución más adecuada al tomar una decisión esto es para evitar inconvenientes a futuro.	<ul style="list-style-type: none"> • Los ESS ayudan a los directivos y ejecutivos analizar, procesar y destacar tendencias. • Facilita el seguimiento del desempeño organizacional • Dificultad en mantenimiento de base de datos. • Alta capacidad de cálculo computacional.

6.2 Metodologías de desarrollo de software.

Existe un solo proceso de ingeniería de software, pero muchas perspectivas. (Bagui, 2011) Resalta que en cuanto al término se denomina ingeniería de software al proceso de especificar, diseñar, escribir, entregar y finalmente retirar el software. Los ingenieros de software frecuentemente se refieren al “ciclo de vida del software” (ver figura 3).

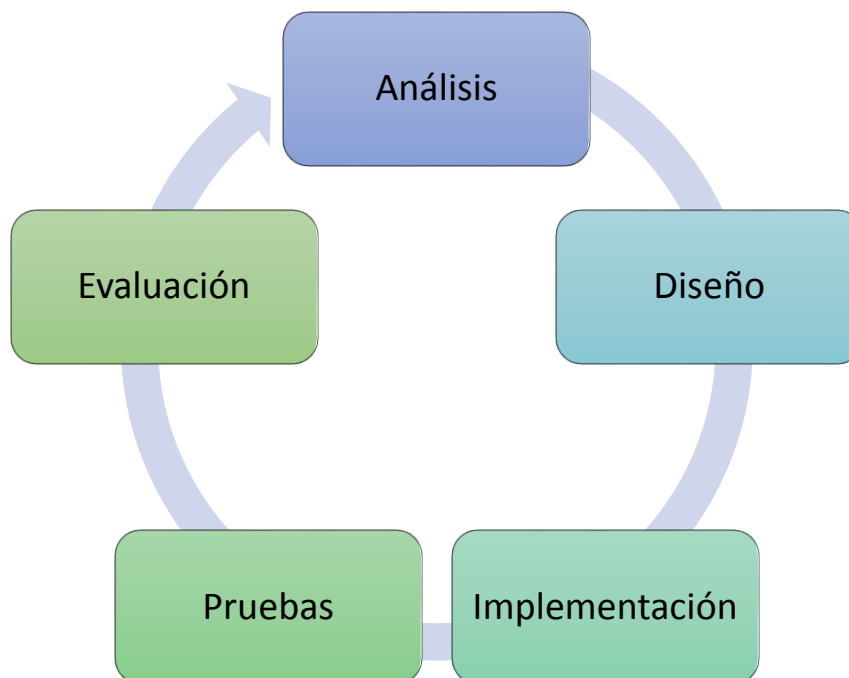


Figura 3. Ciclo de vida del sistema. (Pluke. 2011).

Recuperado de <https://commons.wikimedia.org/wiki/File:CPT-SystemLifeCycle.svg>

Cualquier sistema de información necesita un proceso para ser desarrollado correctamente. Para avanzar a través de este ciclo es necesario una serie de pasos o fases los cuales garantizaran que en un pensamiento exhaustivo precede a toda acción. Este “pensar antes de actuar” tiene a su vez distintas representaciones una de ellas es el modelo “cascada” (Bagui, 2011). Se hace tal comparación en el proceso de desarrollo de software por ir en una sola dirección y no retroceder. El modelo de cascada implica que una vez que la especificación del software está escrita y es aceptada por el usuario ya no se cambia, sino que es usada como base para el diseño.

Aquellos que participan en el ciclo de vida de desarrollo de software caen en dos ámbitos el de analista y el de usuario, el usuario es aquel que expresa sus deseos y necesidades al analista, entonces el analista desarrolla software acorde a lo expresado por el usuario.

En general la ingeniería de software adopta un enfoque sistemático y organizado, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias (Sommerville, 2005).

6.2.1 Modelos de desarrollo de software.

Un modelo para el desarrollo de software es una representación abstracta de un proceso. Asimismo cada modelo representa un proceso desde una perspectiva particular y así proporciona información parcial sobre el proceso (Sommerville, 2005), a continuación en la tabla 2 se hace la comparación de los diferentes modelos de desarrollo de software.

Tabla 2. Modelos de desarrollo de software. (Pressman, 2010).

Modelo	Enfoque	Ventajas/Desventajas
Modelo en cascada	<p>El inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.</p> <p>Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo.</p>	<ul style="list-style-type: none"> • Los proyectos raras veces siguen una evolución secuencial. • No todos los requisitos son expuestos, al principio, de forma explícita como requiere este modelo. • El cliente debe tener paciencia, ya que la aplicación sólo estará disponible en un estado muy avanzado del proyecto. • Ampliamente criticado desde el ámbito académico y la industria.
Modelo basado en prototipos	<p>Prototipos: No posee la funcionalidad total del sistema pero si condensa la idea principal del mismo, Paso a Paso crece su funcionalidad, alto grado de participación del usuario.</p>	<ul style="list-style-type: none"> • El cliente puede pensar que el prototipo es una versión acabada. • Pueden llegar a pasarse por alto la calidad del software global o el mantenimiento a largo plazo. • La clave del éxito de este modelo consiste en definir bien, desde el principio.
Modelo incremental o evolutivo	<p>El sistema no se entrega de una vez, sino que se divide y se entregan incrementos.</p> <p>Con cada incremento se entrega la parte de la funcionalidad que se ha establecido.</p> <p>Los requisitos son priorizados. Los requisitos con una más alta prioridad se incluyen en los incrementos más tempranos.</p>	<ul style="list-style-type: none"> • Los clientes no tienen que esperar hasta tener el sistema completo. El primer incremento satisface los requisitos más críticos. • Los primeros incrementos sirven como prototipo y ayudan en la tarea de detectar los posteriores requisitos. • Existe un riesgo bajo de fallar en el proyecto total. • Puede ser difícil ajustar los requisitos a los incrementos.

<p>Modelo espiral</p>	<p>Es una mejora del Modelo Basado en prototipos. Cada vuelta en la espiral representa una fase del proceso. No hay fases fijas, cada vuelta en la espiral determina las actividades a realizar.</p>	<ul style="list-style-type: none"> • Requiere comunicación permanente con el cliente por lo tanto si se cambia el contacto con el cual se realiza desarrollo es necesario que esté al tanto de lo realizado y lo pendiente, cliente debe ser gran conocedor del sistema.
----------------------------------	--	---

6.3 Base de datos

Las bases de datos tienen gran relevancia a nivel personal, pero más a nivel empresarial, y se considera una de las mayores aportaciones que ha dado la informática a las empresas. En la actualidad cualquier organización que se aprecie, por pequeña que sea, debe contar con una base de datos, pero para que sea del todo efectiva, no basta con tenerla hay que saber cómo gestionarla. (Saiz, 2013).

Una base de datos es un “Conjunto de datos comunes que se almacenan sin redundancia para ser útiles en diferentes aplicaciones” (Coronel, 2011). Asimismo las bases de datos son una “colección organizada de datos” (Deitel & Deitel, 2008)

Por lo tanto, la necesidad de tener una organización de datos en nuestra información hace necesario reemplazar la forma tradicional del tratamiento de la información, por lo que esta debe ser almacenada de forma digital en una base de datos, misma que requiere de un sistema que introduzca, guarde y administre los datos para los usuarios finales. Además, de considerar los detalles de almacenamiento físico de datos también se debe considerar la representación lógica así como sus relaciones.

6.3.1 Sistema de Gestión de Base de Datos (SGBD)

Un SGDB es una aplicación que permite a los usuarios definir, crear y mantener la base de datos así como proporcionar un acceso controlado (Ramos, 2007). Se compone de un lenguaje de definición de datos, manipulación de datos así como de consulta de datos.

6.3.1.1 Ventajas y desventajas de los SGBD:

La funcionalidad de un SGBD es fundamental ya que de ello depende la elección del usuario por consiguiente, a continuación en la tabla 3 se muestran las ventajas y desventajas de los sistemas gestores de base de datos.

Tabla 3. Ventajas y desventajas de un SGBD. (Pressman, 2010).

Ventajas de Los SGBD	Desventajas de los SGBD
<ul style="list-style-type: none">• Control sobre la redundancia de datos En los sistemas de bases de datos todos estos archivos están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, debido a que en ocasiones es necesaria para modelar las relaciones entre los datos.• Consistencia de datos Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias.• Compartición de datos En los sistemas de archivos, los archivos pertenecen a las personas o a los departamentos que los utilizan.	<ul style="list-style-type: none">• Complejidad: Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.• Costo del equipamiento adicional: Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD.• Vulnerable a los fallos: El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (<i>Backup</i>).

6.3.2 Motores de Base de Datos

En el campo del desarrollo de software existen distintos gestores de base de datos, pero no todos cumplen con nuestras expectativas por consiguiente, a continuación en la tabla 4 se muestran los diferentes gestores así como las ventajas y desventajas que poseen.

Tabla 4. Motores de base de datos. (González, 2013).

SGDB	VENTAJAS	DESVENTAJAS
MySQL	Es el más fácil y simple de usar para los DBA que empiezan.	Tiene limitaciones.
ORACLE	Es soportado por todas las plataformas y sistemas de computadora.	Para que un DBA pueda utilizarlo debe tener la licencia lo cual es muy costoso.
SQL Server	Brinda mucha estabilidad a la base de datos dándole mucha seguridad a la base de datos.	Solamente puede utilizarse en Windows no acepta otra plataforma.
DB2	Es rápido al dar las respuestas de las consultas.	Son pocos los DBA que los utilizan

6.3.3 SQL y MySQL

SQL” es un lenguaje de base de datos normalizado, que utilizan los diferentes motores de bases de datos para efectuar determinadas operaciones sobre los datos o sobre su estructura”. (Sánchez, 2016)

Entre sus características se pueden mencionar las siguientes:

- Permite el acceso a las bases de datos.
- Aprovecha el poder y la flexibilidad de los sistema relacionales, facilitando las operaciones sobre ellos.
- Se considera un lenguaje declarativo de alto nivel.
- Acepta negación y concesión de permisos, restricciones de integridad, controles a la transacción, entre otros.
- Se puede usar interactivamente, para lograr esto, sus sentencias se escriben y se llevan en líneas de comandos.

Por el contrario, “MYSQL es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para crear base de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones on-line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas” (Cobo, 2005).

Dada sus características es un SGBD muy usado en la actualidad y lo emplean empresas como Google, NASA, Yahoo, entre otras.

6.3.4 Modelo de base de datos.

Básicamente es la conjunción de entidades, campos y registros en donde las entidades representan elementos reales o abstractos, acerca de las cuales se requiere registrar información. Asimismo, campo es un grupo de caracteres que tienen un significado específico siendo su función la de definir y guardar datos así como un registro es de cada uno de esos datos (Rob & Coronel, 2007).

Existen diversos modelos de base de datos tales como el modelo entidad relación, modelo relacional, modelo de red, entre otros sin embargo, para efectos de este trabajo analizaremos el modelo relacional.

6.3.5 Modelo relacional

El modelo relacional fue desarrollado en 1970 por Edgar Frank Codd y consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Por lo tanto este modelo debe contener una estructura predefinida para el correcto desempeño el cual se mencionara enseguida:

6.3.5.1 Estructura básica de las bases de datos relacionales

El modelo relacional se ejecuta por medio de un sistema de gestión de base de datos relacional (SGBDR) y se caracteriza de los otros modelos por ser fácil de operar para los programadores y usuarios (Gomez M. d., 2013).

La estructura se compone por tablas las cuales representan las entidades y estas a su vez se relacionan con otras entidades que tienen una relación en común (ver tabla 5)

Tabla 5. Estructura de una base de datos relacional. (Sánchez, 2016).

Estructura	Descripción
Atributo	Atributos. Es cada una de las propiedades de los datos de la relación (nombre, DNI,...). Las relaciones representan conjuntos de objetos o elementos reales, cada atributo es propiedad o característica de dicho elemento.
Tuplas	Se llama tupla a cada uno de los elementos de una relación. Hablando en términos de tabla, una tupla es una fila. Las tuplas, en el modelo relacional, cumplen estas premisas: <ul style="list-style-type: none"> • Cada tupla se debe corresponder con un elemento del mundo real. • No puede haber dos tuplas iguales (con todos los valores iguales).
Dominio	Un dominio contiene todos los posibles valores que puede tomar un determinado atributo. Dos atributos distintos pueden tener el mismo dominio.
Cardinalidad	Número de tuplas de una relación, o número de filas de una tabla. Hay tablas que pueden tener una enorme cardinalidad: cientos, miles e incluso millones de filas.
Claves	<ul style="list-style-type: none"> • Clave candidata • Clave primaria • Clave alternativa

En definitiva, en un modelo relacional existen limitaciones que deben respetarse para el correcto funcionamiento de una base de datos, es por esa razón que a continuación se hace mención de los tipos de restricciones que existen dentro del modelo relacional.

6.3.5.2 Restricciones del modelo relacional

Inherentes

Son aquellas que no requieren que se establezcan de forma explícita, sino que son definidas por el propio hecho de que la base de datos sea relacional. (Alcalde, 2016) Las más importantes son:

- a) No puede haber dos filas iguales
- b) El orden de las filas no es significativo
- c) El orden de las columnas no es significativo
- d) Cada atributo sólo puede tomar un valor en la intersección entre fila y columna

Semánticas

En las semánticas el modelo relacional permite incorporar restricciones personales a las tablas. Son las más importantes y son fundamentales para que la información de la base de datos sea coherente así como eficiente, a continuación en la tabla 6 se muestran las principales restricciones semánticas.

Tabla 6. Restricciones semánticas del modelo relacional. (Sánchez, 2016).

Restricción	Descripción
Restricción de clave principal (primary key)	También llamada restricción de clave primaria. Marca uno o más atributos como identificadores de la tabla (lo cual es imprescindible en toda tabla del modelo relacional).
Restricción de unicidad (unique)	Impide que los valores de los atributos marcados de esa forma, puedan repetirse en distintas filas. Esta restricción debe indicarse en todas las claves alternativas (las primarias la tienen de forma implícita).
Obligatoriedad (not null)	Prohíbe que el atributo marcado de esta forma quede vacío (es decir impide que pueda contener el valor nulo, null) en alguna fila. También se debe de indicar en las columnas que son clave alternativa.
Clave alternativa (alternate key)	Salvo en algunas bases de datos (muy pocas), no es posible marcar esta restricción, que indicaría sobre una o más columnas, el hecho de que no pueden repetir valores ni quedar vacías en cada fila.

<p>Integridad referencial (foreign key)</p>	<p>Sirve para indicar una clave externa (también llamada secundaria o foránea) sobre uno o más atributos. Los atributos marcados de esta forma sólo podrán contener valores que estén relacionados con la clave principal de la tabla que relacionan (llamada tabla principal). Dichos atributos sí podrán contener valores nulos.</p>
<p>Regla de validación (check)</p>	<p>Condición lógica que debe de cumplir un dato concreto en una o más columnas (a las que se aplicará esta restricción) para darlo por válido. Por ejemplo, podríamos restringir la columna llamada sueldo para que siempre acepte valores mayores de 1000; no se permitiría entonces indicar sueldos menores de 1000.</p>
<p>Disparadores o triggers</p>	<p>Son restricciones más complejas presentes en sistemas avanzados de bases de datos. Se trata de código almacenado en la base de datos, cuyas instrucciones se ejecutan automáticamente cuando ocurre un determinado evento (añadir una fila, modificar filas, iniciar sesión por parte del usuario).</p>

El modelo relacional posee tanto ventajas como desventajas las cuales lo definen como un sistema eficiente o en su defecto ineficiente. Por último se presentan los puntos a favor y en contra que (Cristian, 2010) define como:

Ventajas del modelo relacional

- Se pueden hacer modificaciones en la estructura de la base de datos sin afectar el acceso a los datos.
- Es más fácil de administrar y diseñar
- Tiene una potente y flexible capacidad de consulta
- El SGBDR realiza tareas complejas que facilitan el diseño de la parte lógica de la base de datos.

Desventajas del modelo relacional

- Es relativamente costoso implementarlo
- Al ser fácil de usar, los usuarios llevan a cabo reportes sin haber diseñado una base de datos adecuada.

6.3.6 Normalización

Para mantener una base de datos sin problemas de redundancia, actualización de datos así como integridad en la información es fundamental aplicar un procedimiento de reglas llamado proceso de normalización el cual sirve para ayudar a desarrollar un esquema que minimice los problemas de lógica (Silberschatz & Korth, 2006).

En el proceso de normalización cada regla está basada en la que le antecede, de esta manera el proceso se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos.

Para llevar a cabo la normalización existen reglas que (Gomez M. d., 2013) define como:

Primera Forma Normal (1FN)

- Una tabla está en Primera Forma Normal si:
- Todos los atributos son atómicos.
- La tabla contiene una clave primaria.
- La llave primaria no contiene atributos nulos.
- No posee ciclos repetitivos. No debe de existir variación en el número de columnas.

Segunda Forma Normal (2FN)

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales.

Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Forma normal de Boyce-Codd (FNBC)

La tabla se encuentra en FNBC si cada determinante, atributo que determina completamente a otro, es clave candidata. Deberá registrarse de forma anillada ante la presencia de un intervalo seguido de una formalización perpetua, es decir las variantes creadas, en una tabla no se llegaran a mostrar, si las ya planificadas, dejan de existir.

Cuarta Forma Normal (4FN)

Una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una súper-clave que, X es o una clave candidata o un conjunto de claves primarias.

Quinta Forma Normal (5FN)

Una tabla se encuentra en 5FN si: La tabla está en 4FN No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que está en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas.

6.4 Lenguaje de programación Java

Con el surgimiento de nuevos y modernos lenguajes, Java dejó de ser tan popular sin embargo, gracias a su potencia, muchas empresas siguen creyendo en este lenguaje y sus beneficios. Tanto es así que no solo es uno de los lenguajes más demandados, sino también de los más utilizados. (Patrizio, 2015).

En conclusión, el lenguaje de programación java es “un lenguaje completo que permite realizar todo tipo de aplicaciones” (Sánchez, 2009).

6.4.1 Historia de java

En diciembre de 1990 Patrick Naughton, ingeniero de Sun Microsystems, reclutó a varios colegas entre ellos James Gosling y Mike Sheridan para trabajar sobre un nuevo proyecto conocido como "El proyecto verde". Con la ayuda de otros ingenieros, empezaron a trabajar en una pequeña oficina en Sand Hill Road en Menlo Park, California. Y así interrumpió todas las comunicaciones regulares con Sun y trabajó sin descanso durante 18 meses. Intentaban desarrollar una nueva tecnología para programar la siguiente generación de dispositivos inteligentes, en los que Sun veía un campo nuevo a explorar. Crear un lenguaje de programación fácil de aprender y de usar.

En un principio se consideraba C++ como lenguaje a utilizar, pero tanto Gosling como Bill Joy lo encontraron inadecuado. Gosling intentó primero extender y modificar C++ resultando el lenguaje C++ porque se añadían y eliminaban características a C++, pero lo abandonó para crear un nuevo lenguaje desde cero al que llamó Oak (roble en inglés, según la versión más aceptada, por el roble que veía a través de la ventana de su despacho).

El resultado fue un lenguaje que tenía similitudes con C, C++ y Objective C y que no estaba ligado a un tipo de CPU concreta. (Belmonte, 2005).

6.4.2 Características de java.

6.4.2.1 *Compilación tradicional*

En el mundo de la programación siempre se ha hablado de lenguajes compilados y de lenguajes interpretados. En el segundo caso, un programa intérprete se encarga de traducir cada línea al código máquina correspondiente. Los lenguajes interpretados a partir de los setenta se han dejado de usar porque no eran los apropiados para conseguir código eficiente.

Por el contrario, los lenguajes compilados producen código máquina analizando todas las líneas de código en conjunto. Los compiladores buscan el mejor código máquina posible. El resultado del proceso de compilación es un archivo ejecutable.

Un archivo ejecutable es un programa que se puede lanzar directamente en el sistema operativo; en el caso de Windows o Linux simplemente con hacer doble clic sobre el archivo, se ejecutan sus instrucciones. La ventaja es que los programas ejecutables no necesitan compilarse de nuevo, son programas terminados.

Sin embargo, el problema es que los sistemas operativos utilizan diferentes tipos de archivos ejecutables: es decir, un archivo ejecutable en Linux no sería compatible con Windows (Sánchez, 2009), en la siguiente figura se muestra el proceso de compilación de ambos sistemas operativos.

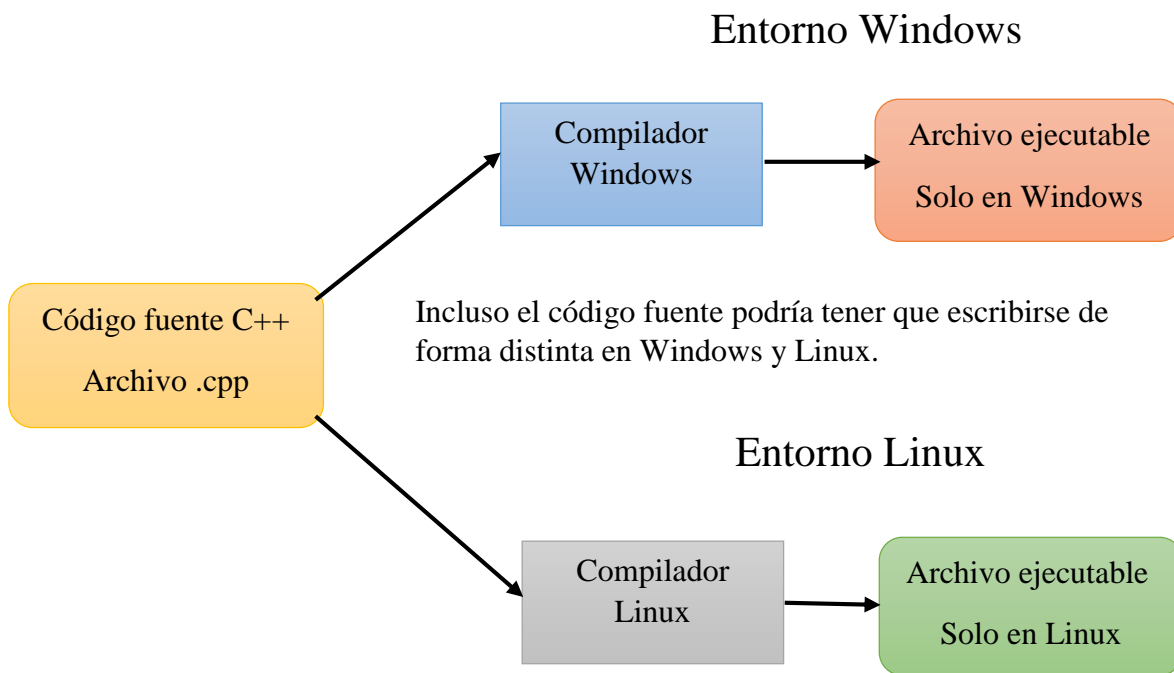


Figura 4. Compilación de un programa C en Windows y Linux. (Sánchez 2009).

Recuperado de <https://jorgesanchez.net/>

6.4.2.2 La compilación en java

En Java el código no se traduce a código ejecutable. En Java el proceso se conoce como pre compilación y sirve para producir un archivo (de extensión *class*) que contiene

código que no es directamente ejecutable (no es código Java). Es un código intermedio llamado *bytecode* (también se le llama *J-code*).

Al no ser ejecutable, el archivo *class* no puede ejecutarse directamente con un doble clic en el sistema. El *bytecode* tiene que ser interpretado (es decir, traducido línea a línea) por una aplicación conocida como la máquina virtual de Java (JVM). Hoy se conoce como JRE (*Java Runtime Environment*, entorno de ejecución de Java).

La gran ventaja es que el entorno de ejecución de Java lo fabrica *Sun* para todas las plataformas; lo que significa que un archivo *class* se puede ejecutar en cualquier ordenador o máquina que incorpore el JRE. Sólo hay un problema, si programamos utilizando por ejemplo la versión 1.6 de Java, el ordenador en el que queramos ejecutar el programa deberá incorporar el JRE al menos de la versión 1.6.

El JRE o la máquina virtual de Java son unos programas muy pequeños y se distribuyen gratuitamente para prácticamente todos los sistemas operativos.

Por lo tanto, a la forma de producir código final de Java se la llama JIT (*Just In Time*, justo en el momento) ya que el código ejecutable se produce sólo en el instante de ejecución del programa. Es decir, no hay en ningún momento código ejecutable. (Sanchez, Lenguaje Java Fundamentos de Programacion, 2009). La figura muestra el proceso de compilación de un archivo java para ambos sistemas operativos.

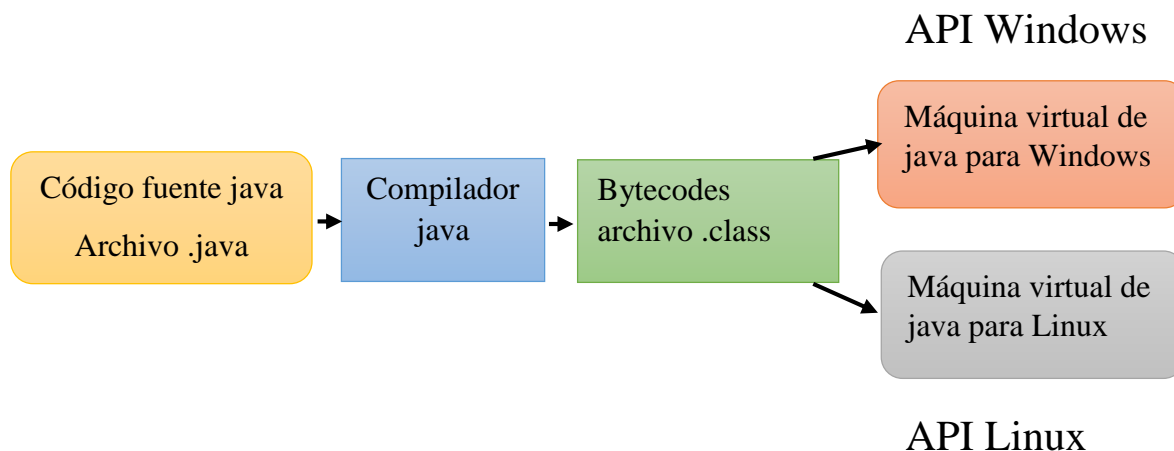


Figura 5. Proceso de compilación de un programa Java. (Sánchez 2009).

Recuperado de <https://jorgesanchez.net/>

6.4.2.3 Seguridad en el lenguaje de programación java

En el lenguaje de programación java existen reglas de seguridad que pueden hacer de nuestras aplicaciones más integras. (Sánchez, 2009) menciona lo siguiente:

Al interpretar el código, el JRE puede delimitar las operaciones peligrosas, con lo cual la seguridad es fácilmente controlable. Además, Java elimina las instrucciones dependientes de la máquina y los punteros que generaban terribles errores en C y la posibilidad de generar programas para atacar sistemas. Tampoco se permite el acceso directo a la memoria y recursos del ordenador.

La primera línea de seguridad de Java es un verificador del *bytecode* que permite comprobar que el comportamiento del código es correcto y que sigue las reglas del lenguaje Java, normalmente los compiladores de Java no pueden generar código que se salte las reglas de seguridad de Java. Pero un programador malévolo podría generar artificialmente *bytecode* que se salte las reglas. El verificador intenta eliminar esta posibilidad, es decir, hay un segundo paso que verifica la seguridad del código que es el verificador de clase que es el programa que proporciona las clases necesarias al código. Lo que hace es asegurarse que las clases que se cargan son realmente las del sistema original de Java y no clases creadas reemplazadas artificialmente.

Finalmente hay un administrador de seguridad que es un programa configurable que permite al usuario indicar niveles de seguridad a su sistema para todos los programas de Java.

Sin embargo, hay una forma de seguridad relacionada con la confianza. Esto se basa en saber que el código Java procede de un sitio de confianza y no de una fuente no identificada, se consigue gracias a que en Java se permite añadir firmas digitales al código para verificar la autoría del mismo.

6.4.3 Plataformas

Actualmente hay tres ediciones de Java. Cada una de ellas se corresponde con una plataforma que incluye una serie de funciones, paquetes y elementos del lenguaje es decir la API “*Application Program Interface*” (Berzal, 2008), a continuación en la tabla se muestran las tres ediciones de java.

Tabla 7. Plataformas de java. (Sánchez, 2009).

Java SE	Java EE	Java ME
<ul style="list-style-type: none">• Java Standard <i>Edition</i>. Antes se la conocía como J2SE. Permite escribir código Java relacionado con la creación de aplicaciones y <i>applets</i> en lenguaje Java común. Es decir, es el Java normal. La última versión del kit de desarrollo de aplicaciones en esta plataforma es la JSE 1.6.14.	<ul style="list-style-type: none">• Java Enterprise <i>Edition</i>. Todavía conocida como J2EE. Pensada para la creación de aplicaciones Java empresariales y del lado del servidor. Su última versión es la 1.4	<ul style="list-style-type: none">• Java Mobile <i>Edition</i>. También conocida como J2ME. Pensada para la creación de aplicaciones Java para dispositivos móviles.

6.4.4 JDBC API.

Para establecer una conexión con la base datos es necesario la implementación del *Java database Connectivity* o mejor conocido como JDBC. Esta API es un conjunto de clases e interfaces que están escritas en lenguaje java. Por consiguiente hace fácil el envío de instrucciones en SQL para base de datos relacionales además, hace posible el acceso a archivos que contienen datos tabulados, su valor radica en que las aplicaciones tienen acceso de manera virtual a cualquier fuente de información y en cualquier plataforma con una JVM (Fisher, Ellis, & Bruce, 2005).

Básicamente existen tres acciones que un JDBC realiza:

- 1) Establece una conexión con la fuente de información.
- 2) Enviar *queries* a la fuente de información.
- 3) Procesar los resultados.

La figura muestra el proceso para el correcto funcionamiento de un API JDBC.

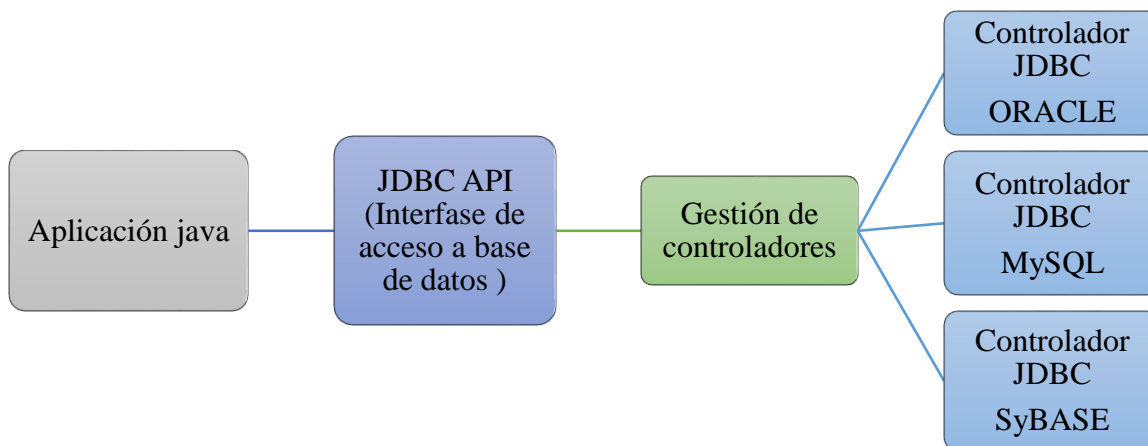


Figura 6. JDBC API. (Valdez, 2011).

Recuperado de <https://avaldes.com/connecting-to-sqlserver-using-jdbc/>

Como se muestra en la figura 6 el JDBC permite conectarse a cualquier tipo de base de datos con el mismo código en la aplicación programada en java. O sea, en un modelo de dos-tercios en el que la aplicación java se comunica directamente con la fuente de datos, se requiere un driver JDBC que se pueda comunicar con esa fuente de datos. Aun así los comandos del usuario son entregados a la base de datos y los resultados son enviados de regreso.

JDBC Statements

Un *statement* es lo que envían las consultas al DBMS (Fisher, Ellis, & Bruce, 2005). Lo que se tiene que hacer es crear un objeto *Statement* y ejecutarlo, enviando las consultas o inserciones al método *execute* apropiado. Es decir para una consulta SELECT se debe usar el método *executequery* y la creación y modificación de tablas debe hacerse con el método *executeupdate*.

Por ejemplo, después de crear el objeto se debe de ejecutar el método para llevar a cabo las consultas/modificaciones. Los resultados de las consultas realizadas son devueltos por java en un objeto *ResultSet*.

6.5 Patrones de diseño

El poder identificar un problema una y otra vez, sea cual sea el proyecto y aplicar el patrón correcto es algo que se aprende con el buen conocimiento de los patrones en definitiva, un patrón de diseño nos ayuda tener un código entendible para trabajar en el desarrollo de una aplicación.

Algunos patrones de diseño para el desarrollo de una API son los siguientes:

6.5.1 Patrón de diseño DTO

Objeto de Transferencia de Datos, es un objeto que transporta datos entre procesos. La motivación de su uso tiene relación con el hecho que la comunicación entre procesos es usualmente realizada mediante interfaces, donde cada llamada es una operación costosa. Como la mayor parte del costo de cada llamada está relacionado con el tiempo round-trip entre el cliente y servidor, una forma de reducir el número de llamadas es usando un objeto (el DTO) que agrega los datos que habrían sido transferidos por cada llamada, pero que son entregados en una sola invocación (MSDN, 2010).

6.5.2 Patrón de diseño DAO

En software de computadores, un Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object (Sun Microsystems, 2007).

Ventajas de DAO

La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula (Castellanos, 2013).

Desventajas de DAO

Por el contrario la flexibilidad tiene un precio. Cuando se añaden DAOs a una aplicación, la complejidad adicional de usar otra capa de persistencia incrementa la cantidad de código ejecutado durante tiempo de ejecución. La configuración de las capas de persistencia requiere en la mayoría de los casos mucho trabajo. Las aplicaciones críticas con el rendimiento no deberían usar DAOs (Castellanos, 2013).

6.5.3 Patrón de diseño FACADE

(Bellás, 2006) Define el patrón *facade* como una interfaz de alto nivel que simplifica el uso de un subsistema, a continuación se hace mención de algunas características de la capa modelo de un *facade*:

- 1) Una fachada representa un conjunto de casos de uso lógicamente relacionados
- 2) Cada operación corresponde a un caso de uso
- 3) La declaración de las operaciones oculta la tecnología usada en su implementación
- 4) Puede usarse una factoría para crear instancias
- 5) En una aplicación real normalmente hay más de una fachada

6.5.4 Patrón de diseño DELEGATE

El Business Delegate actúa como una abstracción de negocio del lado del cliente; proporciona una abstracción para la implementación de los servicios del negocio. Es decir, utilizando Business Delegate se reduce el acoplamiento entre los clientes de la capa de presentación y los servicios de negocio del sistema. Dependiendo de la estrategia de implementación, Business Delegate podría aislar a los clientes de la posible volatilidad en la implementación del API de los servicios de negocio. Potencialmente, esto reduce el número de cambios que se deben hacer en el código de cliente de la capa de presentación cuando cambie el API del servicio de negocio o su implementación subyacente (Vp, 2013).

No obstante, el objetivo de un *delegate* es Facilitar el acceso de la capa cliente a la capa EJB, ocultando el API de EJB. Por lo tanto un objeto que reside en la capa de presentación y en beneficio de los otros componentes de la capa de presentación llama a métodos remotos en los objetos de la capa de negocios. (Heno, 2013)

Algunas ventajas importantes del patrón *delegate* son las siguientes:

- Mejora el mantenimiento.
- Permite separación de roles: desarrolladores de la capa cliente y desarrolladores de la capa EJB.
- Puede proporcionar caché para mejorar la eficiencia. (Gomez M. , 2013)

Como se menciona anteriormente el uso de patrones es fundamental para un código entendible y un preciso desarrollo de una aplicación pero otra cosa fundamental dentro de una API es el framework.

6.5.5 Framework

(Bellas, 2006). Define un framework como:

- 1) Conjunto de clases que implementan de manera parcial la arquitectura de una aplicación (o parte de la arquitectura)
- 2) Implementan una colección de patrones
- 3) Para completar la arquitectura, el desarrollador tiene que implementar algunos interfaces y/o extender algunas clases abstractas
- 4) El framework invoca al código de la aplicación
- 5) También suelen tener una parte que actúa como librería

6.6 Ubuntu

Como anteriormente se comentaba el uso de patrones de diseño es primordial para el desarrollo de una aplicación pero también es importante tener un sistema operativo potente donde desarrollarlo.

El sistema operativo *Ubuntu* está basado en la distribución *Debian*, pero a diferencia de esta, se saca una versión cada seis meses, aproximadamente, y se mantiene actualizada durante 18 meses después de su lanzamiento, otra diferencia es el empleo del escritorio Gnome como escritorio principal, esta distribución pretende no romper la compatibilidad con *debian*, de modo que puedan intercambiarse paquetes sin problemas por consiguiente, esto la hace una distribución muy orientada al escritorio, pero con bastante estabilidad. Ubuntu comparte las ventajas de *debian* (exceptuando que tiene ligeramente menos paquetes, y no están tan probados), añadiéndole el hecho de tener una distribución bastante actualizada (Grant, 2006).

A continuación se mencionan algunas ventajas y desventajas del sistema operativo Ubuntu:

6.6.1 Ventajas de Ubuntu

- 1) Es totalmente Gratuito y aunque posea versiones de paga (con soporte técnico) es aún más barato que comprar Windows.
- 2) Las distribuciones importantes tienen muchos programas muy útiles y que se pueden encontrar muy fácilmente en internet.
- 3) Un punto muy importante es la seguridad, los Hackers y/o creadores de virus rara vez atacan a Software de Linux.
- 4) Como se puede observar en muchas webs, existe infinidad de Información técnica que servirá de ayuda.
- 5) Se lleva bien en el arranque en conjunto con Windows.
- 6) Carga y realiza tareas con mayor eficiencia que Windows.
- 7) La constante actualización y nuevas versiones es asombrosa. Existen infinidades de Distribuciones de Linux. (Denis, 2013)

6.6.2 Desventajas de Ubuntu.

- 1) El origen técnico de Linux aún se percibe; antes de que puedas creerlo, estarás abriendo una “Terminal” de Linux y escribiendo comandos. Algo que nunca harías con Windows.
- 2) No es muy común su uso en Compañías, por lo que generalmente se usa para trabajar desde Casa, así que debes hacer ajustes laboriosos.
- 3) A la hora de trabajar con documentos de Windows complejos, se podría convertir en una tareas difícil o casi imposible debido a la poca compatibilidad para importar desde Windows para Linux.
- 4) Instalar controladores de Hardware y programas resulta ser más complicado que en Windows. Esto debido a que las empresas creadoras de controladores crean sus productos en base a Windows, el sistema operativo más usado a nivel mundial.
- 5) Muchos juegos no se ejecutan en Linux así que olvídate de grandes Gráficos y poner a trabajar tu tarjeta de video al máximo. Claro existen sus excepciones. (Denis, 2013)

VII METODOLOGÍA

7.1 Análisis y diseño del sistema

En esta etapa fue muy importante la interacción que hubo con el usuario para recolectar la información necesaria para el desarrollo del sistema, para esto se aplicaron varias técnicas de recopilación de datos las cuales se mencionan a continuación:

7.1.1 Cuestionario de requerimientos de software

El usuario como antes se menciona es aquel que solicita al analista el sistema de información y proporciona las especificaciones del mismo, las cuales son sus deseos o necesidades (Bagui, 2011). Hay que considerar que el usuario muchas veces no tiene un vasto conocimiento del área tecnológica de la información por lo cual identificar estas necesidades o deseos será una ardua tarea para el analista, ya que el usuario procurara describir lo que hace y lo que le gustaría que hiciera el sistema con sus propias palabras.

Por lo mencionado anteriormente se elaboró un cuestionario para la recolección de los requerimientos del software a desarrollar (ver en anexo A).

7.1.2 Control de mermas en el almacén de materia prima.

INDUSTRIAL DE HERRAMIENTAS, S.A. DE C.V.

Catálogo de multialmacén

Productos:	Todos		Almacén:	2	
Producto	Descripción	Ctrl. almacén	Línea	Exist. almacén	Exist. Inventario
Almacén: 2	Materia Prima				
13011	CAJA P/PIST 29 EXP P/ECUADOR		N3NA	991.00 ✓	991.00
140011	ROLLO DE 100MTS MANG DE HULE DE 3/8"		N3NA	44.00 ✓	44.00
140021	FERULA DE 14MM		N3NA	4,247.00 ✓	4,247.00
140031	FERULA DE 17 MM		N3NA	3,303.00 ✓	3,303.00
140071	FERULA 12.0 MM.		N3NA	7,651.00 ✓	7,651.00
140081	FERULA 13 MM		N3NA	1,693.00 ✓	1,693.00
158	TUBO DE FLUIDO DE ALUMINIO PARA P.B.		T3NA	19,337.00 ✓	19,345.00
158-A	TUBO DE ALUMINIO ANODIZADO MATE		E3AO	1,424.00 ✓	1,424.00
160011	SOLERA DE ALUMINIO DE 3/4" X 1/8" x 3.66		N2NA	40.00 ✓	40.00
180101	BARRA DE ACERO REDONDO 3/4" 12L14		N2NA	352.00 ✓	352.00
180111	BARRA DE ACERO REDONDO 1" 1018		N2NA	330.00 ✓	330.00
180151	MT DE TUBO FLEXIBLE DE COBRE 3/8"		N2NA	0.01 ✓	0.02
180261	BARRA RED. DE AL 6063-T5 DE		N2NA	39.00 ✓	39.00
180271	BARRA RED. DE AL 6063-T5 DE		N2NA	24.50 ✓	24.50
180381	MT TUBO DE ALU CED.40 6063T-6 DE 3/4"		D2NA	293.00 ✓	293.00
1P000	CTO. GENERAL DE CARTÓN MANGUERA			2,581.00 ✓	2,561.00
1P001	CENTRO GENERAL PARA MANGUERA PVC			3,148.00 ✓	3,148.00
210043	JUEGO DE ACCESORIOS PARA PISTOLA 26		E1NA	1,640.00 ✓	1,640.00
210063	JGO. DE ACCESORIOS PARA PISTOLA 29		E1NA	555.00 ✓	555.00
210083	JUEGO DE ACCESORIOS PARA PISTOLA 29		E1NA	256.00 ✓	256.00
210073	JUEGO DE ACCESORIOS PARA PISTOLA		E1NA	235.00 ✓	235.00
210083	JUEGO DE ACCESORIOS PARA PISTOLA		E1NA	500.00 ✓	500.00
210103	JUEGO DE ACCESORIOS PARA PISTOLA		E1NA	979.00 ✓	979.00
210143	JUEGO DE ACCESORIOS PARA PISTOLA		E1NA	11.00 ✓	11.00
210203-CA	TOBERA DOBLE ABANICO GONI		E3CA	147.00 ✓	147.00
210213-RA	BOQUILLA DOBLE ABANICO GONI		S2RA	1,309.00 ✓	1,309.00
210243-LA	ANILLO DOBLE ABANICO LATONADO		E3GA	624.00 ✓	624.00
210253-MA	CABEZA PISTOLA DOBLE ABANICO		T2MA	1,201.00 ✓	1,201.00
210253-TR	CABEZA PISTOLA DOBLE ABANICO		I2IN	842.00 ✓	842.00
210262-AD	VASO DE BAJA PRESION ANODIZADO ORO		V3AO	139.00 ✓	139.00
210272-NI	CONEXIÓN DE ACERO NIQUELADA		E3NI	1,589.00 ✓	1,589.00
210293-NI	BOQUILLA PUNTO P/3350 NIQUELADA		S3NI	797.00 ✓	798.00
210302-LA	TORNILLO REG B-P LATONADO		E3LT	900.00 ✓	900.00
210323-MA	CABEZA 26VB MAQUINADA BROTIMEX			36.00 ✓	36.00
210333-MA	CABEZA 26 EVERLAST MAQUINADA			550.00 ✓	550.00
210333-TR	CABEZA 26 EVERLAST INYEC Y TROD			393.00 ✓	393.00
210343-MA	CABEZA 28 EVERLAST MAQUINADA			22.00 ✓	22.00
210352-ZN	TOBERA CUADRADA ZINC-NIQUEL			6,289.00 ✓	6,289.00
220003IN	CAB P PIST D GRAV BROTIMEX		N3NA	400.00 ✓	400.00
220003MA	CABEZA P PIST D GRAV BROTIMEX MAQUIN		M2MA	11.00 ✓	11.00
220011	OPRESOR DE 3/8 x 1/2" 16 UNC		N3NA	1,289.00 ✓	1,289.00
220013-GA	CONECTOR P VASO D RETOO		E3GA	1.00 ✓	1.51
220023-CA	TOBERA HEXAGONAL CARBONITURADA		E3CA	2,346.00 ✓	2,346.00
220033-GA	ANILLO PARA PISTOLA MOD.77		E3GA	1,000.00 ✓	1,000.00
220043-MA	AGUJA PARA PISTOLA MOD. 77 MAQUINADA		T2MA	10.00 ✓	10.00
220053-BA	BOQUILLA 77 BARRENADA		S3BA	344.00 ✓	344.00
220053-MA	BOQUILLA 77 MAQUINADA		T2MA	33.00 ✓	33.00

Usuario:ADMINISTRADOR

Fecha y hora:04/01/2017 14:41

Pág. 1

Figura 7. Catálogo de multi-almacén. (Industrial de herramientas Goni)

La figura muestra el catálogo de productos manejados por el almacén de materia prima, la mayoría de los productos corren el riesgo de padecer alguna imperfección por lo cual son candidatos para mermas.

Actualmente, el manejo de las mermas se realiza manualmente lo cual dificulta un control preciso así como es vulnerable al robo o alteración de información, a continuación se muestra un reporte donde el almacén de materia prima lleva el control de *scrap*.

INDUSTRIAL DE HERRAMIENTAS, S.A. DE C.V.

REPORTE DIARIO DE SCRAP DE PRODUCCIÓN				
SUPERVISOR:		FECHA:		AREA:
TURNO:				
MAQUINADO	DEFECTO	NOMB. OPERARIO	No. PZAS.	OBSERVACIONES
TOTAL:				repscrap

Figura 8. Reporte de SCRAP de producción. (Industrial de herramientas goni).

La figura fue proporcionada por el personal de materia prima con el fin de conocer el manejo actual de las mermas y en base a esto saber cuáles son las deficiencias de un control deficiente dentro del almacén, para poder implementar un sistema eficiente.

7.1.3 Organización y depuración de la información

7.1.3.1 Elección de la metodología, herramientas y técnicas para el desarrollo del sistema.

- 1 *Ubuntu*, Se eligió Ubuntu como sistema operativo debido a su buena funcionalidad y rendimiento para el desarrollo de software. La única desventaja es que este sistema operativo no es tan difundido y conocido como lo es, Microsoft Windows.
- 2 *Modelo incremental o evolutivo*, El modelo seleccionado para el desarrollo del sistema fue el modelo incremental o evolutivo debido a que el cliente no tiene que esperar a que el sistema esté totalmente terminado sino que con el primer incremento satisface los requisitos más críticos, además los primeros incrementos sirven como prototipo y ayudan en la tarea de detectar los posteriores requisitos.
- 3 *Sistema de información DSS*, El tipo de sistema bajo el cual se desarrolló la aplicación fue el DSS debido a que se basan en la necesidades especificadas por persona o un grupo de personas en este caso todo el almacén de materia prima de la empresa, además de los elemento fundamentales con los que cuenta este sistema como la gestión de base de datos para llevar todo el tratamiento de la información del *scrap*.
- 4 *MySQL*, El sistema gestor de base de datos seleccionado para desarrollar la API fue MySQL debido a la facilidad que tiene para el manejo de la información, velocidad al realizar operaciones así como seguridad de los datos.
- 5 *SDK 8*, Otro requerimiento fundamental es el kit de herramientas mejor conocido como JDK o SDK para la ejecución del sistema, instalaremos la versión 8.
- 6 *JDBC*, Fundamental Para la conexión entre la base de datos y la API.
- 7 *JXL API*, También conocido como Java Excel API, es la API más ampliamente utilizado para la ejecución de las pruebas basadas en los datos de selenio, lo que permite a los usuarios leer, escribir, crear y modificar hojas en un libro de Excel (.XLS) en tiempo de ejecución. (Selenium, 2016)

7.1.4 Diagrama de casos de uso

Basados en los antecedentes del capítulo 4, en donde se trató el panorama general de la problemática, se realizó una recopilación de información y datos para hacer un análisis de la información que deberá manejar el sistema de información. La información que se corresponderá manejar es la siguiente:

Actores.

Después de realizar entrevistas así como la aplicación del cuestionario en el almacén de materia prima se definieron los siguientes actores:

1. *Administrador*

El administrador es la persona encargada del control del sistema es decir, puede realizar adecuaciones y correcciones, dependiendo de las necesidades de los usuarios así como actuar en caso de cualquier emergencia relacionada con el sistema.

2. *Supervisora*

Persona encargada de la mayor parte de la acciones del sistema, juega un papel importante ya que la información proporcionada por los departamentos es recibida por ella y tiene la obligación de ingresarla al sistema.

3. *Empleado*

Persona que solo puede consultar datos, generar reportes de la base de datos solo en caso de que el supervisor lo autoriza o por ausencia.

Casos de uso

Después de realizar un análisis en el área de materia prima se diseñó el siguiente diagrama de caso de uso en el cual se pueden apreciar los tres actores manipulando el sistema de información, con la diferencia de que cada uno posee permisos distintos es decir, el administrador gestiona todo el sistema mientras que, el supervisor tiene accesos a todas las funciones del sistema y por último, el empleado autorizado consultara datos e impresión de reportes, a continuación en la figura 7 se muestra el diagrama realizado.

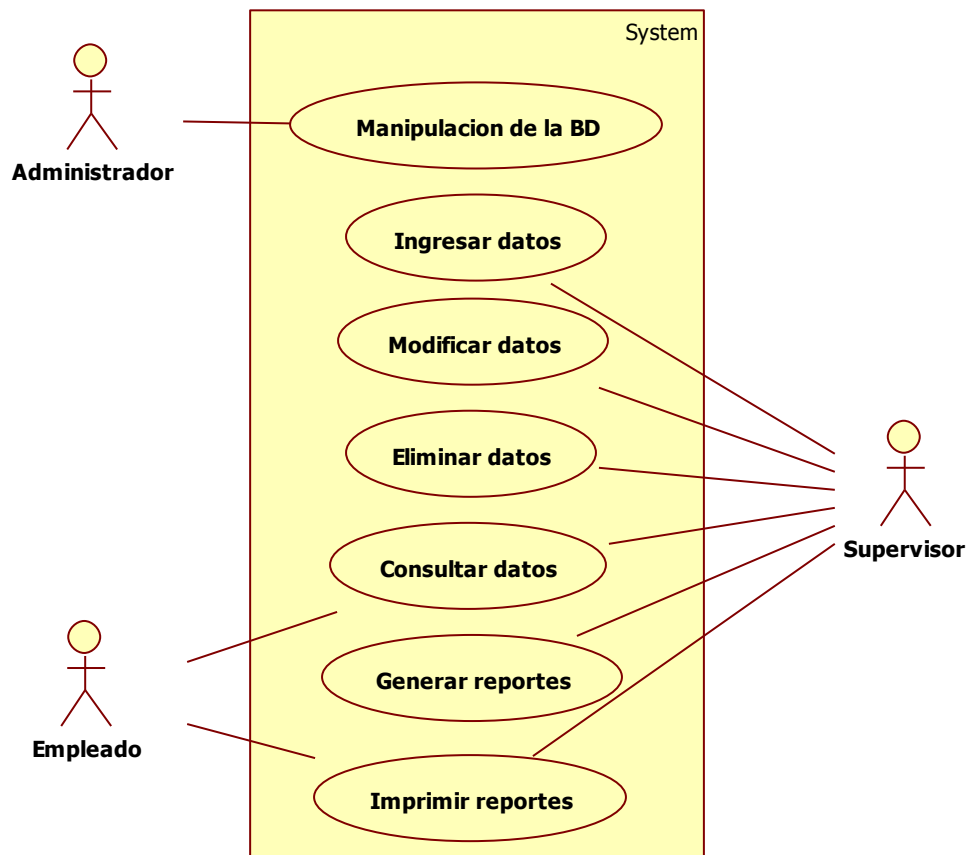


Figura 9. Diagrama de casos de uso de INTELESIS 13. (Autoría propia).

7.1.5 Diagramas de actividades

Una vez finalizado el diagrama de caso de uso, se diseñaron los diagramas de actividades con el fin de conocer el proceso que se manejará en el sistema de información así como la interacción que tendrá el usuario. A continuación se definirán los procesos de cada uno de los diagramas de actividades.

Diagrama de manipulación de la base de datos

Consiste en la manipulación que tendrá el administrador sobre la base de datos y todos los permisos así como funciones que podrá realizar. Por consiguiente, algunas de las funciones que podrá realizar administrador es el poder dar de alta nuevos usuarios o en su defecto eliminarlos, también podrá realizar las copias de seguridad, actualización de estadísticas y limpieza del historial de registros que se almacenan en la base de datos, a continuación en la figura se muestra la manipulación que tendrá el administrador sobre el sistema.

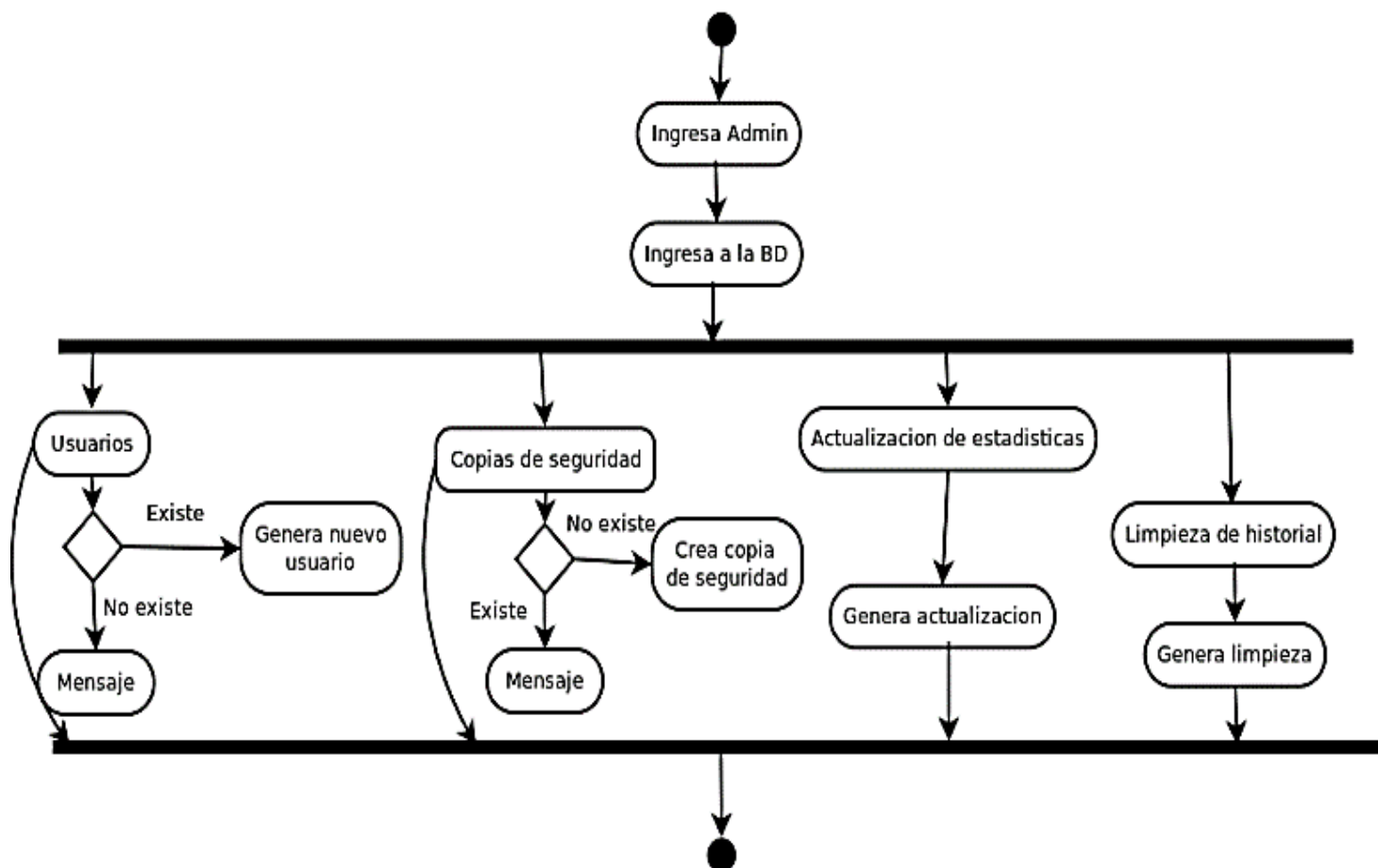


Figura 10. Diagrama de actividades, manipulación de la base de datos. (Autoría propia).

Diagramas de Ingreso de información

En los diagramas de ingreso de información es importante mencionar que el supervisor será el único que podrá incorporar información en el sistema. Por el contrario, un empleado no está autorizado para realizar esta función.

Las figuras que a continuación se muestran explican como el supervisor podrá dar de alta un registro no obstante, se debe realizar un proceso como se muestra posteriormente.

- 1) Debe ingresar a la clase donde desea dar de alta información
- 2) El usuario debe llenar el formulario
- 3) Se guarda la información

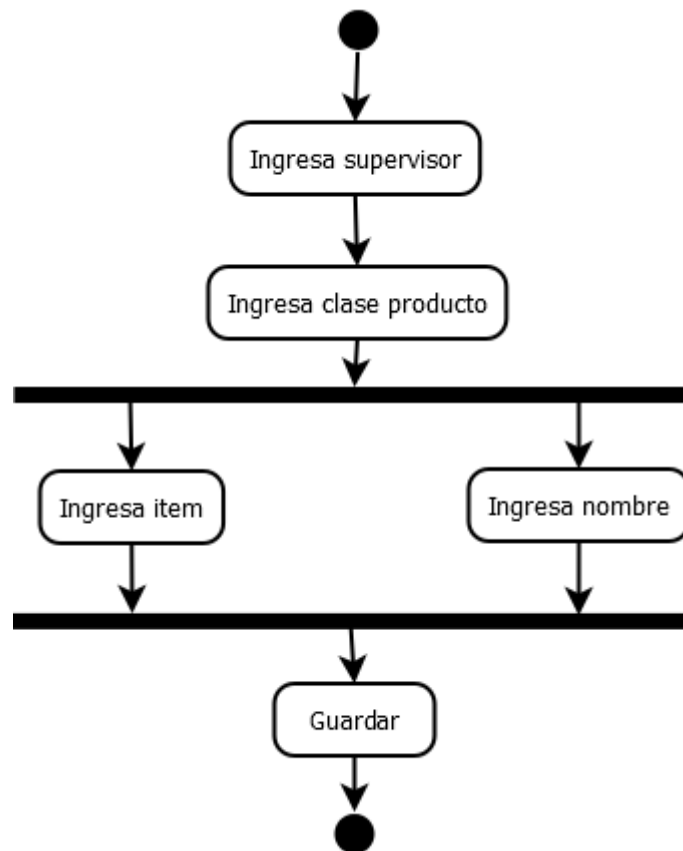


Figura 11. Ingresa información del producto. (Autoría propia).

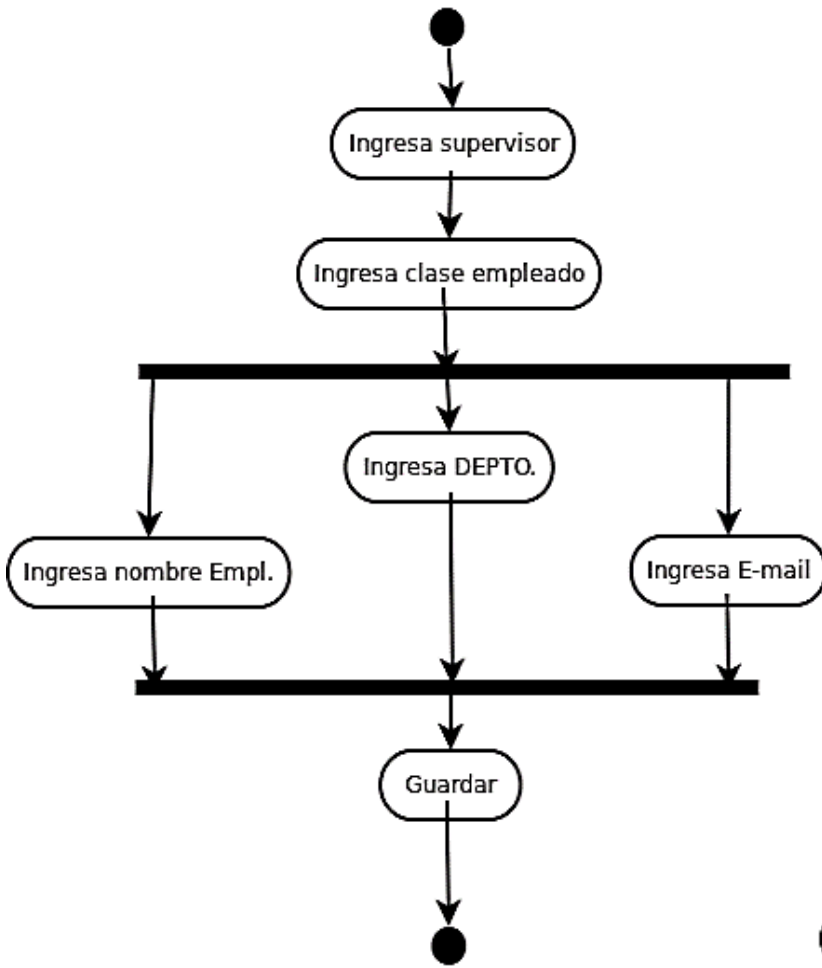


Figura 12. Ingresa información del empleado. (Autoría propia).

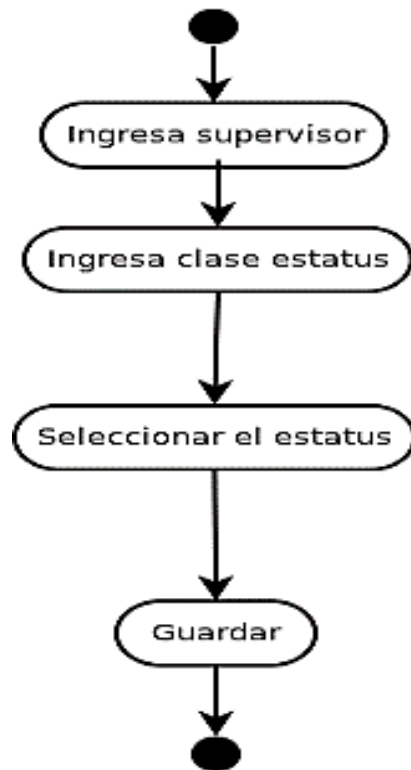
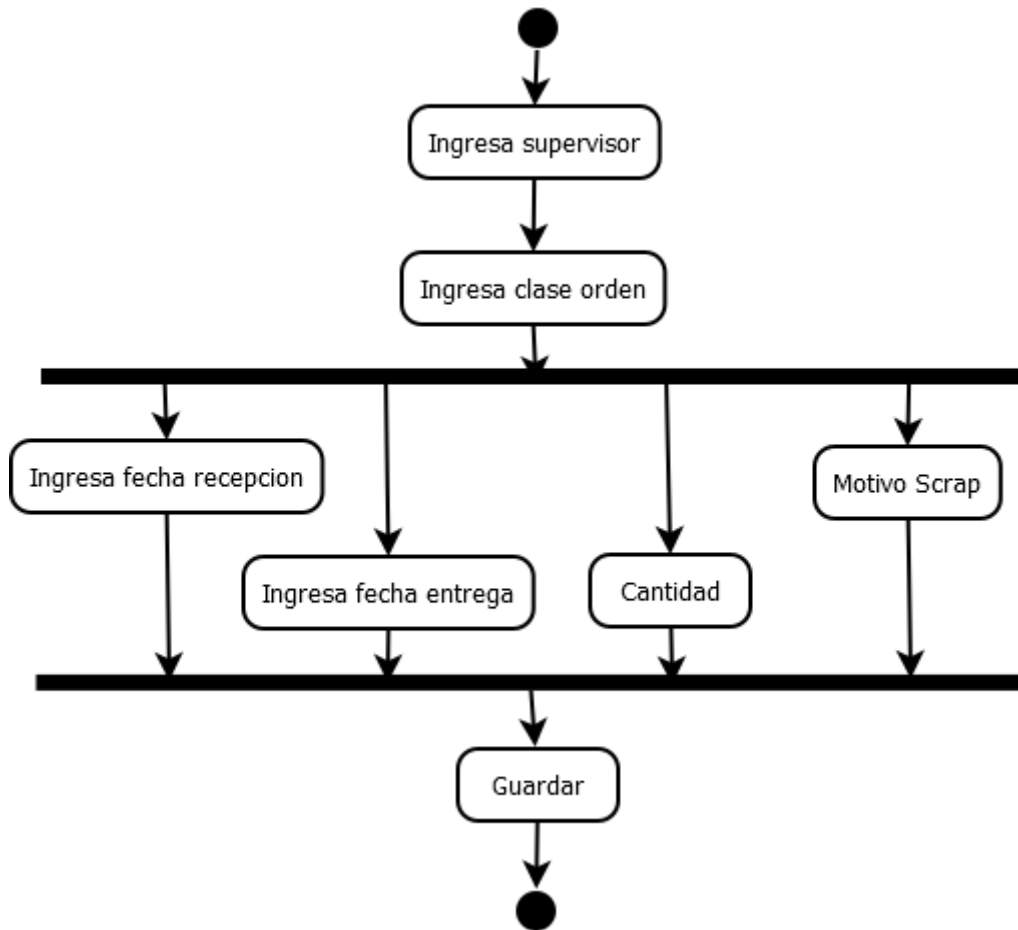


Figura 13. Ingresa información del estatus. (Autoría propia).

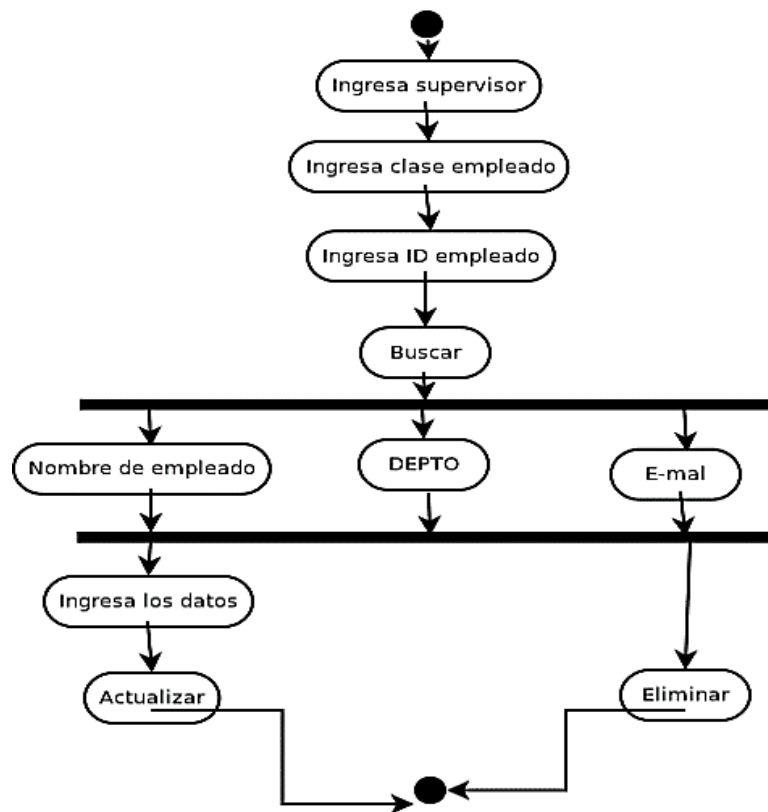


*Figura 14. Ingresar información de la orden.
(Autoría propia).*

Diagramas de actualización y eliminación de información

En los diagramas de actualización así como eliminación de datos el supervisor será el único que podrá realizar estas funciones y para poder hacerlo se debe seguir un proceso como se muestra en las siguientes figuras.

- 1) Debe ingresar a la clase a modificar
- 2) Después es importante especificar el id del registro que se desea modificar o eliminar.
- 3) Una vez que el sistema haya encontrado el registro modificaremos la información en cada atributo y actualizaremos o en su defecto se podrá eliminar la información seleccionada.



*Figura 15. Actualización y eliminación de información, Empleado.
(Autoría propia)*

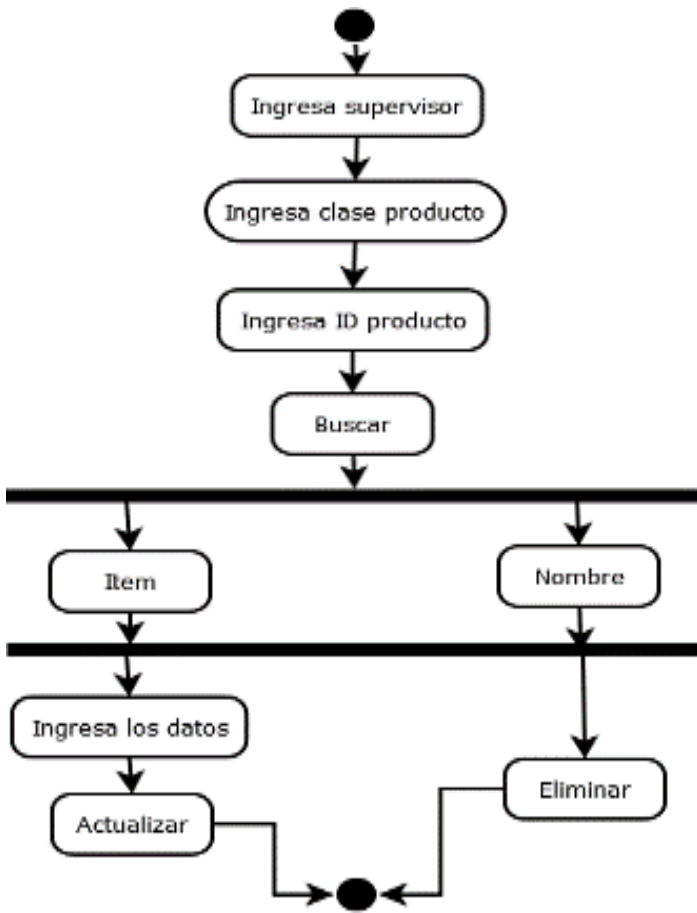


Figura 17. Actualización y eliminación de información Producto. (Autoría propia).

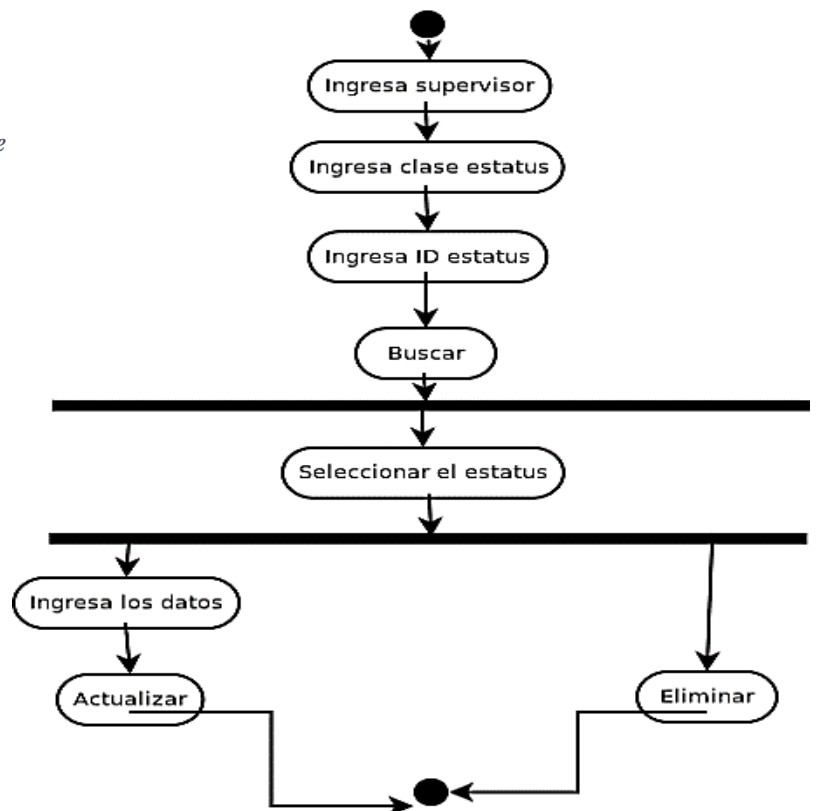
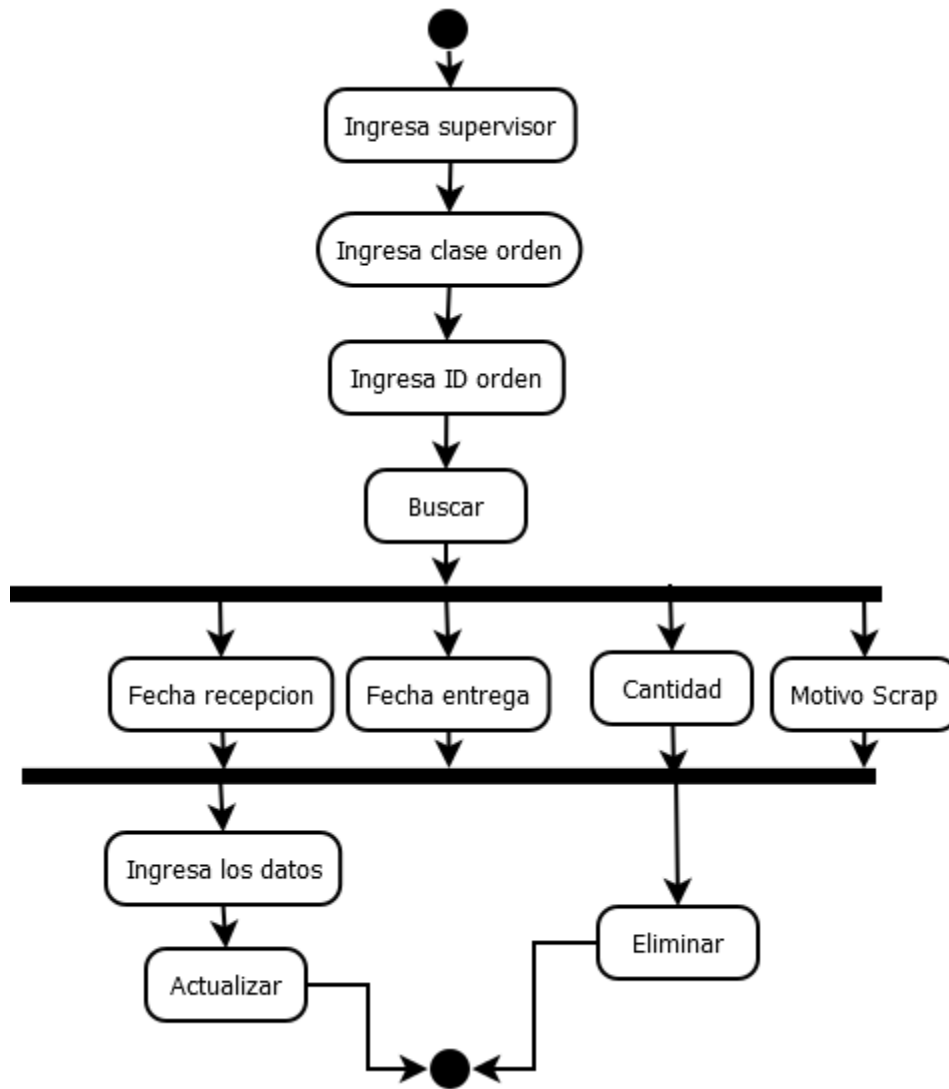


Figura 16. Actualización y eliminación de información Estatus. (Autoría propia).



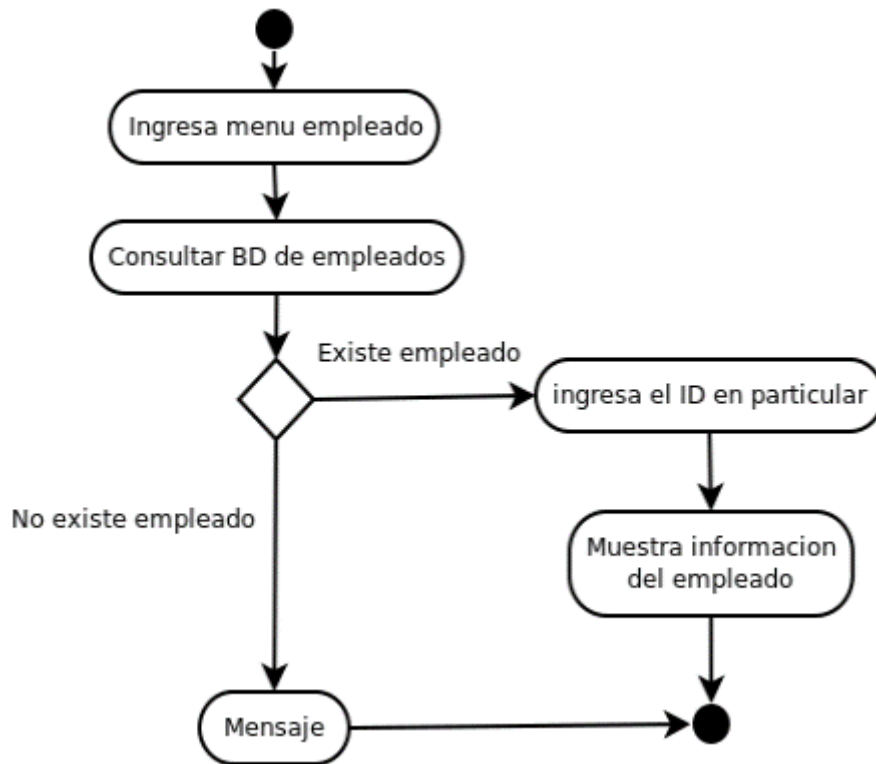
*Figura 18. Actualización y eliminación de información Orden.
(Autoría propia).*

Diagramas de consulta de información

Para consultar información los permisos no serán tan restringidos debido a que, el supervisor y un empleado autorizado podrán realizar esta función sin ningún problema.

Los pasos para consultar información de cada entidad se muestran en las siguientes figuras.

- 1) Para poder consultar algún registro el usuario primero deberá ingresar a la entidad que desee
- 2) Después se hace la consulta en la base de datos, si no existe el registro aparecerá un mensaje
- 3) Si existe el registro deberá ingresar el id en particular para mostrar la información y si el usuario lo desea hacer modificaciones



*Figura 19. Consulta de información Empleado.
(Autoría propia).*

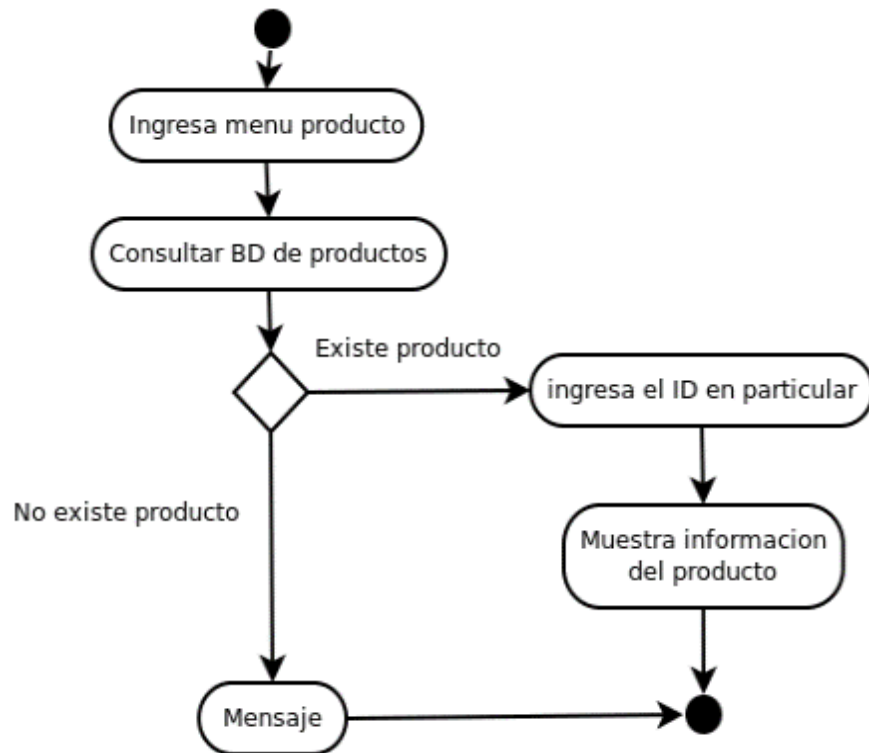


Figura 20. Consulta de información Producto, (Autoría propia).

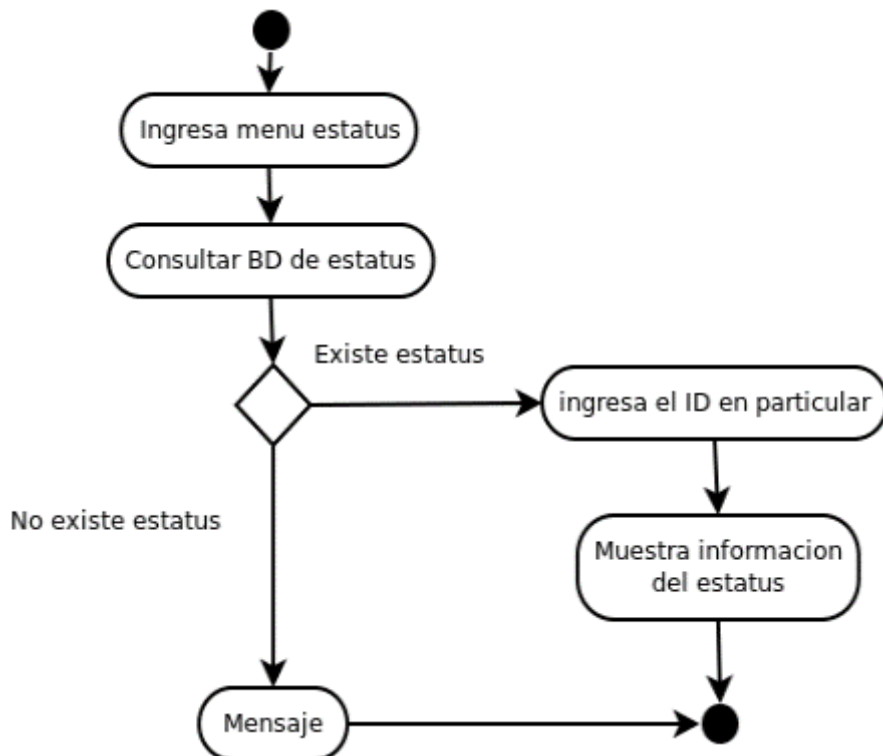
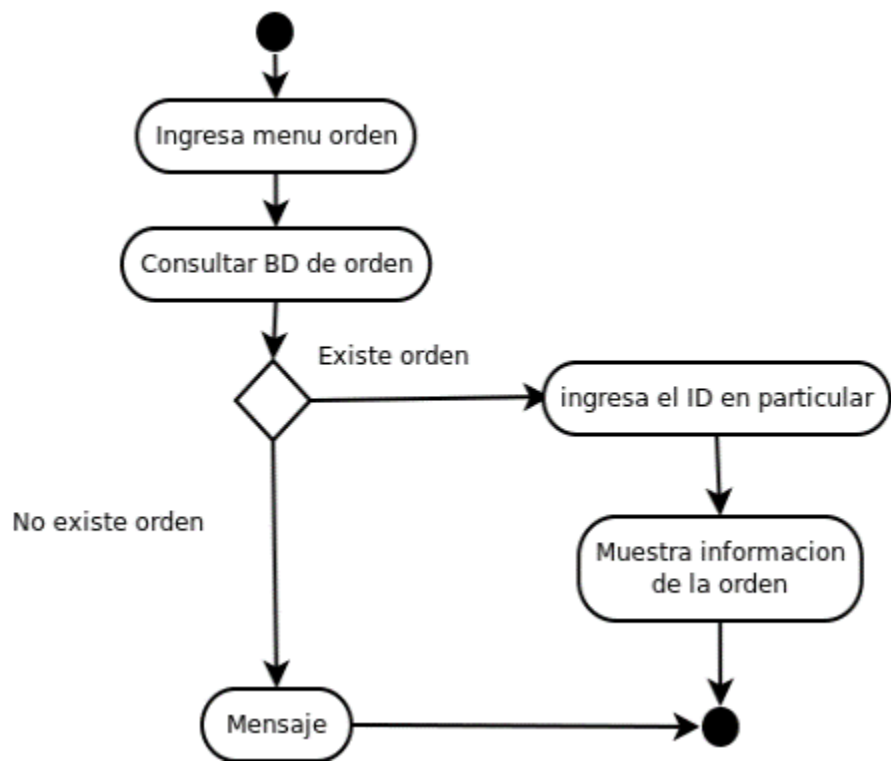


Figura 21. Consulta de información Estatus. (Autoría propia).



*Figura 22. Consulta de información Orden.
(Autoría propia).*

Diagrama impresión de órdenes

En el diagrama de órdenes el supervisor así como un empleado podrán realizar esta función, una vez autorizado el usuario puede hacer la impresión de órdenes, a continuación en las figuras se muestra el proceso que el usuario debe seguir.

- 1) Ingresar al menú consultas
- 2) Después el usuario deberá seleccionar un método de búsqueda
- 3) El sistema mostrara la información seleccionada por el usuario
- 4) El usuario imprimirá la orden

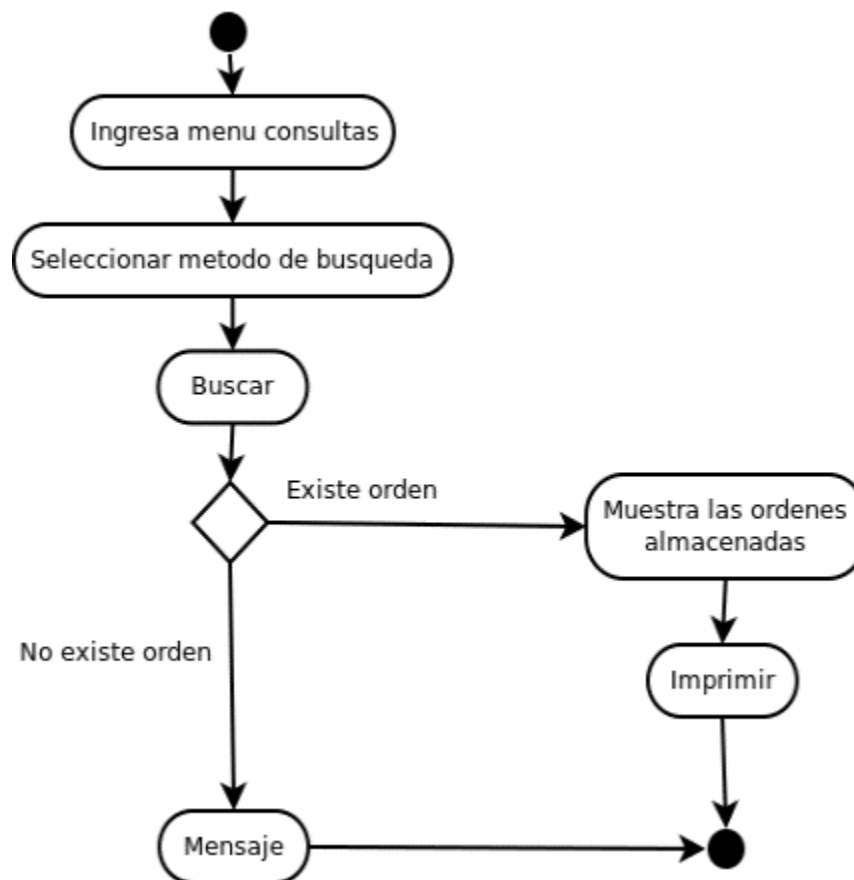
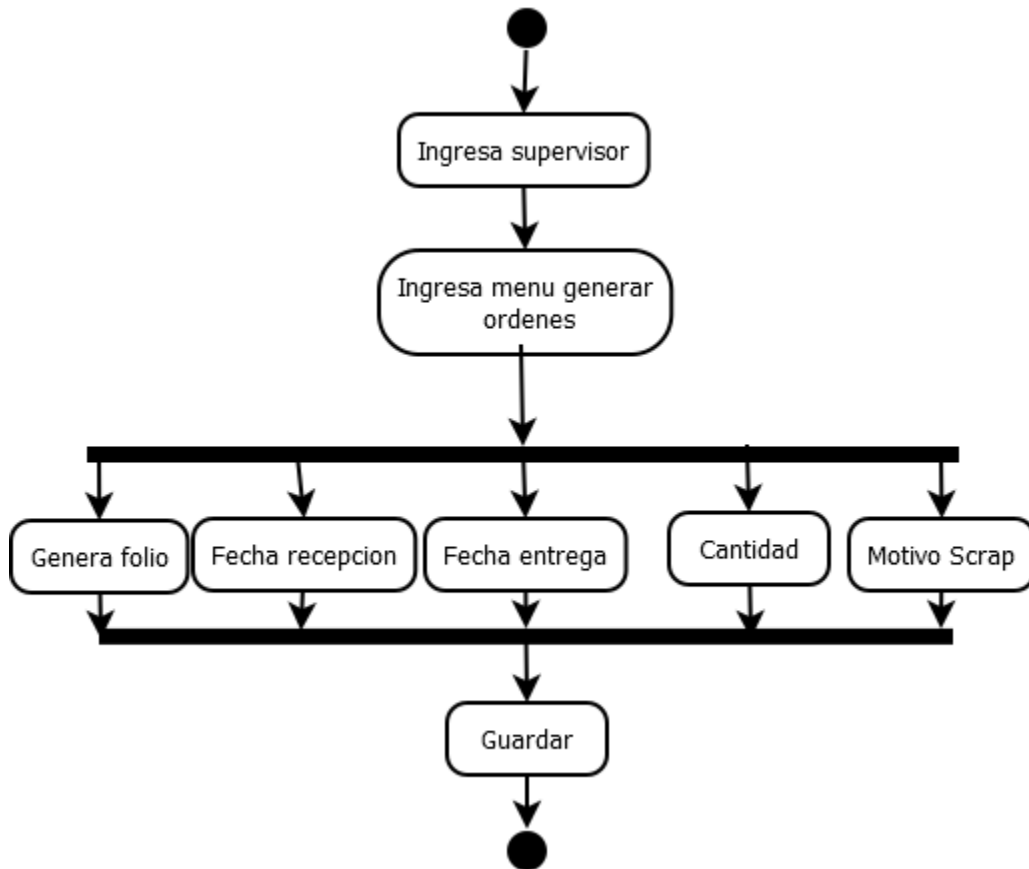


Figura 23. Impresión de órdenes. (Autoría propia).

Diagrama generar órdenes

El supervisor será el único que podrá generar las órdenes, a continuación en la figura 22 se muestran los pasos para generar órdenes.

- 1) Ingresar al menú generar órdenes
- 2) Llenar el formulario
- 3) Guardar la orden



*Figura 24. Generar órdenes.
(Autoría propia).*

7.1.6 Diseño de interfaces (Prototipo)

Una vez terminados los diagramas de actividades se procede al diseño de las interfaces, las cuales (Luna González, 2005) define como “todos los elementos gráficos que nos ayudan a comunicarnos con un sistema”.

Interfaz de login

Debe contar con seguridad en la autenticación del usuario mediante la validación del usuario y contraseña, al realizar esta comprobación se accede a la interfaz menú del sistema. Siendo que este es el primer contacto con el usuario es importante que la presentación sea amigable para que sea sencillo el manejo por parte del mismo.

Por lo tanto se ha diseñado la siguiente presentación para las pantallas del sistema asimismo, la base será la misma, en ella hacemos el uso de elementos importantes para la visualización del sistema tales como, el título de la pantalla en cuestión y un logo, para comodidad del usuario se manejara todo mediante botones y pestañas. El diseño de la pantalla principal está dividido en *frames* para mayor accesibilidad y despliegue de información.



INTELESI 13

Usuario

Contraseña

Accesar Salir

Figura 25. Interfaz login. (Autoría propia).

Interfaz menú

En la interfaz menú el usuario podrá acceder a otras 6 interfaces donde llenará los formularios para generar las ordenes de *scrap*, también podrá ingresar a la interfaz de consultas para verificar las ordenes almacenadas. Es decir, el usuario podrá navegar en los diferentes *frames* sin ningún problema al llenar los formularios este podrá saltar de pantalla en pantalla y al finalizar simplemente tendrá que salir del sistema con el botón designado.



Figura 26. Interfaz menú, (Autoría propia).

A continuación se muestra una tabla con el funcionamiento de cada botón de la interfaz menú.

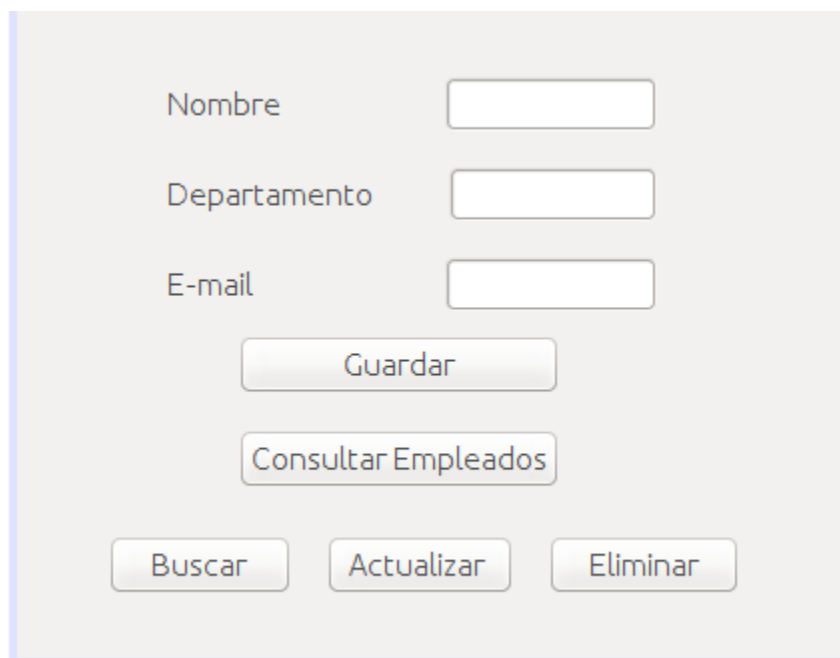
Tabla 8. Funciones de los botones de la interfaz menú, (Autoría propia).

Botón	Descripción
Empleado	Direcciona a la interfaz empleado
Producto	Permite acceder a la interfaz producto
Diagnóstico	Accede a la interfaz diagnóstico
Estatus	Accede a la interfaz estatus
Generar orden	Direcciona a la interfaz donde se generan las ordenes
Consultar ordenes	Accede a la interfaz donde se hacen las consultas de las ordenes

Interfaz empleado

El empleado será el que autorice los envíos del *scrap* en el proceso productivo este deberá ser almacenado para mantener un registro de las ordenes. Es decir, el empleado pertenece a un departamento en específico en cual deberá ser ingresado junto con el nombre del empleado y su correo electrónico para contactarlo por cualquier problema.

En esta interfaz el usuario podrá realizar distintas funciones de acuerdo a los requerimientos que necesite, como dar de alta registros, modificarlos, eliminarlos y consultarlos.

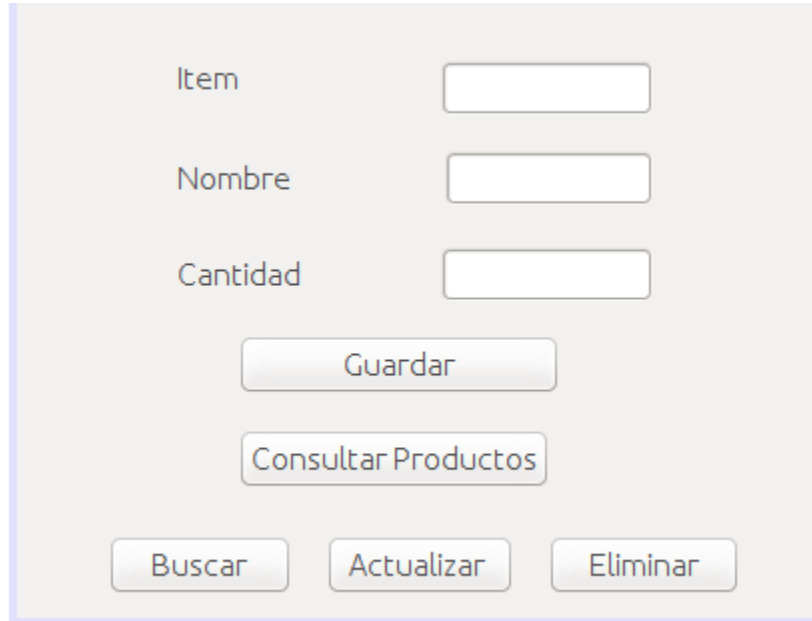


The image shows a web form for employee management. It consists of three text input fields labeled 'Nombre', 'Departamento', and 'E-mail'. Below the input fields are five buttons: 'Guardar', 'Consultar Empleados', 'Buscar', 'Actualizar', and 'Eliminar'. The form is enclosed in a light gray box with a thin blue border.

Figura 27. Interfaz empleado. (Autoría propia).

Interfaz producto

Esta interfaz consiste en ingresar los productos defectuosos que se generan en el proceso productivo y que posteriormente se envían a *scrap*.



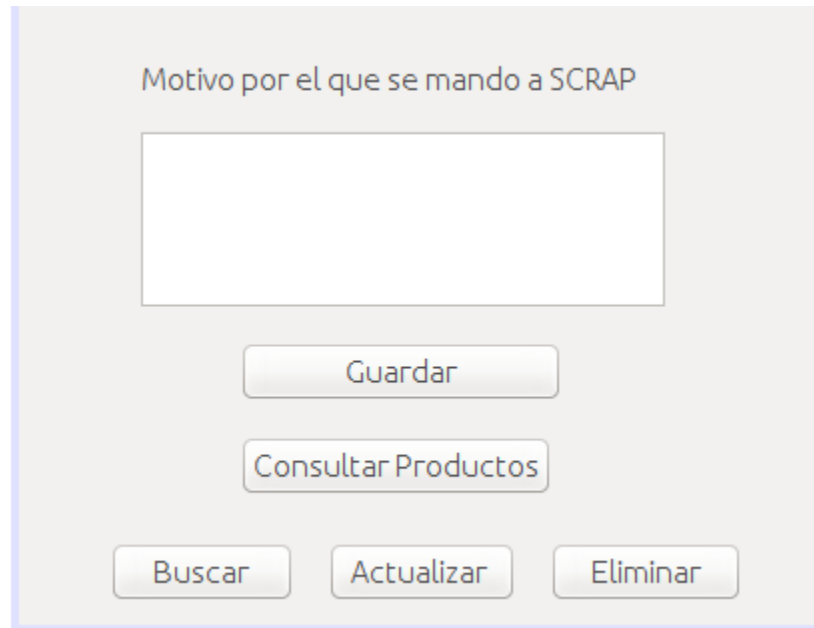
The image shows a web form for entering defective products. It features three input fields for 'Item', 'Nombre', and 'Cantidad'. Below these fields are four buttons: 'Guardar', 'Consultar Productos', 'Buscar', 'Actualizar', and 'Eliminar'.

Item	<input type="text"/>
Nombre	<input type="text"/>
Cantidad	<input type="text"/>

Figura 28. Interfaz producto. (Autoría propia).

Interfaz diagnóstico

En la interfaz diagnóstico se especificara el motivo por el cual un producto se mandó a *scrap*. Es decir, el usuario guardara la información y continuara con el siguiente *frame*.



Motivo por el que se mando a SCRAP

Guardar

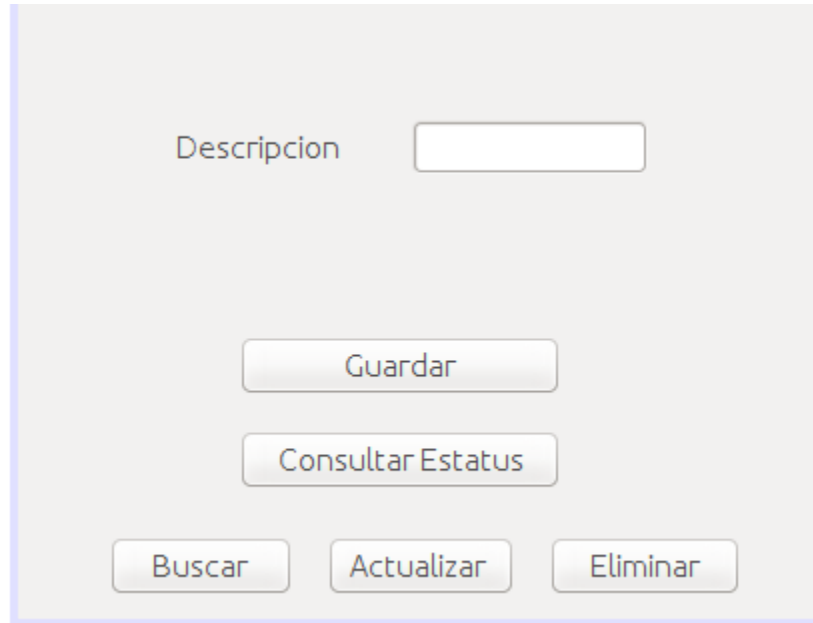
Consultar Productos

Buscar Actualizar Eliminar

Figura 29. Interfaz diagnóstico. (Autoría propia).

Interfaz estatus

En esta interfaz únicamente se especificara el estado de la orden si está abierta, cerrada, pendiente o cancelada.

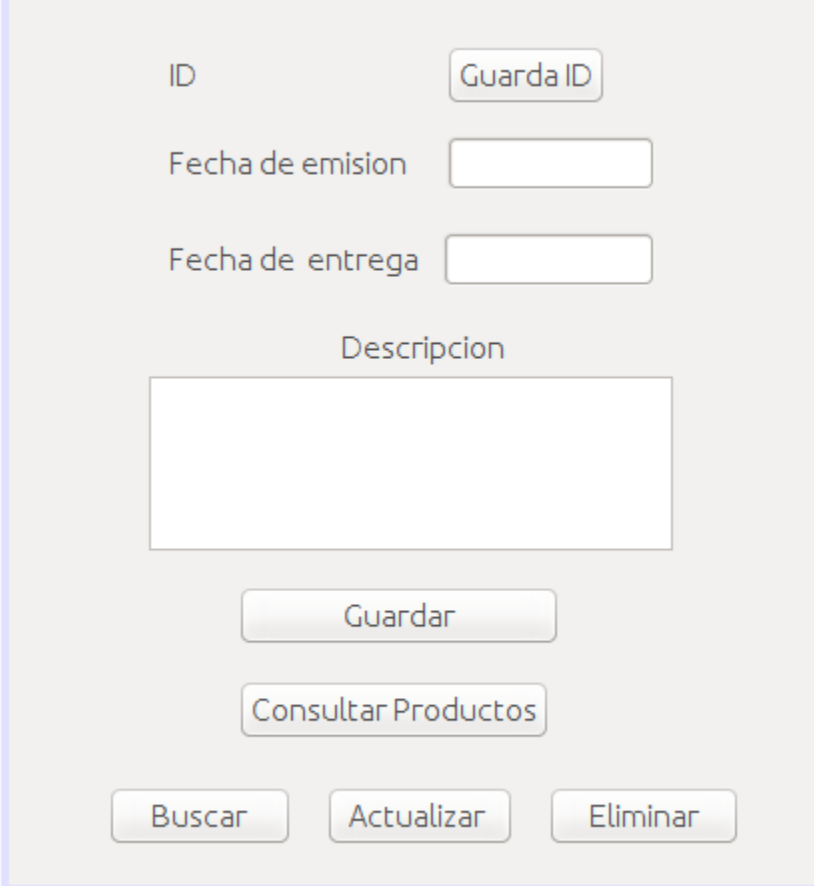


The image shows a user interface for managing order status. It features a text input field labeled "Descripcion" at the top. Below the input field are three buttons: "Guardar" (Save), "Consultar Estatus" (Check Status), and a row of three buttons: "Buscar" (Search), "Actualizar" (Update), and "Eliminar" (Delete). The buttons are light gray with rounded corners and a subtle gradient.

Figura 30. Interfaz estatus. (Autoría propia).

Interfaz orden

Finalmente una vez que el usuario haya llenado todos los formularios ingresara a esta pantalla donde emitirá una orden de *scrap*.



The image shows a web form for creating an order. It features several input fields and buttons. At the top, there is an 'ID' label and a 'Guarda ID' button. Below that are two date input fields labeled 'Fecha de emision' and 'Fecha de entrega'. A large text area labeled 'Descripcion' is positioned below the date fields. At the bottom of the form, there are four buttons: 'Guardar', 'Consultar Productos', 'Buscar', 'Actualizar', and 'Eliminar'.

Figura 31. Interfaz orden. (Autoría propia).

Interfaz consultas

Esta interfaz es muy importante debido a que se podrá llevar el monitoreo de todas las ordenes almacenadas. En definitiva, se asignarán métodos de búsqueda para que el usuario lleve un manejo de la información con mayor rapidez asimismo, tendrá la opción de generar reportes en Excel (ver figura 30).



The image shows a web-based query interface. At the top, there is a search field labeled 'Buscar por:' with an empty input box. Below this is a table with five columns: 'Orden', 'Empleado', 'Producto', 'Diagnosti...', and 'Estatus'. The table has four empty rows. To the right of the table is a vertical scrollbar. At the bottom of the interface, there are two buttons: 'Ordenes Almacenadas' and 'Exportar a Excel'.

Figura 32. Interfaz consultas. (Autoría propia).

Es importante señalar que los usuarios que tendrán acceso a esta interfaz serán el supervisor así como un empleado autorizado.

7.2 Implementación del sistema.

7.2.1 Creación de la base de datos relacional.

Finalmente concluida la etapa de análisis y diseño se procede al desarrollo del sistema de información donde inicialmente se comenzará con la creación de la base de datos esta consiste en la creación de tablas que componen el modelo relacional.

7.2.1.1 Características

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

Para definir la base de datos que se implementara se realizó un análisis e investigación previo tomando en cuenta todos los elementos que serán necesarios para el buen funcionamiento de INTELESIS 13 esto conlleva a identificar las identidades que componen el sistema así como las relaciones existentes entre ellas.

Al realizar este modelo, se procede con la creación de las tablas que componen el diccionario de datos que a su vez almacenará la información del *scrap* y que posteriormente, se vinculara con la API en java, se ha tratado de seguir la metodología adecuada y recomendada para la creación del sistema de información.

Asimismo después de realizar un estudio con los actores descritos en la fase de análisis y diseño el modelo relacional INTELESIS 13 se muestra terminado en la figura 31.

7.2.1.3 Modelo relacional INTELESIS 13

Como vemos en el siguiente modelo realizado en MySQL *workbench*, existen tablas que representan cada una de las entidades del sistema de información las cuales son: empleado, producto, estatus y orden. Además, están relacionadas entre ellas de modo que si un producto pertenece a una orden, se relaciona por el campo *Producto_idProducto* y lo mismo con las demás tablas. Una vez terminado el modelo relacional se generó un script el cual se ejecutó de manera correcta.

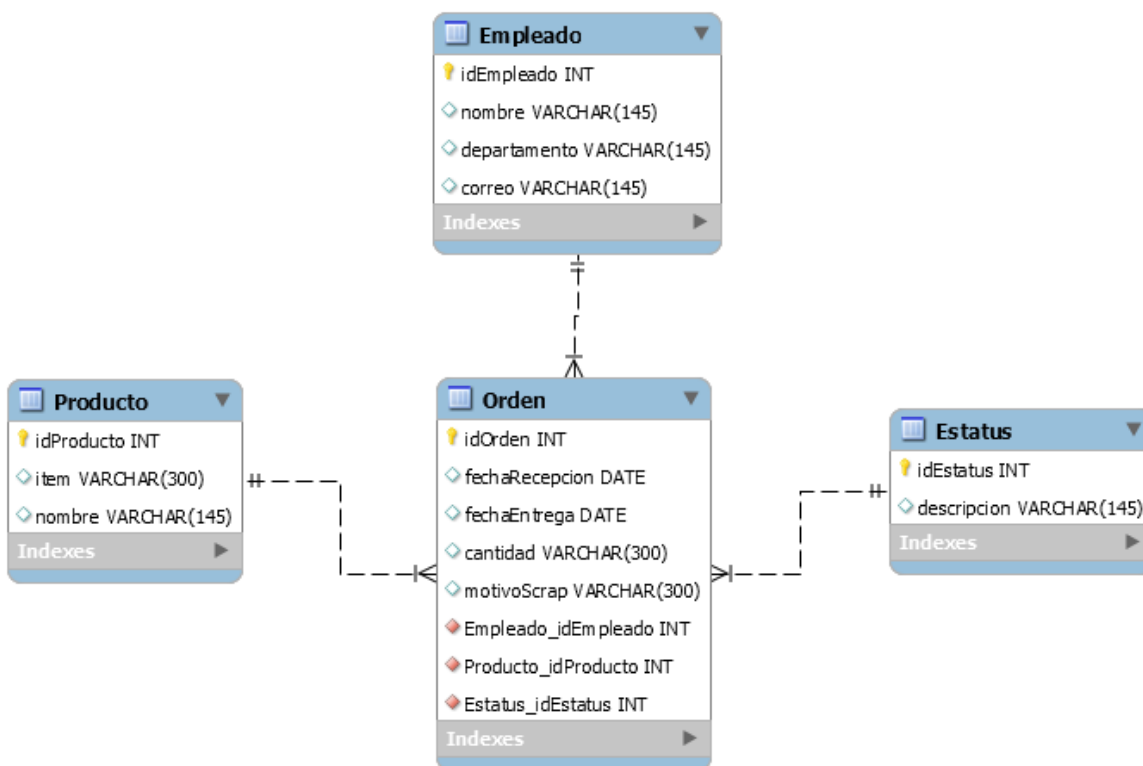


Figura 33. Modelo relacional SCRAP. (Autoría propia).

7.2.2 Creación de las vistas (interfaces)

Interfaz Login

Una vez finalizada la base de datos se comienza a desarrollar la aplicación en java específicamente con la creación de las interfaces. Por consiguiente, en la interfaz *login* se visualiza de manera general el inicio del sistema INTELESIS 13, que es donde los usuarios ingresan sus datos para tener acceso a la sesión correspondiente esto es únicamente para aquellos usuarios que estén dados de alta en el sistema.



The image shows a login window titled "INTELESIS 13". On the left side, there is the official coat of arms of the Universidad Autónoma de Tlaxcala, which includes a sun, a bell, and the motto "PATRIA - CIENCIA Y TRABAJO". To the right of the coat of arms, there are two text input fields. The first is labeled "Usuario" and the second is labeled "Contraseña". In the bottom right corner of the window, there is a green checkmark icon.

Figura 34. Interfaz LOGIN. (Autoría propia).

Una vez que el usuario ingreso al sistema este lo direccionara al menú principal del sistema INTELESIS 13, donde vendrán distintas opciones para generar las órdenes de *scrap*. Cabe mencionar que con este tipo de restricciones de seguridad el sistema incrementara bastante la protección de la información evitando el ingreso de cualquier usuario que pueda generar alteraciones así como fraudes.

Interfaz Menú

Mientras que la interfaz menu es la pantalla principal del sistema de informacion asi como la que direccionara a todos los formularios, dentro de esta existen las siguientes opciones (ver figura 33).

- 1) *Empleado*; Contiene informacion de empleados pertenecientes a los diferentes departamentos los cuales son responsables del producto que se envia a *scrap*.
- 2) *Producto*; Contiene informacion de los productos que se mandan a *scrap* por algun defecto en particular.
- 3) *Diagnostico y Estatus*; Registro de informacion de los diagnosticos y estatus de cada orden que se va generando.
- 4) *Generar orden*; Contiene un pequeño formulario para dar de alta la orden.
- 5) *Consultar ordenes*; Contiene informacion de todas las ordenes almacenadas en el sistema.



Figura 35. Interfaz menú. (Autoría propia)

Interfaz empleado

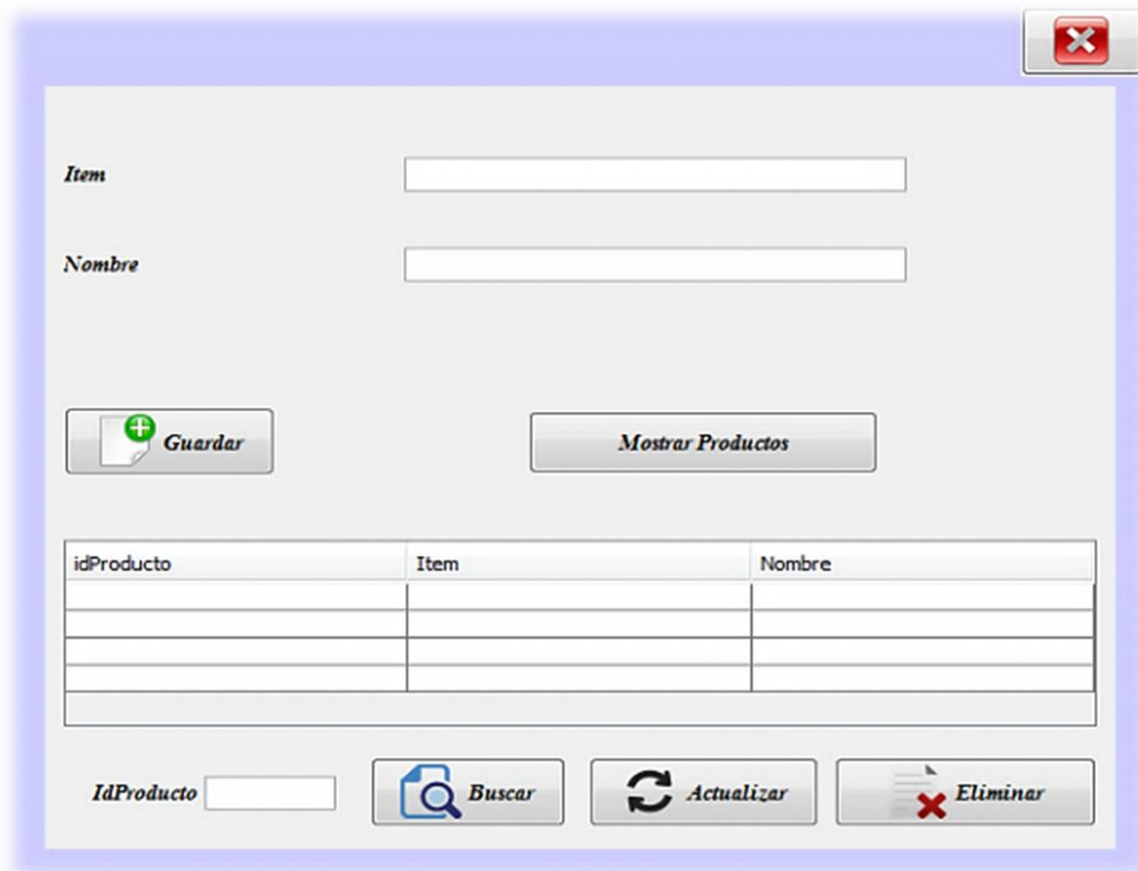
En la interfaz empleado se encuentra un pequeño formulario el cual permite el registro del personal responsable de cada departamento. Además, el usuario tiene la posibilidad de hacer una consulta de los registros que han sido dados de alta así como buscar registros por medio de su ID y de esta forma poder actualizar o eliminar la información e posteriormente almacenar la información.

idEmpleado	Nombre	Departamento	Correo
------------	--------	--------------	--------

Figura 36. Interfaz empleado. (Autoría propia).

Interfaz producto

En la interfaz producto se permite ingresar la siguiente información: ítem y nombre del producto. Estos productos son enviados por los empleados de los distintos departamentos de la empresa generalmente por algún defecto de fabricación. Una vez que un producto es almacenado se guarda y es considerado como *scrap* (ver figura 35).



The interface consists of a window with a close button in the top right corner. Below the window title bar, there are two input fields: one labeled 'Item' and one labeled 'Nombre'. Below these fields are two buttons: 'Guardar' (with a green plus icon) and 'Mostrar Productos'. Below the buttons is a table with three columns: 'idProducto', 'Item', and 'Nombre'. The table has five rows, with the first row being the header. Below the table is a horizontal bar containing an 'IdProducto' input field, a 'Buscar' button (with a magnifying glass icon), an 'Actualizar' button (with a refresh icon), and an 'Eliminar' button (with a red X icon).

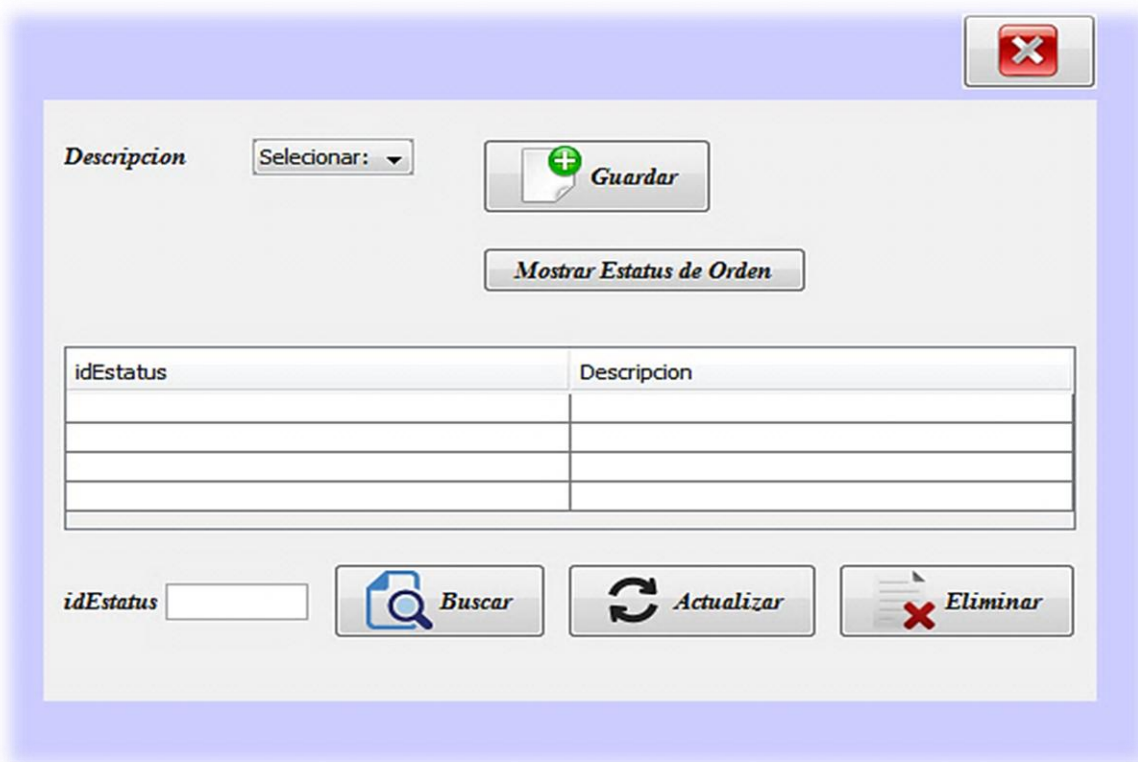
idProducto	Item	Nombre

Figura 37. Interfaz producto. (Autoría propia).

Interfaz estatus

Los atributos que maneja la interfaz estatus para definir el estado de una orden son los siguientes:

- 1) Abierta
- 2) Cerrada
- 3) Cancelada
- 4) Pendiente



idEstatus	Descripcion

Figura 38. Interfaz estatus. (Autoría propia)

La figura muestra las funciones con las que cuenta la interfaz así como, los estatus que serán asignados para cada orden generada.


Interfaz generar orden

En esta interfaz se generan las órdenes con la información del empleado, producto y estatus.

Esta sección está conformada por 4 partes (ver figura 37).

- 1) Generar ID.
- 2) Fecha de recepción y entrega.
- 3) Descripción de la orden.
- 4) Cantidad del producto.

No obstante, una vez almacenada la información en automático se generará la orden en la base de datos y podrá ser consultada mediante su ID en la interfaz de consultas.



The screenshot shows a software window titled "Interfaz generar orden" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface is light gray and contains several input fields and buttons:

- Top left: A text box labeled "ID" with a "Generar ID" button to its right.
- Below that: A text box labeled "Introduce el ID" with a "Guardar ID" button to its right.
- Top right: Two date pickers. The first is labeled "Fecha de emision" and the second "Fecha de entrega". Below them is a "Guarde las fechas" button.
- Middle: Four empty text boxes arranged horizontally.
- Bottom left: A text box labeled "Cantidad del producto".
- Bottom center: A large text area labeled "Motivo por el que se mando a SCRAP".
- Bottom: A large button with a floppy disk icon and the text "Guardar Orden".

Figura 39. Interfaz generar orden. (Autoría propia)

Interfaz consultas

En la interfaz consultas el usuario podrá informarse con el historial de todas las ordenes generadas de principio a fin además, tendrá distintos métodos de búsqueda para facilitar la revisión en caso de saturación de la información. Asimismo se implementó un método para exportar los registros a Excel (ver figura 38).

El usuario podrá buscar las órdenes por medio de su:

- 1) ID
- 2) Nombre del empleado
- 3) Ítem del producto
- 4) Estado de la orden

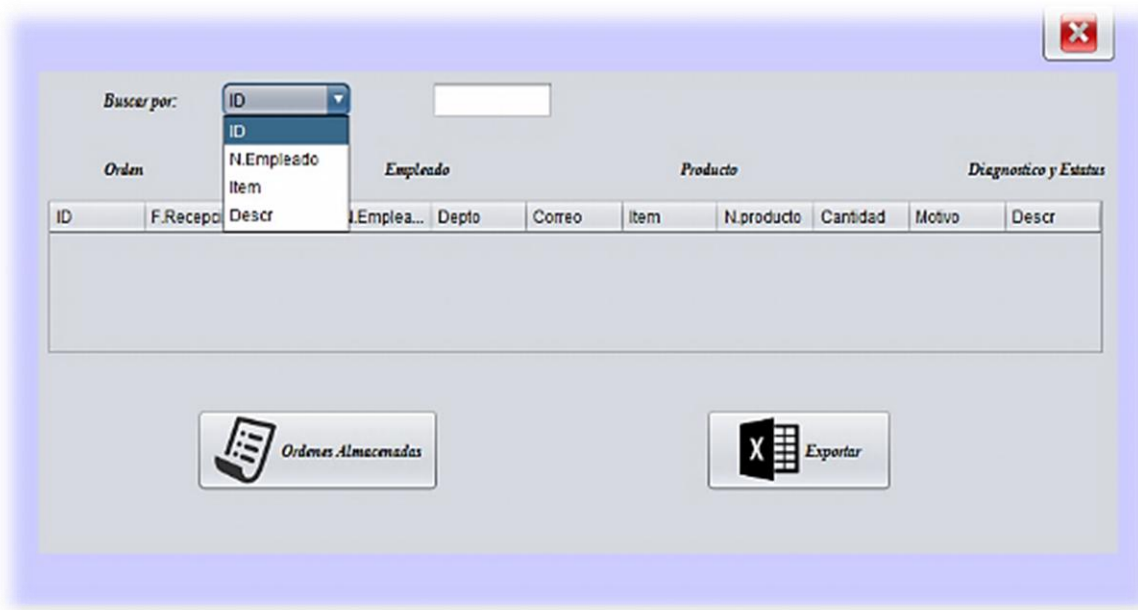


Figura 40. Interfaz consultas. (Autoría propia)

Una vez que el usuario termine de consultar toda la información, este tendrá la opción de generar un documento en Excel con los registros que seleccionados como se muestra en la figura 39.

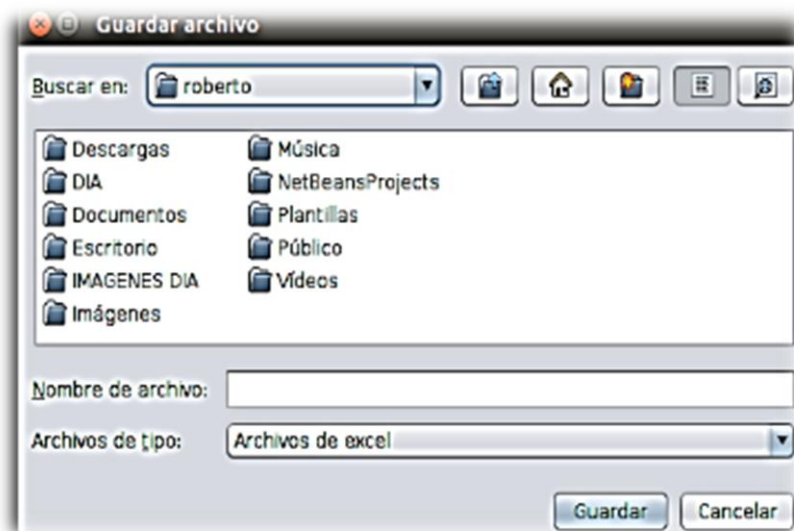


Figura 41. Interfaz consultas Opción exportar. (Autoría propia)

En definitiva, los registros serán almacenados automáticamente en la base de datos así como exportados para uso exclusivo del usuario y de esta manera tener un respaldo de toda la información que en algún momento será requerida por el supervisor.

Por todo lo mencionado anteriormente la creación así como el diseño del sistema están adaptados correctamente para que el usuario del sistema mantenga la información debidamente organizada y clasificada, lo cual incrementara bastante el control de la información comparándola antes de la implementación del sistema.

7.2.3 Codificación de aplicación (clase controlador)

Finalmente concluida la etapa de creación de interfaces se procede a codificar la aplicación. Para esto la clase controlador se encargará de la comunicación entre la interfaz gráfica y la base de datos. Asimismo, cuando el usuario interactúe con la vista, desencadenará la ejecución de ciertos eventos que serán recogidos por el controlador y lanzados al modelo, cuando el modelo devuelva la información, se actualizará la interfaz. Por ese motivo, la clase controlador tendrá un objeto de la clase *View*. En esta clase se implementará el código para la creación del sistema de información y la vinculación de la base de datos, por medio de los patrones de diseño *dao*, *dto*, *delegate* y *facade*.

En la figura 40 se puede observar los paquetes de los patrones de diseño que contienen todas las clases de la API.

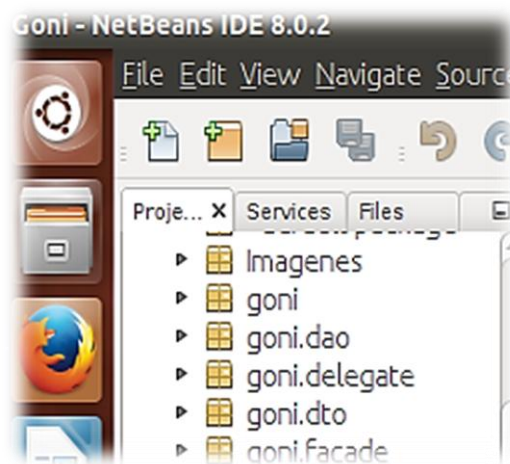


Figura 42. Patrones de diseño del sistema INTELESIS 13. (Autoría propia)

7.2.3.1 Librerías:

Una vez definida la estructura que tendrá el código de la aplicación se instalaron, las siguientes librerías de acuerdo, a las necesidades de nuestro sistema (ver figura 41).

1. *Librería MySQL JDBC*, Esta librería se instaló con el fin de vincular la base de datos del sistema *scrap* con la API en java.
2. *Librería JCALENDAR 1.3.3*, Esta librería es un selector de fecha de Java para recoger gráficamente una fecha.
3. *Librería JXL*, Esta librería se instaló con el fin de exportar las órdenes generadas a un documento en Excel.

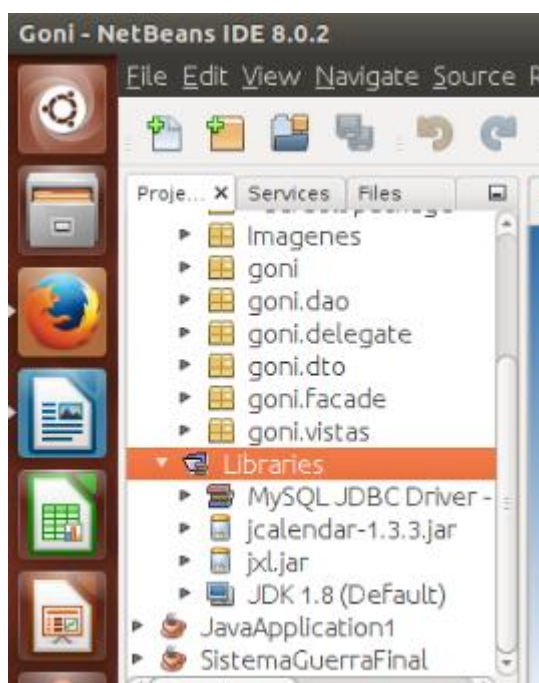


Figura 43. Librerías del sistema INTELESIS 13. (Autoría propia)

7.2.3.2 ActionPerformed

Este método se ejecutará cuando el usuario pulse sobre alguno de los tres botones de las interfaces que contienen formularios de la aplicación.

En función del comando que se ejecute:

- a) Insertar
- b) Borrar
- c) Modificar
- d) Buscar

7.2.3.3 MouseClicked

Este método se ejecutará cuando el usuario pulse sobre una fila de la tabla.

Cuando se pulse sobre una fila, se invocará a un procedimiento almacenado de la base de datos que por ejemplo, obtiene el nombre, departamento así como e-mail del empleado marcado y cargará los datos en los cuadros de texto de la ventana. Para obtener la información devuelta por el procedimiento almacenado, se hará uso del ResultSet. El resto de métodos cuya definición es obligatoria al implementar la interfaz MouseListener, no contendrán código alguno.

Creación del *DTO*

Una vez programados los botones y las tablas se hace la creación del DTO para cada entidad como fue establecido en el modelo relacional, esto permitió transportar toda la información de la base de datos manteniendo una comunicación con todas las interfaces del sistema, de esta manera se hace efectivo el almacenamiento y envío de datos entre todo el sistema.

Creación del *DAO*

Para la creación de las clases DAO se importaron las siguientes librerías que permitirán la conexión entre el DAO y DTO (ver figura 42), además de devolver los resultados generados en la base de datos a partir de los *inserts*, *update*, *delete* y *select*, a continuación se muestran las librerías importadas durante la creación de la clase DAO.

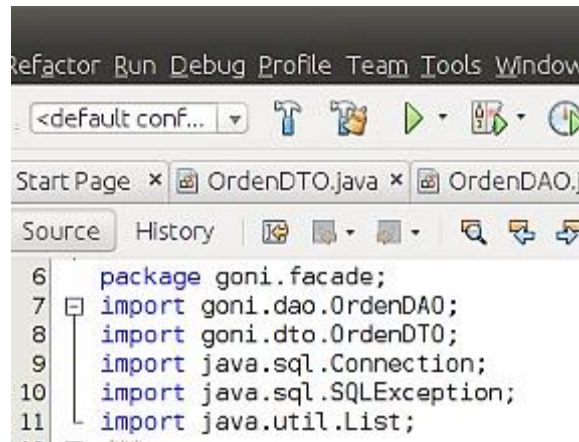
```
5  L  */
6  package goni.dao;
7
8  import java.sql.Connection;
9  import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.util.ArrayList;
13 import java.util.List;
14 L  import goni.dto.OrdenDTO;
15
```

Figura 44. Librerías de la clase DAO. (Autoría propia)

O sea, en la clase DAO, se realizarón funciones como insertar, actualizar, borrar y consultar datos de cada entidad de nuestra base de datos. De esta manera el sistema es capaz de realizar las diferentes funciones.

Creación del *FACADE*

Posteriormente en la clase *facade* se utilizaron las siguientes librerías (ver figura 43).



```
6 package goni.facade;
7 import goni.dao.OrdenDAO;
8 import goni.dto.OrdenDTO;
9 import java.sql.Connection;
10 import java.sql.SQLException;
11 import java.util.List;
```

Figura 45. Librerías de la clase *FACADE*. (Autoría propia)

Este patrón de diseño permite simplificar la interfaz de comunicación entre los objetos DTO y DAO de tal forma que para el objeto DTO sea más sencillo interactuar con el objeto DAO. Además, de realizar las funciones de lectura, creación y actualización de datos.

Creación del *DELEGATE*

Las librerías que se ocuparon en esta clase son las siguientes (ver figura 44).



```
4 * and open the template in the edit
5 */
6 package goni.delegate;
7 import goni.dto.OrdenDTO;
8 import goni.facade.OrdenFacade;
9 import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.SQLException;
12 import java.util.List;
13 /**
14 *
```

Figura 46. Librerías de la clase *DELEGATE*. (Autoría propia)

En el *delegate* se hizo la conexión entre la base de datos en MySQL y la API en java, contiene elementos de la conexión con SQL como el root, contraseña de nuestro localhost y el nombre de la base de datos.

7.2.4 Implementación de políticas de seguridad para INTELESIS 13

Seguridad física

Una vez desarrollado el sistema de información fue necesario implementar políticas de seguridad para evitar problemas físicos, por lo que todo el hardware que soporte de alguna manera la publicación y accesibilidad a un sistema de información debe tener un ambiente con acceso restringido sólo a personas autorizadas, esto es:

- a) Registrar el ingreso y egreso del personal al equipamiento.
- b) Si se cuenta con un control de acceso automático, el mismo debe registrar los ingresos así como egresos para permitir la identificación de todas las personas que entren y salgan del recinto.
- c) Prohibir comer, beber y fumar dentro del ambiente donde reside el equipamiento.
- d) Adoptar controles adecuados para minimizar el riesgo de amenazas potenciales, como pueden ser: robo, incendio, explosivos, humo, inundaciones o filtraciones de agua, polvo, vibraciones, efectos químicos, interferencia en el suministro de energía eléctrica, radiación electromagnética y derrumbes.
- e) Implementar un sistema de refrigeración que permita mantener una temperatura adecuada para el procesamiento. Prever equipamiento alternativo en caso de falla del primario.

Seguridad del sistema operativo y comunicaciones

Para la integridad del software se establecieron las siguientes indicaciones:

- 1) Mantener el sistema operativo actualizado, con los últimos parches de seguridad disponibles.
- 2) Configurar el sistema operativo de manera segura. Esto incluye entre otras actividades, deshabilitar cuentas de invitado o sesiones extras así como cambiar claves por defecto.

- 3) Administrar adecuadamente las cuentas de usuario procurando:
 - Otorgar acceso al sistema operativo sólo a personal autorizado, con privilegios mínimos que le permitan cumplir con sus funciones.
 - Establecer una política de uso de las cuentas críticas.
 - Controlar la vigencia de los accesos otorgados a los usuarios respecto del sistema operativo.
- 4) Asegurar la posibilidad de identificar unívocamente a todos los usuarios del sistema a través de sus cuentas.
- 5) Implementar políticas de contraseñas y de control de acceso a toda la información almacenada en el servidor de forma restrictiva.
- 6) Implementar y mantener actualizado un software antivirus.
- 7) Proteger con un firewall externo la red donde se ubica el servidor.

Resguardo de la base de datos

La seguridad en la base de datos es muy importante ya que contiene toda la información del *scrap* generado, por lo que se implementaron las siguientes medidas de seguridad:

- 1) La base de datos cuenta con un acceso restringido a la información ya que se asignó un usuario y contraseña exclusivos para el administrador.
- 2) Se definieron políticas de seguridad para el respaldo de la base de datos, con un esquema periódico semanal. Debido a que la información contempla datos indispensables en caso de pérdida, estos respaldos se elaborarán de manera manual, mediante el gestor de base de datos MySQL.
- 3) Las copias de seguridad deben ser almacenadas en un lugar físicamente seguro, para prevenir el daño, robo o pérdida de las mismas.
- 4) Se verificará periódicamente la correcta restauración de las copias de seguridad.
- 5) Se efectuarán copias de seguridad de los archivos del sistema, estas copias se realizarán semanalmente.

7.3 Pruebas y ajustes

Una vez establecidas las medidas de seguridad para el sistema de información se realizaron las siguientes pruebas para validar la funcionalidad del sistema y con esto evitar problemas a la hora de realizar la instalación.

7.3.1 Registros de ejecución de pruebas

Todas las pruebas se realizaron en un equipo HP *Pavilion g6* Series con procesador ADM, disco duro de 500 GB y 2GB de RAM, con el sistema operativo Ubuntu 16.04 *Xenial Xerus*, aunque cabe mencionar que el sistema se instaló en un equipo COMPAQ *Presario F755LA* con sistema operativo Windows XP Profesional Intel Celeron, disco duro de 120 GB y memoria RAM de 1.75 GB funcionando de manera correcta.

Así pues, se comprobó que la comunicación cliente servidor accediera a cada uno de los módulos de manipulación de la base de datos del sistema. Por último, fue necesaria la instalación del paquete de herramientas JDK para la correcta ejecución de la aplicación.

7.3.2 Prueba unitaria

Posteriormente cuando el sistema se ejecutó fue fundamental hacer la prueba unitaria para corroborar y asegurar el buen funcionamiento de cada una de las interfaces con la finalidad de aislar cada parte del software y mostrar que las secciones individuales son correctas por lo que se dividieron las pruebas a cada una de las interfaces. Como a continuación se muestran.

Interfaz LOGIN

Se comprobó la integridad de la interfaz de *login*, en cuanto al funcionamiento de sus opciones de acceso a usuario y contraseña (ver figura 45).



Figura 47. Mensaje de validación errónea. (Autoría propia)

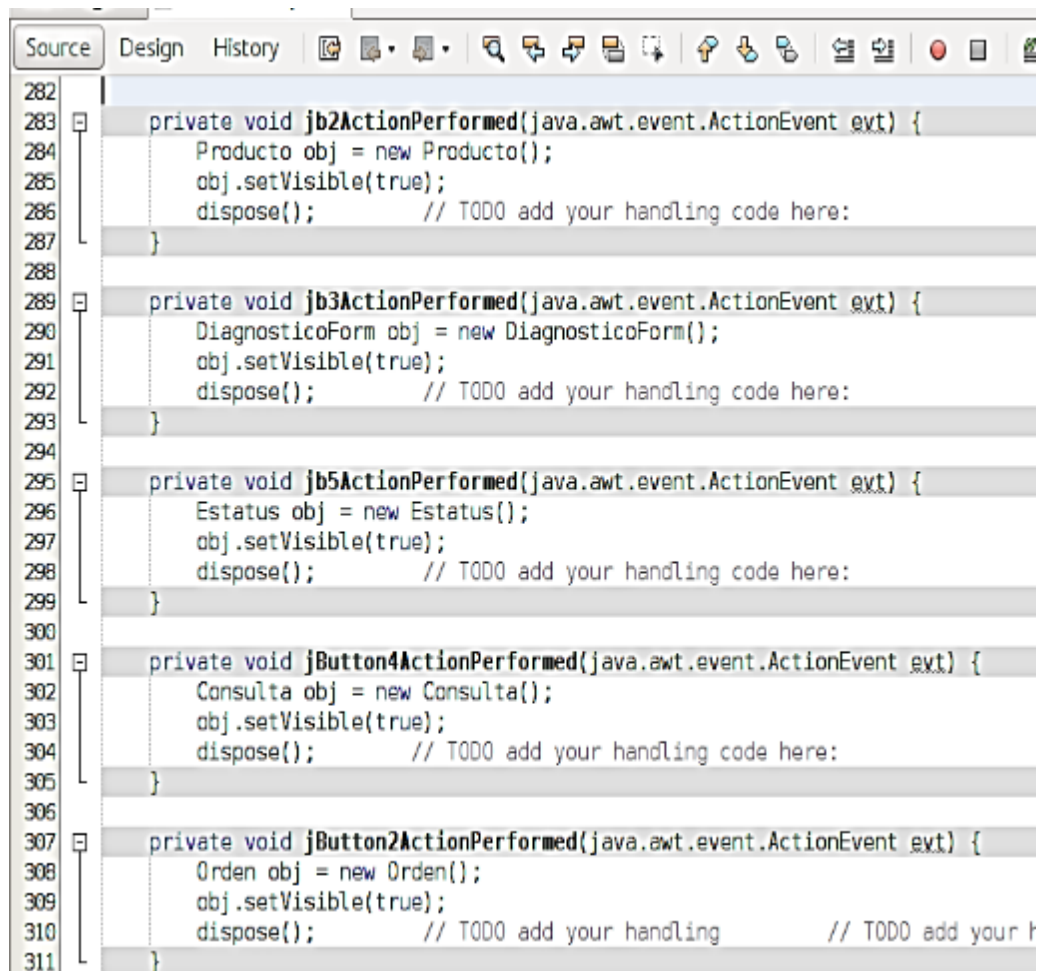
Para la opción de usuario y contraseña se comprobó que los datos almacenados en la base de información sean idénticos a los escritos en el formulario de acceso al menú, es decir se distingue mayúsculas y minúsculas. A continuación se muestra un fragmento de código donde se definen los valores legítimos.

```
198 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
199     String user = usuario.getText();
200     String pass = contraseña.getText();
201
202
203     if(user.equals("hilda") && pass.equals("123")){
204         OrdenForm pp = new OrdenForm();
205         pp.setVisible(true);
206         this.dispose();
207     }
208     else{
209         JOptionPane.showMessageDialog(null, "Usuario o Contraseña Incorrecta");
210     }
211 }
```

Figura 48. Código usuario y contraseña de acceso al menú. (Autoría propia)

Interfaz menú

En la interfaz menú se comprobó que se accediera de manera correcta a los diferentes módulos que contiene el sistema como los son: empleado, producto, estatus, órdenes y consultas.

A screenshot of an IDE window showing Java source code. The window has tabs for 'Source', 'Design', and 'History'. The code is organized into five method blocks, each starting with a private void method signature that takes a java.awt.event.ActionEvent ext parameter. The methods are: 1. jb2ActionPerformed (lines 283-287) which creates a Producto object, sets it visible, and calls dispose(). 2. jb3ActionPerformed (lines 289-293) which creates a DiagnosticoForm object, sets it visible, and calls dispose(). 3. jb5ActionPerformed (lines 295-299) which creates an Estatus object, sets it visible, and calls dispose(). 4. jButton4ActionPerformed (lines 301-305) which creates a Consulta object, sets it visible, and calls dispose(). 5. jButton2ActionPerformed (lines 307-311) which creates an Orden object, sets it visible, and calls dispose(). Each method block contains a comment: // TODO add your handling code here:.

```
282 |
283 | private void jb2ActionPerformed(java.awt.event.ActionEvent ext) {
284 |     Producto obj = new Producto();
285 |     obj.setVisible(true);
286 |     dispose(); // TODO add your handling code here:
287 | }
288 |
289 | private void jb3ActionPerformed(java.awt.event.ActionEvent ext) {
290 |     DiagnosticoForm obj = new DiagnosticoForm();
291 |     obj.setVisible(true);
292 |     dispose(); // TODO add your handling code here:
293 | }
294 |
295 | private void jb5ActionPerformed(java.awt.event.ActionEvent ext) {
296 |     Estatus obj = new Estatus();
297 |     obj.setVisible(true);
298 |     dispose(); // TODO add your handling code here:
299 | }
300 |
301 | private void jButton4ActionPerformed(java.awt.event.ActionEvent ext) {
302 |     Consulta obj = new Consulta();
303 |     obj.setVisible(true);
304 |     dispose(); // TODO add your handling code here:
305 | }
306 |
307 | private void jButton2ActionPerformed(java.awt.event.ActionEvent ext) {
308 |     Orden obj = new Orden();
309 |     obj.setVisible(true);
310 |     dispose(); // TODO add your handling // TODO add your t
311 | }
```

Figura 49. Código de acceso a las diferentes interfaces del sistema INTELESIS13. (Autoría propia)

Asimismo, como se puede observar en la figura 47 también se hizo la comprobación del método `obj.setVisible` para que el usuario pueda navegar entre el menú principal y los formularios.

Interfaz empleado

En la interfaz empleado se corroboró el correcto almacenamiento de información dentro de la base de datos y la correcta visualización de cada una de las opciones, en cuanto a las consultas solicitadas se comprobó la similitud de los datos descritos en la pantalla con la información resultante dentro la base de datos. A su vez se validó la modificación y eliminación de información. En las figuras siguientes se muestran las validaciones de todas las funciones de la interfaz empleado.

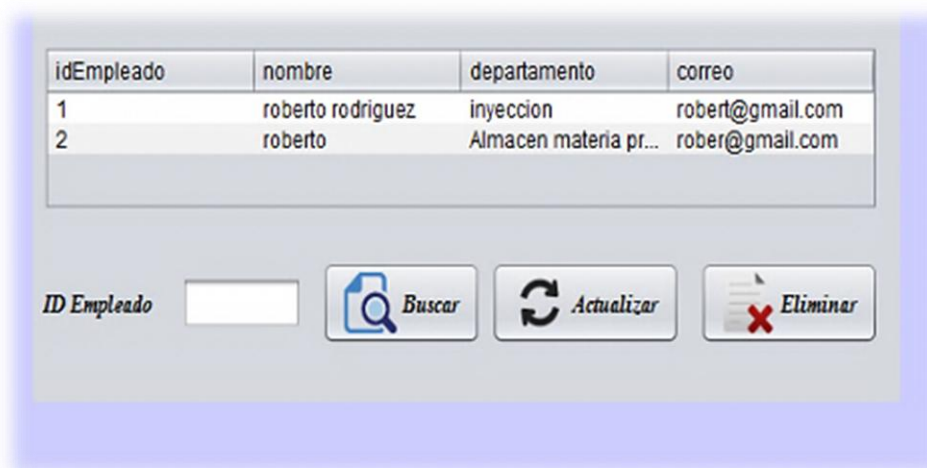


Figura 50. Entidad empleado con conexión a SQL. (Autoría propia)

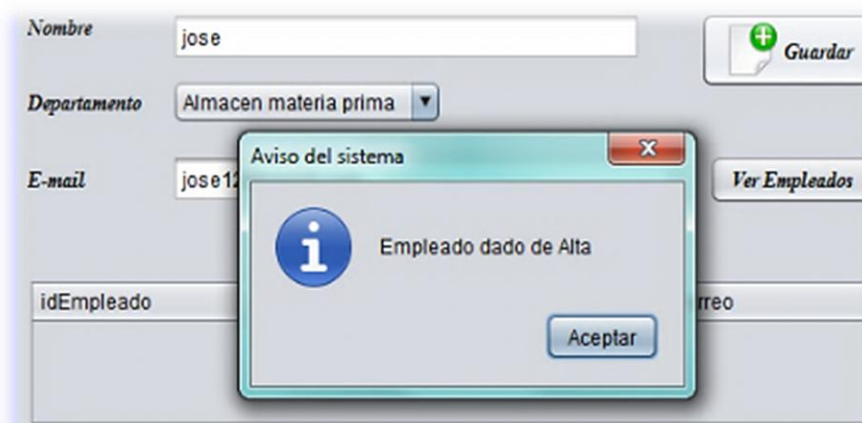


Figura 51. Validación de inserción de datos ha empleado. (Autoría propia)

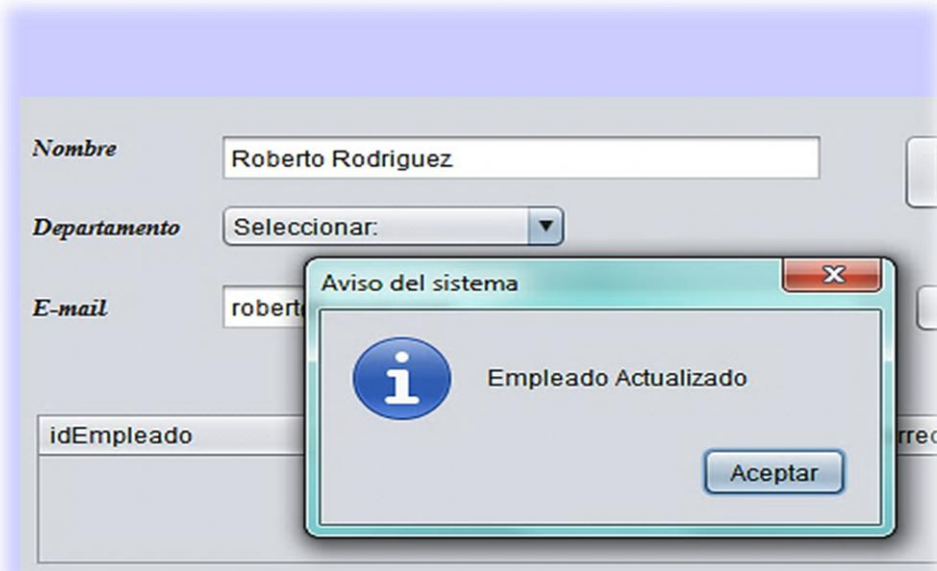


Figura 52. Validación de actualización de datos ha empleado. (Autoría propia)

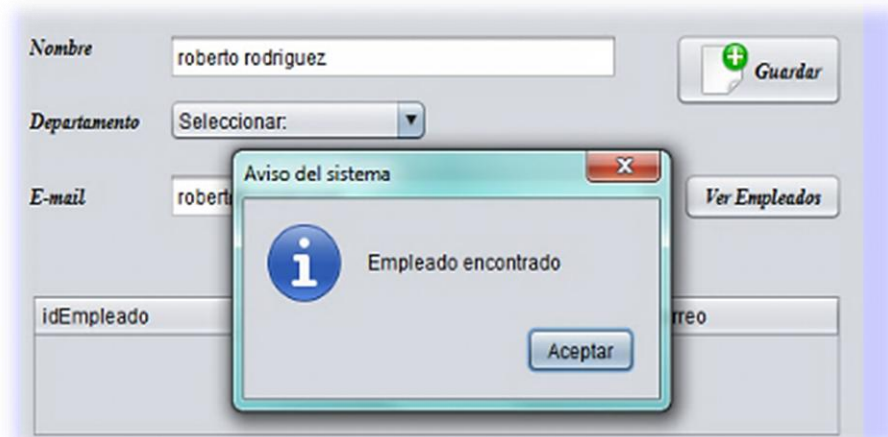


Figura 53. Validación de consulta de datos a empleado. (Autoría propia)

Interfaz producto

En el módulo producto se comprobó que los datos sean congruentes con la base de información, así como la inserción, actualización y consulta de datos. A continuación se muestran las pantallas con cada una de las validaciones de todas las funciones.

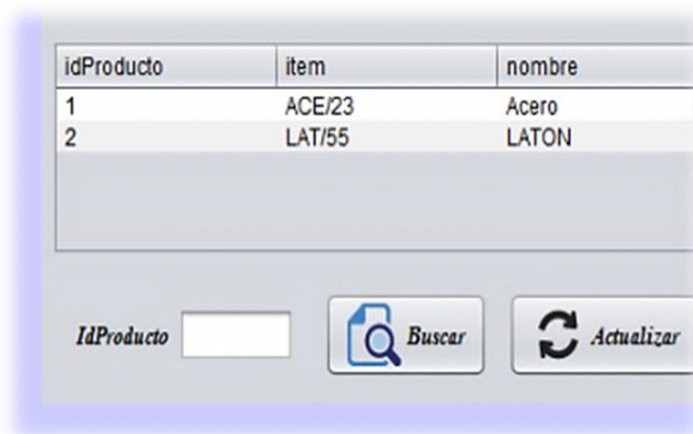


Figura 54. Entidad producto con conexión a SQL. (Autoría propia)



Figura 55. Validación de inserción de datos a producto. (Autoría propia)



Figura 56. Validación de consulta de datos a producto. (Autoría propia)



Figura 57. Validación de actualización de datos a producto. (Autoría propia)

Interfaz estatus

En la interfaz estatus se corroboró que todas las funciones trabajen de manera correcta, como se puede apreciar en las siguientes figuras.



Figura 58. Entidad estatus con conexión a SQL. (Autoría propia)

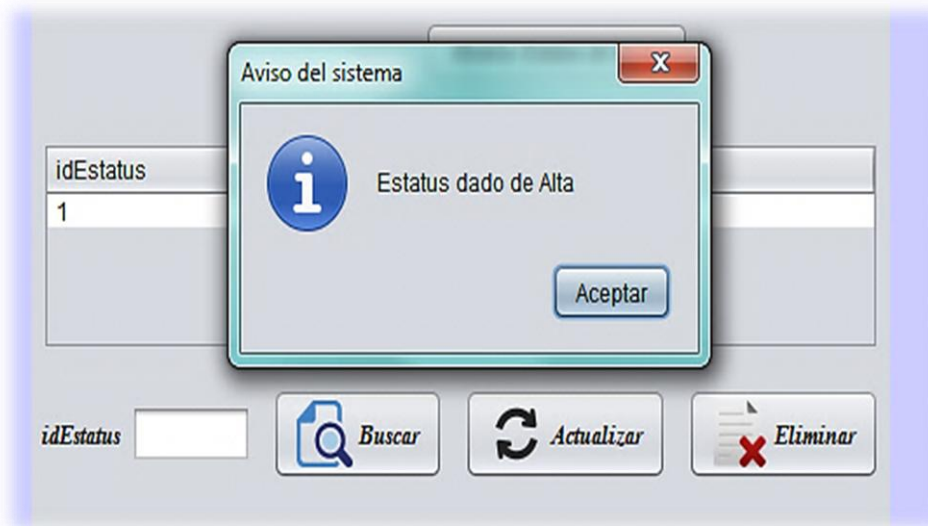


Figura 59. Validación de inserción de datos a estatus. (Autoría propia)

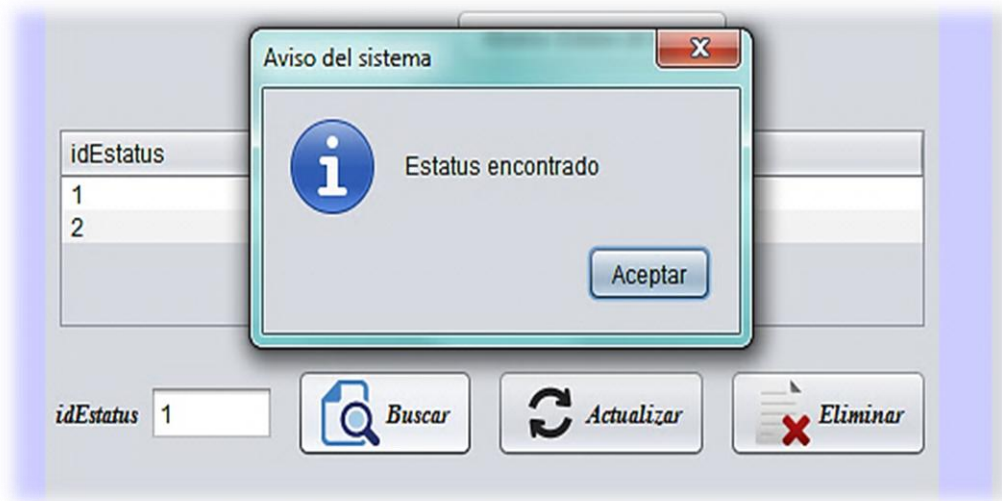


Figura 60. Validación de consulta de datos a estatus. (Autoría propia)

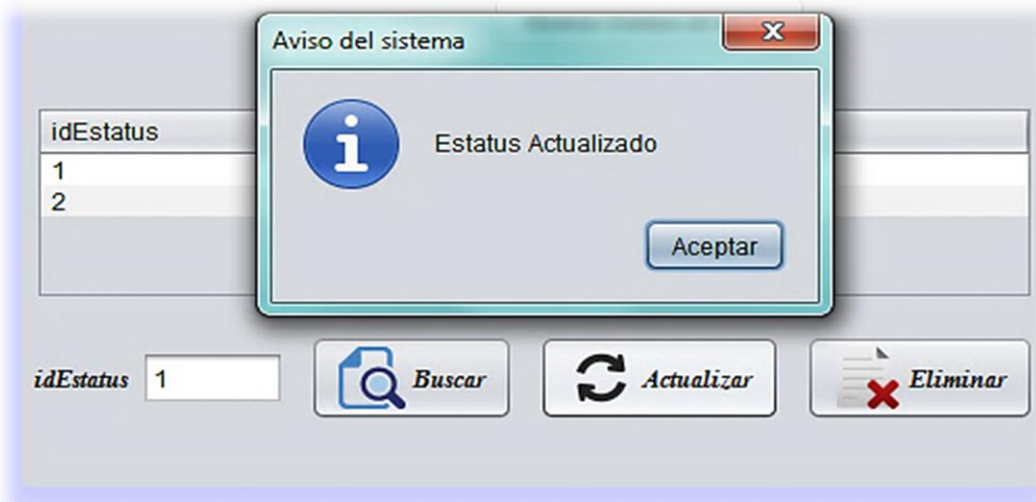


Figura 61. Validación de actualización de datos a estatus. (Autoría propia)

Interfaz generar orden

En esta interfaz se comprobó que el id se genere automáticamente y que las fechas se guarden de manera correcta, de igual manera se comprobó que la orden se almacene en la base de datos. Asimismo, en la figura 60 se puede validar lo comentado anteriormente.

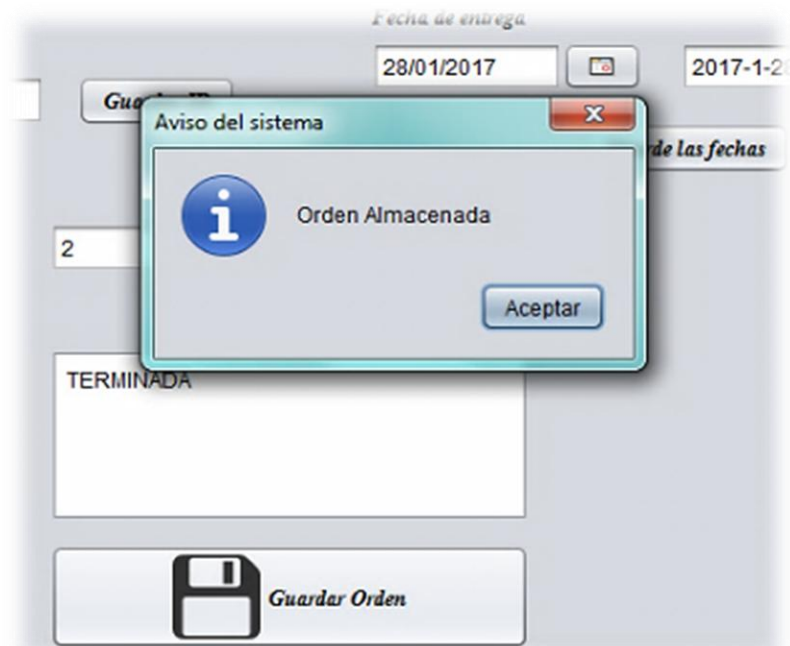
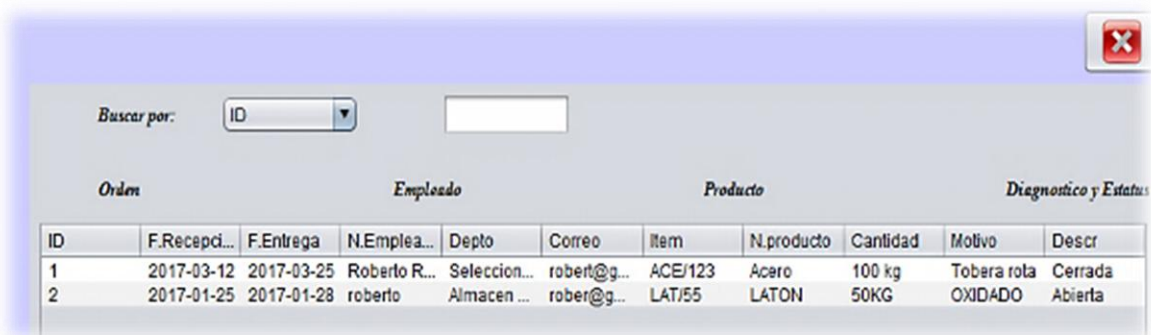


Figura 62. Validación de inserción de datos a orden. (Autoría propia)

Interfaz consultas

En la interfaz consulta se comprobó que los métodos de búsqueda se ejecuten de manera correcta es decir, se comprobó que la consultas de órdenes almacenadas funcionen aceptablemente así como la exportación de órdenes (ver figuras 61^a 64).



Buscar por: ID

Orden		Empleado			Producto			Diagnostico y Estatus		
ID	F.Recepci...	F.Entrega	N.Emplea...	Depto	Correo	Item	N.producto	Cantidad	Motivo	Descr
1	2017-03-12	2017-03-25	Roberto R...	Seleccion...	robert@g...	ACE/123	Acero	100 kg	Tobera rota	Cerrada
2	2017-01-25	2017-01-28	roberto	Almacen ...	rober@g...	LAT/55	LATON	50KG	OXIDADO	Abierta

Figura 63. Entidad ordenes con conexión a SQL. (Autoría propia)



Buscar por: N.Empleado

Orden		Empleado			Producto			Diagnostico y Estatus		
ID	F.Recepci...	F.Entrega	N.Emplea...	Depto	Correo	Item	N.producto	Cantidad	Motivo	Descr
2	2017-01-25	2017-01-28	roberto	Almacen ...	rober@g...	LAT/55	LATON	50KG	OXIDADO	Abierta

Figura 64. Validación de métodos de búsqueda de datos a órdenes. (Autoría propia)



Figura 65. Validación de exportación de datos a Excel. (Autoría propia)

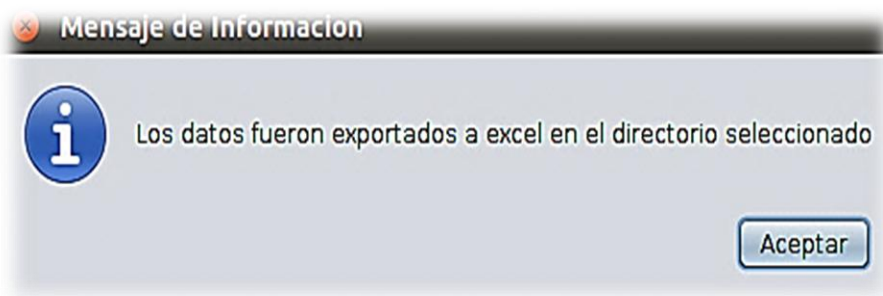


Figura 66. Mensaje de validación de exportación de datos. (Autoría propia).

7.3.3 Pruebas funcionales

Mediante las pruebas realizadas anteriormente, se logró visualizar que cada una de las interfaces por independiente ejecutan las acciones de manera correcta, también se obtuvieron buenos resultados, para cada una de las opciones individuales en cada interfaz. Por consiguiente, estas pruebas de funcionalidad ayudaron a corroborar el óptimo desempeño de las funciones de todas las interfaces.

7.3.4 Pruebas de integración

Como se logró comprobar anteriormente en cada una de las interfaces se observa que sus funciones se manipulan y visualizan de forma correcta, ahora se integran todos los módulos y existe una relacionan entre sí. Asimismo, para el caso del módulo *login*, existe una opción (inicio) donde dependiendo del usuario, el sistema accede como primera instancia a su sesión y posteriormente al módulo correspondiente para interactuar de manera directa con los formularios pertinentes.

7.3.5 Pruebas de validación

Estas pruebas se llevaron a cabo para todas las interfaces de manera independiente. Sin embargo en el caso de la interfaz de *login* se comprobó el uso de acceso para cada módulo o sesión dependiendo del usuario en cuestión.

7.3.6 Prueba de aceptación

Mediante las pruebas realizadas al sistema de información, fue necesario modificarlo hasta su correcta funcionalidad. Una vez validado por el supervisor del almacén de materia prima se dio por hecho la aceptación. De igual manera con las distintas pruebas realizadas se modificaron los métodos de búsqueda y de manipulación de información esto con la finalidad de que el usuario final tenga un sistema funcional.

7.4 Entrega

Al comprobarse que se cumplan los requerimientos establecidos y que no existan errores en base a las pruebas llevadas a cabo, se considera que el sistema está listo para ser usado, por lo que se hace de manera formal la entrega de la siguiente manera:

Nombre de usuario y contraseña de supervisor. Esta información se entrega al almacén de materia prima específicamente al supervisor el cual será el encargado de administrar el sistema para poder tener acceso así como poder utilizar las diferentes funciones del mismo.

Cuentas de usuario del servidor. Esta información también será entregada al departamento de materia prima y permite dar acceso a la base de datos así como al servidor donde se aloja el sistema.

Código del sistema. Todos los archivos que conforman el sistema, se encuentran alojados en el servidor, por lo tanto estos pueden ser intervenidos para visualizar el código y llevar a cabo modificaciones en caso de que el almacén de materia prima lo requiera.

Base de datos. La estructura y registros que conforman la base de datos del sistema, también se encuentran alojados en el servidor, por lo que se puede acceder a ella para hacer modificaciones en su estructura así como consultas y obtener información específica o eliminar registros.

7.5 Capacitación

Siempre que se introduce un sistema nuevo es necesario que todos aquellos usuarios que van a utilizarlo sean capacitados e instruidos. Para que la capacitación sea clara y el usuario pueda hacer uso del sistema sin muchos problemas, es conveniente desarrollar un plan de capacitación, mi plan consiste en:

Capacitación del supervisor: Fue el primer usuario en recibir la capacitación y se llevó a cabo por medio de una presentación por computadora, en la cual se explicó el funcionamiento del sistema posteriormente se realizó una explicación asistida donde se expuso cómo ejecutar el sistema, acceder al sistema, capturar la información, subir documentos, crear los reportes así como administrar usuarios.

Capacitación a empleados: Se explicó el funcionamiento del sistema por medio de una presentación por computadora, posteriormente se tomó el mismo programa educativo para llevar a cabo la explicación asistida a los empleados designados por el supervisor, donde se expuso cómo ejecutar el sistema, llenar formularios, capturar la información y subir documentos.

VIII RESULTADOS

8.1 Presentación de resultados

Con la implementación del sistema de información en el almacén de materia prima se obtuvieron los siguientes resultados:

- *Control y gestión de archivos e información clasificada.* Con la implementación de *INTELEISIS 13* se generó un mayor control de la información de *scrap* que se produce en los distintos departamentos de la empresa. Esto mediante una clasificación y organización de la información así como la efectividad en operaciones de consulta, actualización y eliminación de datos por consiguiente, las mermas ahora son clasificadas como desperdicio o reutilización de material de esta manera disminuyeron las pérdidas en el proceso productivo.
- *Ahorro de tiempo.* El sistema de información permite al usuario un acceso rápido a la información que se maneja en la base de datos por medio de una interfaz amigable así como métodos de consulta con los que cuenta el sistema además, de organizar toda la información de *scrap* rápidamente en un reporte el cual podrá ser impreso si el usuario así lo desea. De esta manera los usuarios del sistema trabajaran rápidamente y sin problemas generando en el almacén de materia prima eficiencia y eficacia.
- *Seguridad.* La seguridad en la información incremento bastante esto mediante la implementación de restricciones de seguridad en *INTELEISIS 13*, ahora cada usuario que quiera ingresar al sistema debe validarse esto con el fin de evitar el robo o alteración de la información lo cual sería una pérdida económica para la empresa.
- *Implementación del sistema de información,* La implementación de *INTELEISIS 13* se logró con satisfacción mediante la instalación de *MySQL sever* con el fin de crear un usuario que administre y genere una conexión correcta con la base de datos, después se instaló el gestor de base de datos *MySQL workbench* para administrar la base de datos en caso de cualquier problema posteriormente se instaló el *JDK 8* para que soporte la aplicación y por último se instaló el ejecutable de java en la computadora del almacén de materia prima de la empresa Goni.

IX DISCUSIÓN

Finalmente para el correcto manejo del sistema o mayor optimización se muestran las siguientes recomendaciones:

- Para asegurar el correcto funcionamiento del sistema a largo plazo es recomendable la contratación de un DBA (administrador de base de datos) que se encargue de la gestión de la base de datos y mantenimiento del sistema lo cual involucra el control de usuarios y contraseñas, asignación de roles, permisos y fijación de parámetros
- Se recomienda que los equipos de cómputo se actualicen periódicamente de acuerdo a las necesidades del software como el JDK o Framework.
- Para tener un mejor control de la información de *scrap* que ingresa al sistema se recomienda unificar INTELESIS 13 con el sistema de la empresa.
- La base de datos tendrá que actualizarse de acuerdo a las nuevas necesidades que surjan en el almacén de materia prima, para esto es recomendable programar un nuevo incremento en base a la metodología para atender las necesidades.
- *Resolución de la problemática que genera scrap.* Mediante entrevistas con el personal del almacén de materia prima se determinó que el acumulo de *scrap* se debe a un mal control en el departamento de calidad así como un incorrecto uso de la maquinaria.

X CONCLUSIONES

En base a las entrevistas realizadas al personal del almacén de materia prima se determinó que el acumulo de *scrap* se debe a la mala organización en el departamento de control de calidad así como un mal desempeño de los operadores de maquinaria.

Una vez determinada la causa del acumulo de merma se desarrolló e implemento un sistema de información bajo la metodología de desarrollo incremental el cual clasifica de manera organizada la información, esto mediante la implementación una base de datos. Por consiguiente, Aumento el control de *mermas* producidas en la empresa además, cada una de las operaciones como consulta, actualización y eliminación de información funcionan eficientemente.

Un objetivo fundamental fue generar un ahorro de tiempo, para lograr esto el usuario debe estar cómodo con la manipulación de *INTELESIS 13* por consiguiente, este sistema incorpora una interfaz “amigable” que contiene una comunicación funcional y técnica muy fácil de manejar por los usuarios de esta manera se obtuvo un ahorro de tiempo en el manejo de la información, debido a que anteriormente un informe de *scrap* se realizaba en 45 minutos ahora se genera en tan solo 10 minutos.

Cabe mencionar que antes de la implementación del sistema, la información no contaba con ningún método de seguridad por lo que cualquier persona con intenciones de manipular la información podía hacerlo, ahora el sistema de información mantiene sobre resguardo todos los datos de *scrap*, incrementado la confidencialidad, integridad y disponibilidad de la información de esta manera se disminuyeron las pérdidas de información, espionaje, alteraciones así como fraudes, esto mediante la creación de usuarios y roles para restringir el acceso a datos.

En definitiva podemos decir que el sistema creado conjunta de una manera sencilla diversas herramientas, que fueron capaces de crear soluciones a los problemas existentes en cuanto al manejo de grandes cantidades de información dentro del almacén de materia prima es decir, se implementó un sistema eficiente y confiable.

XI BIBLIOGRAFÍA

- Alcalde, A. (2016). *Diseño de Bases de Datos (II) - Restricciones*. [Blog Post] Obtenido de <https://elbauldelprogramador.com/disen-de-bases-de-datos-ii/>
- Bagui, S. (2011). *Database Design Using Entity-Relationship Diagrams*. Boca Raton, Florida: Second Edition, Auerbach Publications.
- Bellas, F.(2006). *Introduccion al diseño con patrones. Departamento de tecnologías de la informacion y las comunicaciones, Universidad de A Coruña* [en línea]. [fecha de consulta: 18 Octubre 2016] Disponible en: <http://www.tic.udc.es/~fbellas/teaching/pfc3/IntroPatrones.pdf>.
- Belmonte, O. (2005). *Introducción al lenguaje de programación java* [en línea]. [fecha de consulta: 05 Enero 2017] Obtenido de <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>
- Berzal, F. (2008). *Introduccion a la programación: java*. [en línea]. [fecha de consulta: 05 Enero 2017]Obtenido de <http://elvex.ugr.es/decsai/java/pdf/2B-Java.pdf>
- Castellanos, M. (2013). *Verano patrones de diseño*. [Blog Post] Obtenido de <http://castellanosmiguel.blogspot.mx/2013/07/dataaccess-object-dao-definicion-dao-es.html>
- Castro, G. (2010). *Sistemas de informacion*. [Blog Post]. Obtenido de <http://simarketingcm.blogspot.mx/>
- Cobo, A. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Santander, España: Ediciones Díaz de Santos.
- Cristian. (2010). *Ventajas Y Desventajas de los Modelos de Base de Datos*. [Blog Post] Obtenido de <http://ticcristianguerrero.blogspot.mx/2010/01/ventajas-y-desventajas-de-los-modelos.html>
- Denis, F. (2013). *informatica ventajas y desventajas de ubuntu*. [Blog Post] Obtenido de <http://informatica-denisfabiantn.blogspot.mx/2012/05/ventajas-y-desventajas-de-ubuntu.html>
- Díaz, M. (2009). *Características de los sistemas de información que permiten la gestión oportuna de la información y el conocimiento institucional*. [en línea]. [fecha de consulta: 05 Enero 2017] Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009001100006
- Fisher, M., Ellis, J., & Bruce, J. (2005). *JDBC API Tutorial and Reference, Third Edition*. Santa clara, California: Addison Wesley.

- Gomez, A., & Suarez, C. (2009). *Sistemas de informacion. herramientas practicas para la gestion. 3ª edicion.* ra-ma editorial. [en línea]. [fecha de consulta: 05 Enero 2017] Obtenido de http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm
- Gomez, M. d. (2013). *Notas del curso bases de datos* . [en línea]. [fecha de consulta: 05 Enero 2017] Obtenido de Universidad autonoma metropolitana: http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf
- Gonzalez, D. (2013). *Base de datos* [Blog Post]. Obtenido de http://davidgonzalezbd.blogspot.mx/2013/02/veterinaria_4.html
- Grant, R. (2006). *Ubuntu Linux for Non-Geeks: A Pain-Free.* Phoenix, AZ, U.S.A: No Starch Press.
- Henao, C. (2013). *patron delegate.* [Blog Post]. Obtenido de <http://codejavu.blogspot.mx/2013/06/ejemplo-patron-delegate.html>
- Java, (s.f.). Conozca más sobre la tecnología Java. Recuperado el 21 septiembre de 2016, de <http://www.java.com/es/about/>
- Kendall, K. y. (2005). *Análisis y diseño de sistemas. Sexta edición.* Ciudad de Mexico, México: PEARSON EDUCACIÓN.
- Laudon, K. C., & Laudon, J. P. (2012). *Sistemas de información gerencial.* Ciudad de Mexico, México: PEARSON EDUCACIÓN.
- Luna, L. *El diseño de interfaz Gráfica de usuario para publicaciones digitales. Revista UNAM* [en línea]. 10 de agosto 2005, Volumen 5, Número 7.[fecha de consulta: 8 Octubre 2016] Disponible en: http://www.revista.unam.mx/vol.5/num7/art44/ago_art44.pdf
- MSDN (2010). *Data Transfer Object. Microsoft MSDN Library.* Recuperado de: <http://msdn.microsoft.com/en-us/library/ms978717.aspx>
- Patrizio, A. (2015). *Java is Dead? Not According to These Enterprises.* [Blog Post]. Obtenido de <http://www.datamation.com/applications/java-is-dead-not-according-to-these-enterprises.html>
- Pressman, R. S. (2010). *Ingeniería del Software : un enfoque práctico.* Ciudad de Mexico, Mexico: McGraw-Hill.
- Ramos, A. (2007). *Operaciones con bases de datos ofimáticas y corporativas.* [en línea]. [fecha de consulta: 05 Enero 2017] Editorial Paraninfo.
- Rob, P., & Coronel, C. (2007). *Database Systems: Design, Implementation, and Management.* USA: Cengage Learning.

- Saiz, A. (2013). *Que importancia tienen las bases de datos a nivel empresarial*. [Blog Post]. Obtenido de <http://datacentric.es/blog/index.php/importancia-bases-de-datos-2/>
- Sanchez, J. (2009). *Lenguaje Java Fundamentos de Programacion*. [Blog Post]. Obtenido de <https://jorgesanchez.net/>
- Sanchez, j. (2016). *Manual de gestion de base de datos*. [Blog Post]. Obtenido de Modelo relacional: <https://jorgesanchez.net/manuales/gbd/modelo-relacional.html>
- Selenium, E. (2016). *How to Set Data into Excel sheet using jxl*. [En línea] Obtenido de <http://www.seleniumeasy.com/jxl-tutorials/set-data-into-excelsheet-with-jxl>
- Silberschatz, A., & Korth, H. F. (2006). *Fundamentos de bases de datos Quinta Edicion*. Madrid, España: McGraw-Hill.
- Sommerville, I. (2005). *Ingeniería del software*. [en línea]. [fecha de consulta: 05 Enero 2017]. Pearson Educación.
- Sun Microsystems (2007). *Data Access Objects. Core J2EE Patterns*. Recuperado de: <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>
- Tanenbaum, A. S. (2009). *Sistemas operativos modernos. Tercera edicion*. Ciudad de Mexico, Mexico: Pearson Educacion.
- Thesus de Mexico, S. D. (s.f.). *Goni equipo profesional*. Obtenido de <http://www.goni.mx/>
- Vp, Betty. (2013). *Patrones de diseño, bussines delegate* [Blog Post]. Obtenido de <http://patronesdisenoo.blogspot.mx/2013/07/business-delegate.html>
- Zamora, S. (2016). *Actividades basicas de un sistema de informacion*. [En línea] Obtenido de <http://izamorar.com/actividades-basicas-de-un-sistema-de-informacion/>

ANEXOS

Anexo (A) Cuestionario de requerimientos de software

Este cuestionario se aplicó en el departamento de materia prima con el fin de conocer las necesidades del software a satisfacer.

- ¿Cuál es la actividad económica de su empresa?
- ¿Cómo está compuesta su empresa?
- ¿Cómo manejan la información?
- ¿Qué tipo de información manejan?
- ¿Quiénes tienen acceso a la información?
- ¿Qué modalidades utilizan para archivar la información?
- ¿Cada cuánto actualizan los registros de la información almacenada?
- ¿Cuál es la problemática que buscan solucionar a través del sistema de información?
- ¿Quiénes tendrán acceso al sistema de información?
- ¿Cuáles serán los límites de ingreso al sistema de información según el cargo de los empleados?
- ¿Se ha implementado anteriormente un sistema parecido?
- ¿Qué dependencias de la empresa necesitaran utilizar el sistema de información?