

INSTITUTO TECNOLÓGICO DE CD. GUZMÁN

**TITULACIÓN INTEGRAL
TESIS**

**TEMA:
ANÁLISIS Y DISEÑO DE UNA PLATAFORMA DE
GESTIÓN EN UN ENTORNO DE DESARROLLO
INTEGRADO.**

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO INFORMÁTICO**

PRESENTA:

FRANCISCO JAVIER ALEJO PALACIOS

**ASESORA:
MC. RAQUEL OCHOA ORNELAS**

CD. GUZMÁN JALISCO, MÉXICO, MAYO DE 2017



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

Cd. Guzmán, Municipio de Zapotlán el Grande, Jal, **23/mayo/2017**

ASUNTO: Liberación de Proyecto para Titulación Integral.


M.C. FAVIO REY LUA MADRIGAL
JEFE DE LA DIVISION DE ESTUDIOS PROFESIONALES
PRESENTE

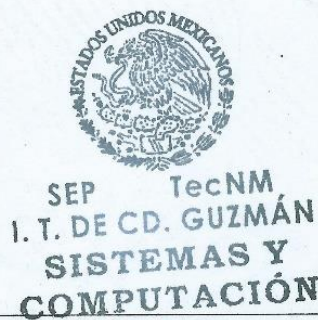
Por este medio le informo que ha sido liberado el siguiente proyecto para la Titulación Integral:

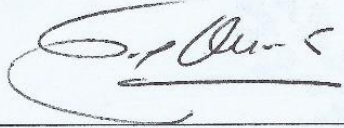

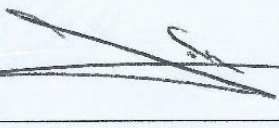
Nombre del Egresado:	Francisco Javier Alejo Palacios
Carrera:	Ingeniería Informática
No. De Control:	12290672
Nombre del Proyecto:	"Análisis y Diseño de una Plataforma de Gestión en un Entorno de Desarrollo Integrado"
Producto:	TESIS

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE


M.C. RUBÉN ZEPEDA GARCÍA
JEFE DEL DEPTO. SISTEMAS Y COMPUTACIÓN



M.C. RAQUEL OCHOA ORNELAS 	M.C. LEONARDO ALCARAZ SILVA 	M.C. JORGE ALFONSO ARAGÓN HOPE 
Nombre y Firma del Asesor	Nombre y Firma del Revisor	Nombre y Firma del Revisor

c.c.p. Archivo
JRGV/JMTS/RZG/lgo



ÍNDICE

ÍNDICE	i
LISTA DE FIGURAS	v
LISTA DE TABLAS.....	vii
SIGLAS	viii
TÉRMINOS	ix
CAPÍTULO 1. INTRODUCCIÓN.....	1
CAPÍTULO 2. PLANTEAMIENTO DE LA INVESTIGACIÓN.....	2
2.1. JUSTIFICACIÓN DE LA INVESTIGACIÓN.	2
2.2. PROBLEMA DE INVESTIGACIÓN.....	3
2.3. OBJETIVOS GENERALES Y ESPECÍFICOS.	3
2.3.1. Objetivo General	3
2.3.2 Objetivos Específicos.....	3
CAPÍTULO 3. REVISIÓN DE LITERATURA.....	4
3.1. UML	4
3.1.1 Diagrama de casos de uso.....	6
3.1.2 Diagrama de objetos (estático)	7
3.1.3 Diagrama de clases.....	8
3.1.4 Diagrama de paquetes	9
3.1.5 Diagrama de componentes	9
3.1.6 Diagrama de despliegue	10
3.1.7 Diagrama de estructuras compuestas.....	11
3.1.8 Diagrama de secuencia.....	11
3.1.9 Diagrama de comunicación (o de colaboración)	12
3.1.10 Diagrama de máquina de estados o de estados	13
3.1.11 Diagrama de actividades.....	14
3.1.12 Diagrama de visión global de la interacción	15
3.1.13 Diagrama de tiempos	16
3.2 BASES DE DATOS	18
3.2.1 Principios de una base de datos	18
Normalización	18
Modelo Entidad/Relación	18
Atributos	19

Formas Normales.....	19
3.3 SQL SERVER.....	20
Sentencias DDL.....	21
Sentencia GRANT.....	21
Sentencia CREATE.....	21
Sentencia ALTER.....	22
Sentencia DROP.....	24
Sentencia DENY.....	25
Sentencia TRUNCATE.....	26
Sentencias DML.....	27
Sentencia INSERT.....	27
Sentencia SELECT.....	28
Sentencia DELETE.....	33
SENTENCIAS DE CONTROL DE FLUJO.....	37
CAPÍTULO 4. METODOLOGÍA.....	38
4.1 ANÁLISIS.....	38
4.1.1 Antecedentes.....	38
4.1.2 Entradas.....	38
4.1.3 Salidas.....	38
4.1.4 Políticas.....	39
4.1.5 Entidades externas.....	39
4.1.6 Procesos.....	39
4.1.7 Almacenes de datos.....	40
4.1.8 Diagramas del sistema de información.....	41
4.1.9 Diagrama de flujo de datos de registrar compra.....	46
4.1.10 Diagrama de estados de registrar compra.....	47
4.1.11 Diagrama de colaboración de registrar compra.....	49
4.1.12 Diagrama de caso de uso expandido de registrar compra.....	49
4.1.13 Diagrama de flujo de datos de registrar pago.....	50
4.1.14 Diagrama de estados de registrar pago.....	52
4.1.15 Diagrama de colaboración de registrar pago.....	54
4.1.16 Diagrama de caso de uso expandido de registrar pago.....	54
4.1.17 Diagrama de flujo de datos de registrar venta.....	55
4.1.18 Diagrama de estados de registrar venta.....	57

4.1.19 Diagrama de colaboración de registrar venta.....	59
4.1.20 Diagrama de caso de uso expandido de registrar venta	60
4.1.21 Diagrama de flujo de datos de registrar abono	61
4.1.22 Diagrama de estados de registrar abono	62
4.1.23 Diagrama de colaboración de registrar abono	64
4.1.24 Diagrama de caso de uso expandido de registrar abono	65
4.1.25 Diagrama de flujo de datos de registrar pedido.....	66
4.1.26 Diagrama de estados de registrar pedido	67
4.1.27 Diagrama de colaboración de registrar pedido.....	69
.....	69
4.1.28 Diagrama de caso de uso expandido de registrar pedido	70
4.1.29 Diagrama de flujo de datos de registrar gastos de producción	71
4.1.30 Diagrama de estados de registrar gastos de producción	72
4.1.31 Diagrama de colaboración de registrar gastos de producción	74
4.1.32 Diagrama de caso de uso expandido de registrar gastos de producción	75
4.1.33 Diagrama de flujo de datos de registrar material sobrante de producción	76
4.1.34 Diagrama de estados de registrar material sobrante de producción. ..	77
4.1.35 Diagrama de colaboración de registrar material sobrante de producción	79
4.1.36 Diagrama de caso de uso expandido de registrar material sobrante de producción.....	80
4.1.37 Diagrama de flujo de datos de registrar productos terminados	81
4.1.38 Diagrama de estados de registrar productos terminados.....	82
4.1.39 Diagrama de colaboración de registrar productos terminados	84
4.1.40 Diagrama de caso de uso expandido de registrar productos terminados	84
4.1.41 Diagrama de flujo de datos de registrar baja de personal	86
4.1.42 Diagrama de estados de registrar baja de personal.....	87
4.1.43 Diagrama de colaboración de registrar baja de personal	87
4.1.44 Diagrama de caso de uso expandido de registrar baja de personal....	88
4.1.45 Diagrama de flujo de datos de reactivar personal	90
4.1.46 Diagrama de estados de reactivar personal.....	91
4.1.47 Diagrama de colaboración de reactivar personal	92
4.1.48 Diagrama de caso de uso expandido de reactivar personal.....	92

4.2. DISEÑO	94
4.2.1 Normalización	94
CAPÍTULO 5. RESULTADOS	105
5.1 Diagrama de clases	105
5.2 Diccionario de datos	107
5.3 Diseño de interfaces	122
5.4 Diseño de Informes.....	128
CONCLUSIONES.....	133
REFERENCIAS BIBLIOGRÁFICAS	134

LISTA DE FIGURAS

Figura 1. Estructura de una clase en UML.	ix
Figura 2. Caso de uso: Cenar ahora.	6
Figura 3. Ejemplo de diagrama de objetos.	8
Figura 4. Ejemplo de diagrama de clases.	8
Figura 5. Ejemplo de diagrama de paquetes.	9
Figura 6. Ejemplo de diagrama de componentes.	10
Figura 7. Ejemplo de diagrama de despliegue.	10
Figura 8. Ejemplo de diagrama de estructuras compuestas.	11
Figura 9. Ejemplo de diagrama de secuencia.	12
Figura 10. Ejemplo de diagrama de colaboración.	13
Figura 11. Ejemplo de diagrama de máquina de estados o de estados.	14
Figura 12. Ejemplo de diagrama de actividades.	15
Figura 13. Ejemplo de diagrama de interacción.	16
Figura 14. Ejemplo de diagrama de tiempos.	17
Figura 15. Diagrama de Warnier/Orr. Parte 1.	42
Figura 16. Diagrama Warnier/Orr. Parte 2.	43
Figura 17. Diagrama de casos de uso general.	44
Figura 18. Diagrama de contexto.	45
Figura 19. Diagrama de flujo de datos: Registrar compra.	46
Figura 20. Diagrama de estados: Registrar compra.	48
Figura 21. Diagrama de colaboración: Registrar compra.	49
Figura 22. Diagrama de caso de uso expandido: Registrar compra.	50
Figura 23. Diagrama de flujo de datos: Registrar pago.	51
Figura 24. Diagrama de estados: Registrar pago.	53
Figura 25. Diagrama de colaboración: Registrar pago.	54
Figura 26. Diagrama de caso de uso expandido: Registrar pago.	55
Figura 27. Diagrama de flujo de datos: Registrar venta.	56
Figura 28. Diagrama de estados: Registrar venta.	58
Figura 29. Diagrama de colaboración: Registrar venta.	59
Figura 30. Diagrama de caso de uso expandido: Registrar compra.	60
Figura 31. Diagrama de flujo de datos: Registrar abono.	61
Figura 32. Diagrama de estados: Registrar abono.	63
Figura 33. Diagrama de colaboración: Registrar abono.	64
Figura 34. Diagrama de caso de uso expandido: Registrar abono.	65
Figura 35. Diagrama de flujo de datos: Registrar pedido.	66
Figura 36. Diagrama de estados: Registrar pedido.	68
Figura 37. Diagrama de colaboración: Registrar pedido.	69
Figura 38. Diagrama de caso de uso expandido: Registrar pedido.	70
Figura 39. Diagrama de flujo de datos: Registrar gastos de producción.	71
Figura 40. Diagrama de estados: Registrar gastos de producción.	73
Figura 41. Diagrama de colaboración: Registrar gastos de producción.	74
Figura 42. Diagrama de caso de uso expandido: Registrar gastos de producción.	75
Figura 43. Diagrama de flujo de datos: Registrar material sobrante de producción.	76

Figura 44. Diagrama de estados: Registrar material sobrante de producción.....	78
Figura 45. Diagrama de colaboración: Registrar material sobrante de producción.	79
Figura 46. Diagrama de caso de uso expandido: Registrar material sobrante de producción.....	80
Figura 47. Diagrama de flujo de datos: Registrar productos terminados.....	81
Figura 48. Diagrama de estados: Registrar productos terminados.	83
Figura 49. Diagrama de colaboración: Registrar productos terminados.....	84
Figura 50. Diagrama de caso de uso expandido: Registrar productos terminados.	85
Figura 51. Diagrama de flujo de datos: Registrar baja de personal.....	86
Figura 52. Diagrama de estados: Registrar baja de personal.	87
Figura 53. Diagrama de colaboración: Registrar baja de personal.....	88
Figura 54. Diagrama de caso de uso expandido: Registrar baja de personal.	89
Figura 55. Diagrama de flujo de datos: Reactivar personal.....	90
Figura 56. Diagrama de estados: Reactivar personal.	91
Figura 57. Diagrama de colaboración: Reactivar personal.....	92
Figura 58. Diagrama de caso de uso expandido: Reactivar personal.	93
Figura 59. Diagrama de clases. Parte 1.	105
Figura 60. Diagrama de clases. Parte 2.	106
Figura 61. Menú principal del sistema.....	122
Figura 62. Catálogo de clientes.....	123
Figura 63. Catálogo de productos.	123
Figura 64. Catálogo de proveedores.....	124
Figura 65. Catálogo de materiales.	124
Figura 66. Catálogo de personal.	125
Figura 67. Catálogo de categorías.	125
Figura 68. Catálogo de unidades.	126
Figura 69. Submenú de procesos.	126
Figura 70. Pantalla principal de registrar ventas.	127
Figura 71. Pantalla principal de registrar abonos.	127
Figura 72. Reporte general de personal.....	128
Figura 73. Reporte general de clientes.	129
Figura 74. Reporte general de clientes con adeudos.....	129
Figura 75. Reporte general de proveedores.....	130
Figura 76. Reporte general de materiales.	130
Figura 77. Reporte de productos por categoría.....	131
Figura 78. Reporte de compras de contado entre dos fechas.....	131
Figura 79. Reporte de ventas a crédito entre fechas.....	132

LISTA DE TABLAS

Tabla 1. Sentencias CREATE.	21
Tabla 2. Sentencias ALTER.	23
Tabla 3. Tabla de Personal.	107
Tabla 4. Tabla de BajaPersonal.	108
Tabla 5. Tabla de ReactivarBaja.	109
Tabla 6. Tabla de Proveedores.	110
Tabla 7. Tabla de Pagos.	111
Tabla 8. Tabla de Compras.	111
Tabla 9. Tabla de DetalleCompra.	112
Tabla 10. Tabla de Unidades.	113
Tabla 11. Tabla de Materiales.	113
Tabla 12. Tabla de Pedidos.	114
Tabla 13. Tabla de DetallePedido.	115
Tabla 14. Tabla de Produccion.	115
Tabla 15. Tabla de DetalleProduccion.	116
Tabla 16. Tabla de DetalleMateProdu.	116
Tabla 17. Tabla de DetalleSobrantes.	117
Tabla 18. Tabla de Clasificacion.	117
Tabla 19. Tabla de Productos.	118
Tabla 20. Tabla de Clientes.	119
Tabla 21. Tabla de Abonos.	120
Tabla 22. Tabla de Ventas.	121
Tabla 23. Tabla de DetalleVenta.	122

SIGLAS

UML: Unified Modeling Language - Lenguaje Unificado de Modelado (Fontela, 2011).

OMG: Object Management Group - Asociación de empresas de desarrollo de estándares (Fontela, 2011).

SQL: Structured Query Language - Lenguaje de Consultas Estructuradas (Rouse, SQL, 2016).

RDBMS: Sistema de Gestión de Bases de Datos Relacionales (Rouse, SQL Server, 2016).

DDL: Data Definition Language - Lenguaje de Definición de Datos (Delgado, 2016).

DML: Data Manipulation Language - Lenguaje de Manipulación de Datos (Delgado, 2016).

TÉRMINOS

UML: Es una notación de modelado visual, que utiliza diagramas para presentar diferentes aspectos de un sistema (Fontela, 2011).

Clase: Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar un entorno de estudio (un Computador, un Celular, un Auto, etc.). En UML, una clase se representa por un rectángulo con tres divisiones, así como se muestra en la Figura 1 (Díaz, 2010).

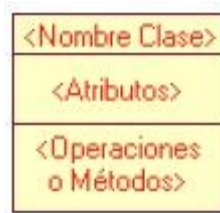


Figura 1. Estructura de una clase en UML.

Modelado Visual: Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual. El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar (Hernández, 2015).

Lenguaje: Es un sistema de comunicación que posee una estructura determinada de uso y contenido (Pérez & Merino, 2017).

Unificado: Hacer de muchas cosas una o un todo, uniendo, mezclando o reduciendo a una misma especie. Hacer que varias cosas distintas sean igual o semejante entre sí; hacerla uniforme (equiparar o uniformar) (BioDic, 2014).

CAPÍTULO 1. INTRODUCCIÓN.

Actualmente las empresas necesitan sistemas de información que les permita obtener información confiable, oportuna e íntegra para la toma de decisiones. Es por esto que el desarrollo de sistemas de información resulta necesario para optimizar los procesos y tareas administrativas de las empresas.

El sistema de información está diseñado para ofrecer el mejor control y administración en los procesos y tareas cotidianas de la empresa “Don Colchonero Muebles”, dicha empresa se dedica a la fabricación y comercialización de muebles. Por lo tanto la empresa requiere el mayor control en sus compras, pagos, ventas, abonos, producciones, pedidos, inventarios de materia prima y de productos terminados, entre otros. Para ello fue necesario analizar su problemática y diseñar un sistema de información a la medida, adaptado completamente a sus necesidades, para que pudiera satisfacer los requerimientos de la empresa en cuanto al control de sus ingresos y egresos de recursos materiales y monetarios.

Para el diseño del sistema de información se utilizó Microsoft Visual Studio Community 2015, en este IDE de desarrollo fueron planteadas cada una de las pantallas requeridas en el sistema de información y la base de datos fue creada en SQL Server 2014.

El presente informe incluye una descripción detallada de cada una de las actividades realizadas durante el análisis y diseño de dicho sistema. También incluye los resultados que se obtuvieron al finalizar el proyecto, así como el diseño de las pantallas del sistema.

CAPÍTULO 2. PLANTEAMIENTO DE LA INVESTIGACIÓN.

2.1. JUSTIFICACIÓN DE LA INVESTIGACIÓN.

La empresa “Don Colchonero Muebles” ha incrementado sus clientes en los últimos años, fabricando muebles sobre pedido así como produciendo colchones, salas y comedores para la venta en sucursales. Anteriormente no existía un sistema confiable para el seguimiento de los pedidos, la mano de obra utilizada en la producción y el control de materia prima, lo cual generaba inconsistencias al momento de obtener los precios de venta si los costos solo resultaban de una aproximada estimación. Por otra parte, el control de ventas se llevaba a través de una hoja de Excel, lo cual no resultaba eficiente ya que generaba inconsistencias, algunos errores y omisiones en los totales.

El inventario de materia prima así como el inventario de productos terminados requería atención, pues existía pérdida de mercancía que difícilmente se lograba rastrear. Por otra parte, en cuestión de materia prima, era necesario contar con un inventario exacto para tener los mínimos requeridos para la producción, ya que se atrasaban los procesos generando costos altos por el pago a mano de obra mientras se conseguían los materiales necesarios.

Las utilidades que se generaban de acuerdo al margen entre el precio y el costo de producción también eran importantes de atender, así como los pagos que se realizaban a los proveedores y los abonos de los clientes.

Durante esta investigación se recabaron todos los requerimientos funcionales y no funcionales del sistema. El análisis y diseño se fundamentó en UML y el análisis estructurado a través de diagramas de flujo de datos y Warnier Orr, implementado el diseño de la base de datos, formularios y reportes. La base de datos se realizó a través de SQL Server, diseñando además diversos procedimientos almacenados que fueron utilizados en consultas e informes a través de Visual Basic. Las fases de análisis y diseño de un sistema requieren especial atención, ya que resultan fundamentales para lograr un sistema de calidad, por lo que es de suma importancia

su estudio antes de implementar un sistema de información en un entorno de programación integrado como lo es Visual Studio Community 2015.

2.2. PROBLEMA DE INVESTIGACIÓN.

A través del análisis de los requerimientos de la empresa “Don Colchonero Muebles” generar un diseño funcional utilizando UML y el análisis estructurado que permita obtener una documentación eficiente para implementar un sistema de información óptimo que de respuesta a las necesidades especificadas.

2.3. OBJETIVOS GENERALES Y ESPECÍFICOS.

2.3.1. Objetivo General

Análisis y diseño de un sistema de información en un entorno de desarrollo integrado, que logre la administración de inventarios, gastos, ventas y compras a crédito y de contado de la empresa “Don Colchonero Muebles”, dedicada a la fabricación y comercialización de muebles.

2.3.2 Objetivos Específicos

- Gestionar el control de ventas.
- Mantener un control en las compras.
- Generar informes con el monto total de las ventas y compras por periodo.
- Emitir reportes para informar estados específicos.
- Controlar los pagos a los proveedores.
- Controlar los abonos de los clientes.
- Mantener un inventario actualizado de productos terminados.
- Administrar la existencia de los materiales.
- Mantener un registro actualizado del personal.

CAPÍTULO 3. REVISIÓN DE LITERATURA.

3.1. UML

UML (por sus siglas en inglés *Unified Modeling Language*, que en español significa Lenguaje Unificado de Modelado) es una notación de modelado visual, que utiliza diagramas para presentar diferentes aspectos de un sistema. Muchas veces se dice que UML es capaz de modelar cualquier sistema, pero su mayor extensión y sus principales virtudes están enfocadas al campo de los sistemas de software. Sin embargo esto no impide a que muchos profesionales usen o intenten utilizar UML en diversas situaciones (Fontela, 2011).

UML surgió en el año de 1995 por Grady Booch, James Rumbaugh e Ivar Jacobson, que habían creado sus propias notaciones de modelado. Precisamente, UML se define como “unificado”, porque surgió como síntesis de los mejores elementos de las notaciones ya creadas. Lo cual ha dado muy buenos resultados, ya que, en la década de 1990 había un caos por la falta de un estándar.

Después UML se estableció con más rigurosidad, en 1997 se presentó la versión 1.0, que fue aprobada y establecida como estándar por el OMG (Object Management Group, una asociación de empresas de desarrollo de estándares). Y siguió evolucionando, formalizándose, creciendo y haciéndose más complejo.

En el año 2000, UML ya se había convertido en el estándar para la modelización de software orientado a objetos.

Actualmente UML es un lenguaje de visualización, especificación y documentación de software, basado en trece tipos de diagramas, cada uno con sus objetivos, destinatarios y contexto de uso. Además, UML es extensible ya que se han establecido muchos mecanismos de extensibilidad, que permiten aumentar los usos de UML.

UML es bastante más amplio de lo que se llega a necesitar muchas veces. Los creadores de la notación llegan a afirmar que el 80% de la mayoría de los sistemas se pueden modelar con un 20% de las construcciones de UML (Fontela, 2011).

En UML 2.2 existen modelos estructurales y otros de comportamiento que se utilizan para modelar aspectos estructurales y de comportamiento, respectivamente, de las aplicaciones de software.

Los modelos estáticos o estructurales sirven para modelar el conjunto de objetos, clases, relaciones y sus agrupaciones, presentes en un sistema. También existen cuestiones dinámicas o de comportamiento que definen como evolucionan esos objetos a lo largo del tiempo, y cuáles son las causas de esa evolución (Fontela, 2011).

UML sirve para definir ambos tipos de modelos, esto es interesante por la interrelación substancial entre ambas cuestiones. UML trabaja con 13 tipos de diagramas.

Diagramas estructurales o estáticos de UML 2.2:

- Diagrama de casos de uso.

- Diagrama de objetos (estático).

- Diagrama de clases.

- Diagrama de paquetes.

- Diagrama de componentes.

- Diagrama de despliegue.

- Diagrama de estructuras compuestas.

Diagramas de comportamiento o dinámicos:

- Diagrama de secuencia.

- Diagrama de comunicación (o de colaboración).

- Diagrama de máquina de estados o de estados.

- Diagrama de actividades.

- Diagrama de visión global de la interacción.

- Diagrama de tiempos (Fontela, 2011).

Los diagramas estáticos o estructurales se encargan de definir que elementos (entidades, objetos, áreas, clases, etc.) deben estar definidas dentro del sistema y como deben estar estructuradas, mientras que los diagramas dinámicos se encargan de definir el comportamiento de los estáticos (Fernando, 2015).

3.1.1 Diagrama de casos de uso

En un diagrama de casos de uso no se muestran los casos de uso en detalle; solamente se resumen algunas de las relaciones entre los casos de uso, los actores y los sistemas. En concreto, en el diagrama no se muestra el orden en que se llevan a cabo los pasos para lograr los objetivos de cada caso de uso. Esos detalles pueden describirse en otros diagramas y documentos, que pueden vincularse a cada caso de uso.

Los casos de uso solamente se usan para los requisitos funcionales de un sistema. Otros requisitos, como las reglas de negocio, los requisitos de calidad del servicio y las restricciones de implementación, deben representarse por separado. La arquitectura y los detalles internos también deben describirse por separado (Corporation, 2016).

En la siguiente figura se muestra un ejemplo de un caso de uso (Figura 2).

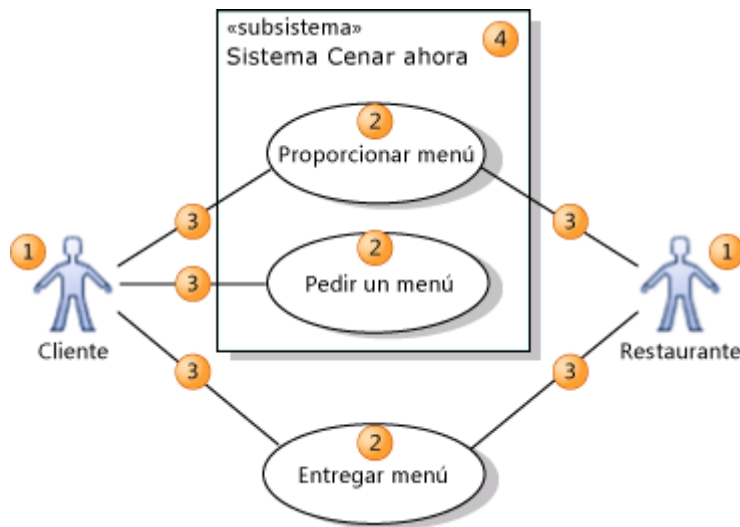


Figura 2. Caso de uso: Cenar ahora.

- Un actor (1) es una clase de persona, organización, dispositivo o componente de software externo que interactúa con el sistema. Los actores del ejemplo son cliente y restaurante.
- Un caso de uso (2) representa las acciones que uno o varios de los actores realizan a fin de conseguir un objetivo determinado.

En un diagrama de casos de uso, los casos de uso están asociados (3) a los actores que los realizan.

- El sistema (4) es aquello que se está desarrollando. Puede ser un pequeño componente de software cuyos actores simplemente son otros componentes de software; puede ser una aplicación completa; o puede ser un gran conjunto de aplicaciones distribuidas que se implementan en muchos equipos y dispositivos. Los subsistemas del ejemplo son “Sitio web de pedidos de menú”, “Empresa de entrega de menús” y “Versión 2 del sitio web”.

En un diagrama de casos de uso pueden mostrarse los casos de uso que el sistema o sus subsistemas admiten (Corporation, 2016).

3.1.2 Diagrama de objetos (estático)

Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma nombre de objeto el nombre de clase (Clara, 2015).

Muestran una serie de objetos (instancias de las clases) y sus relaciones. Estos diagramas se enfocan en la perspectiva de casos reales o prototipos. Es un diagrama de instancias de las clases mostradas en el diagrama de clases (Castillo, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de objetos (Figura 3).

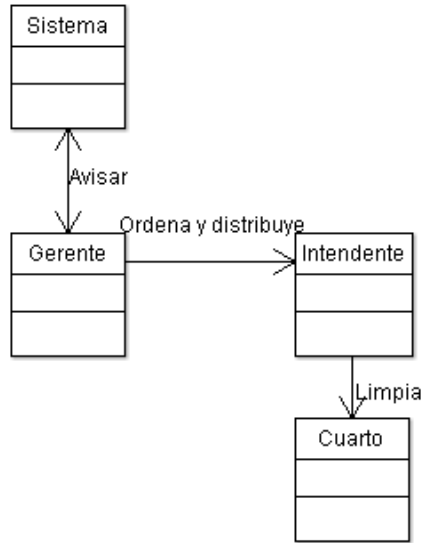


Figura 3. Ejemplo de diagrama de objetos.

3.1.3 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información del sistema (Clara, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de clases (Figura 4).

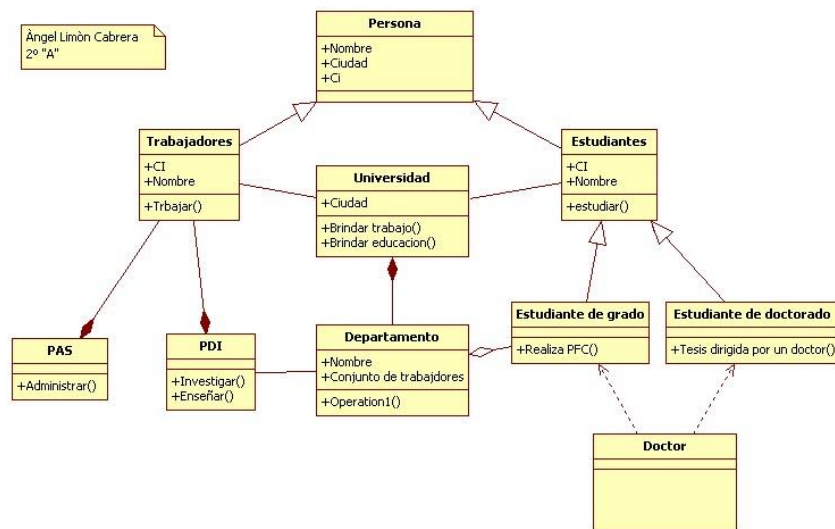


Figura 4. Ejemplo de diagrama de clases.

3.1.4 Diagrama de paquetes

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema (Clara, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de paquetes (Figura 5).

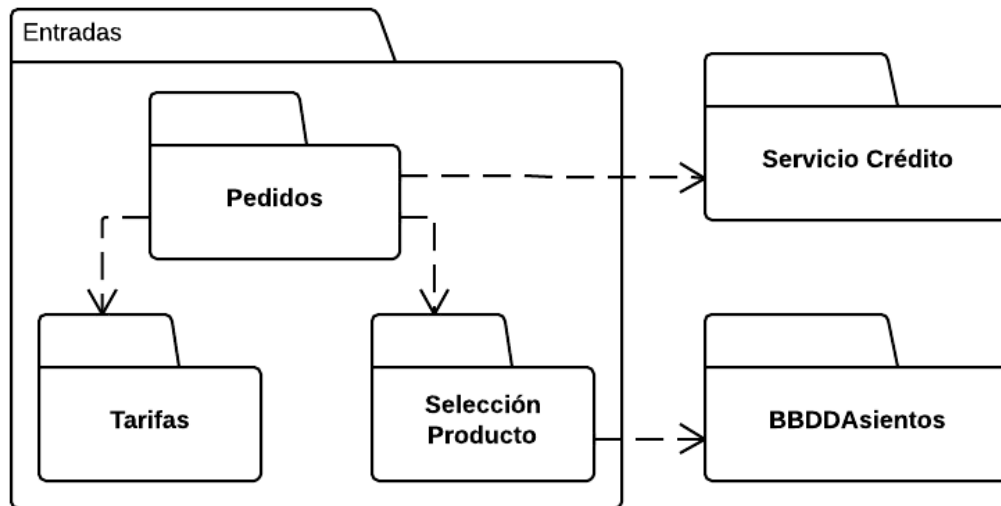


Figura 5. Ejemplo de diagrama de paquetes.

3.1.5 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes (Clara, 2015).

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Se usan para agrupar clases en componentes o módulos (Castillo, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de componentes (Figura 6).

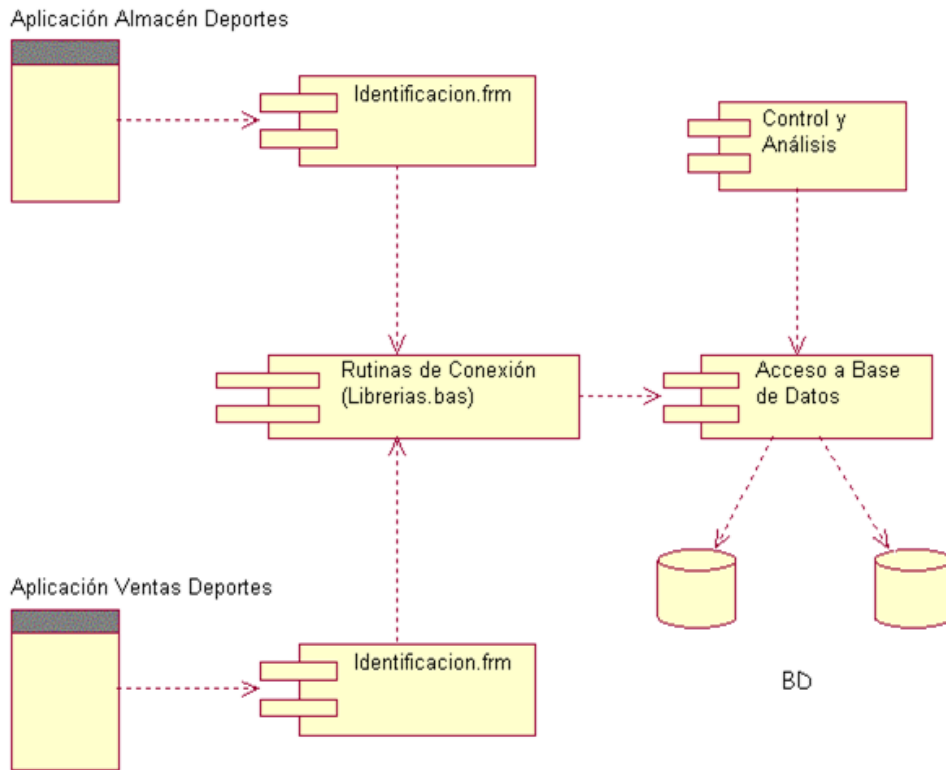


Figura 6. Ejemplo de diagrama de componentes.

3.1.6 Diagrama de despliegue

Un diagrama de despliegue muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo. Se utiliza para identificar Sistemas de Cooperación (Castillo, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de despliegue (Figura 7).

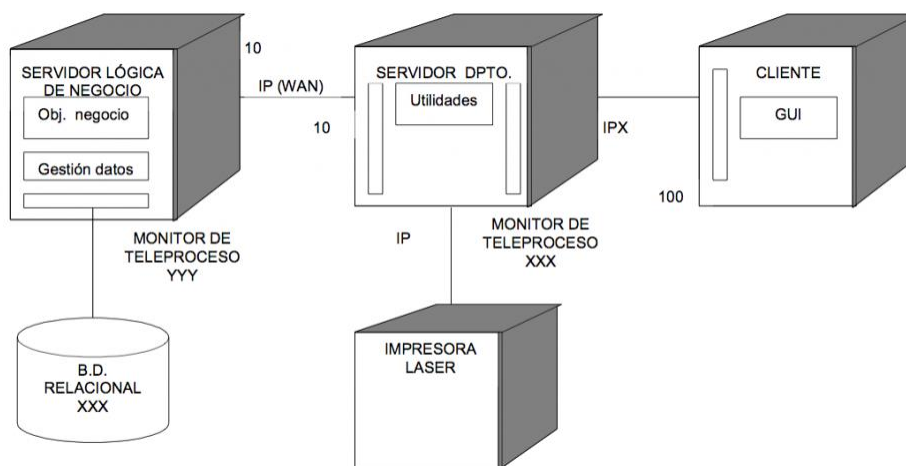


Figura 7. Ejemplo de diagrama de despliegue.

3.1.7 Diagrama de estructuras compuestas

Un diagrama de estructura compuesta muestra la estructura interna de una clase y las colaboraciones que esta estructura hace posibles. Una estructura compuesta es un conjunto de elementos interconectados que colaboran en tiempo de ejecución para lograr algún propósito. Cada elemento tiene algún rol definido en la colaboración (Clara, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de estructuras compuestas (Figura 8).

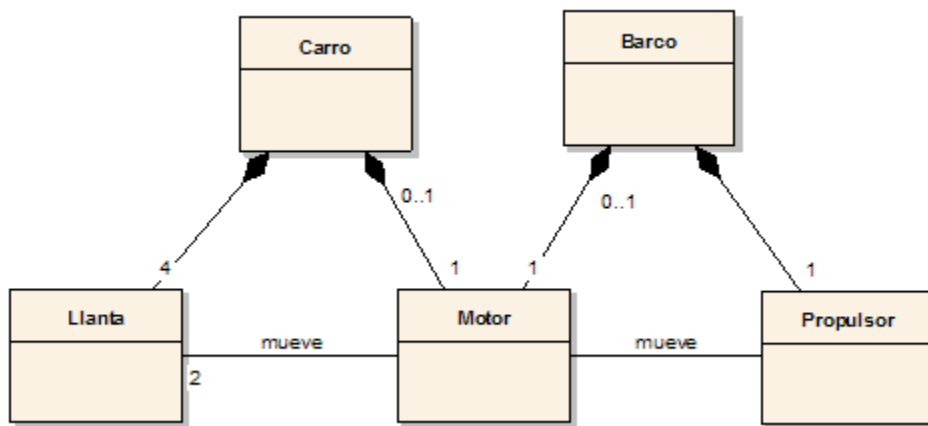


Figura 8. Ejemplo de diagrama de estructuras compuestas.

3.1.8 Diagrama de secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML (Clara, 2015).

El diagrama de secuencia enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos (Castillo, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de secuencia (Figura 9).

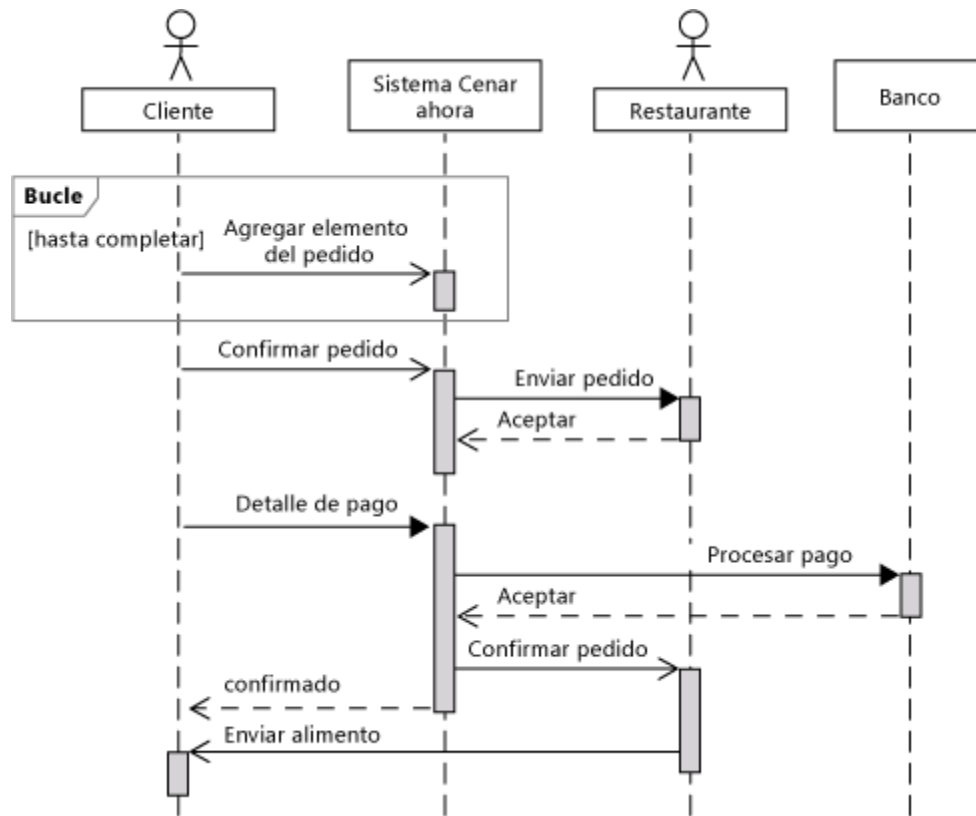


Figura 9. Ejemplo de diagrama de secuencia.

3.1.9 Diagrama de comunicación (o de colaboración)

Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema (Clara, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de colaboración (Figura 10).

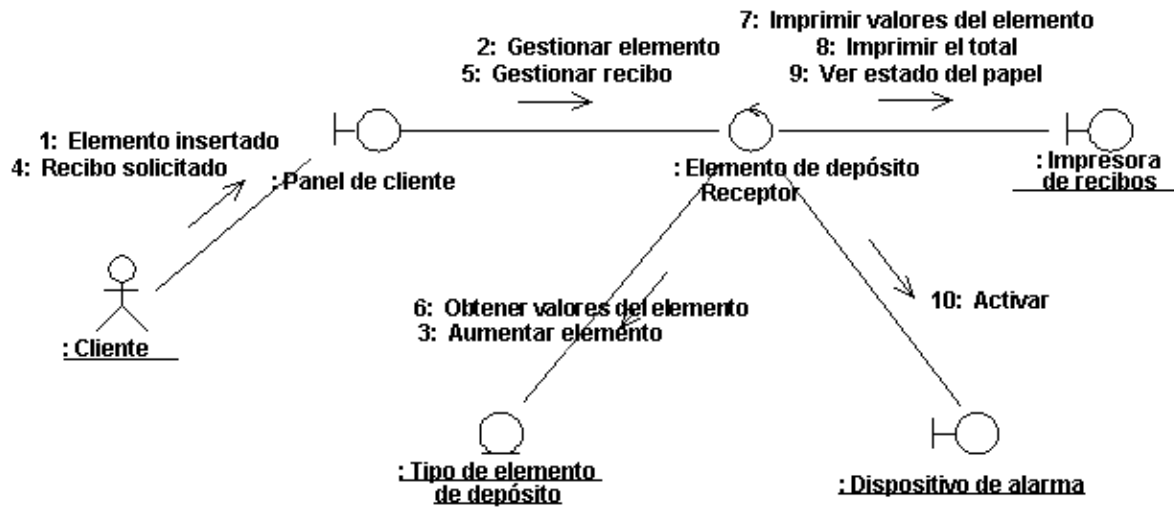


Figura 10. Ejemplo de diagrama de colaboración.

3.1.10 Diagrama de máquina de estados o de estados

Un diagrama de estados es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos. En la siguiente figura se muestra un ejemplo del diagrama de máquina de estados o de estados (Figura 11) (Clara, 2015).

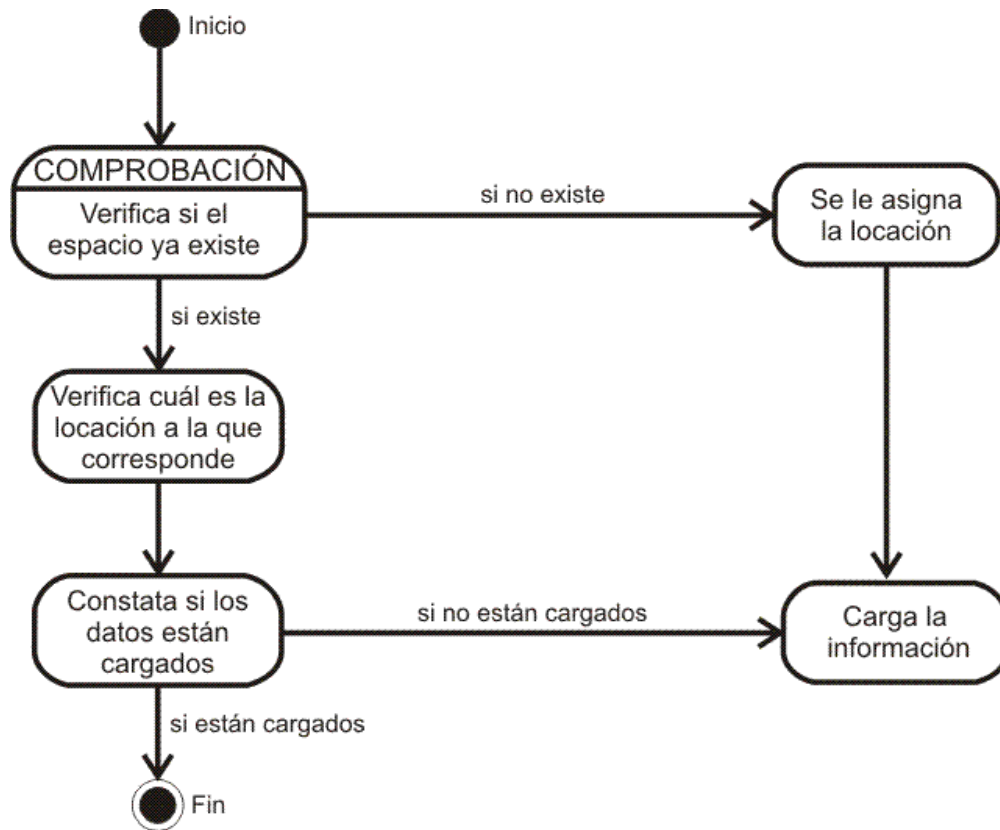


Figura 11. Ejemplo de diagrama de máquina de estados o de estados.

3.1.11 Diagrama de actividades

Un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un diagrama de actividades muestra el flujo de control general (Clara, 2015).

Es un caso especial del diagrama de estados, simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos (Castillo, 2015).

En la Figura 12 se muestra un ejemplo del diagrama de actividades.

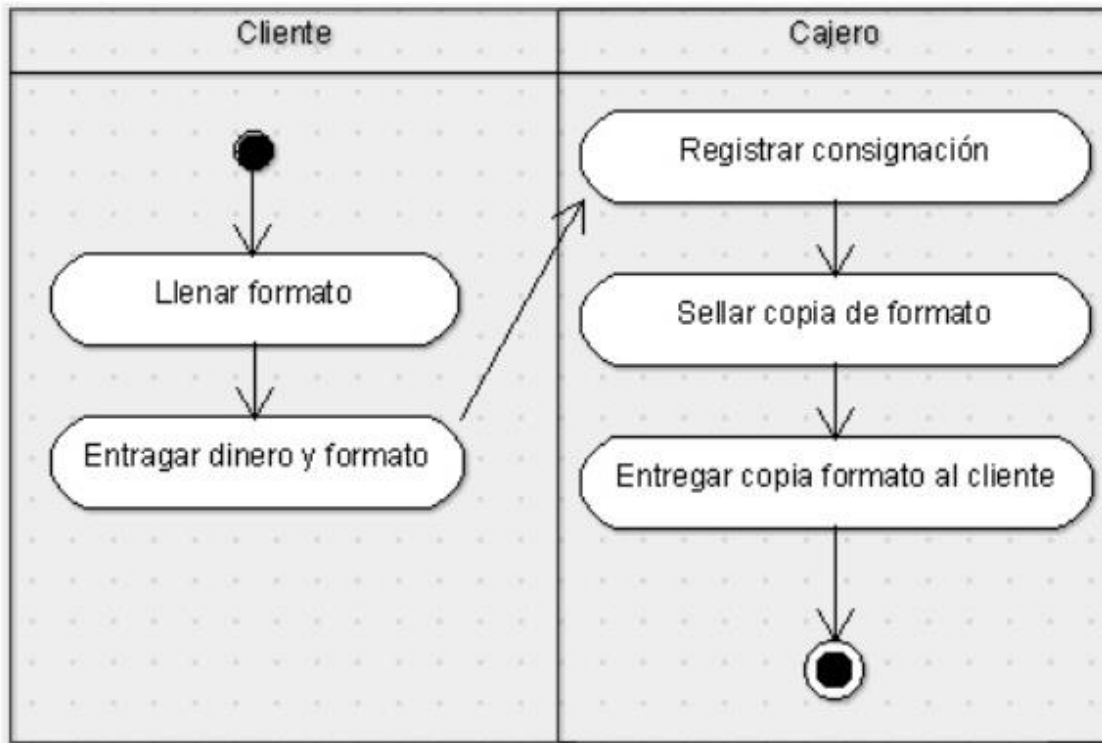


Figura 12. Ejemplo de diagrama de actividades.

3.1.12 Diagrama de visión global de la interacción

Los diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Los diagramas de interacción muestran cierto número de ejemplos de objetos y los mensajes que se pasan entre estos objetos dentro del caso de uso. Hay dos tipos de diagramas de interacción: Diagramas de secuencia y diagramas de colaboración (Clara, 2015).

El diagrama de secuencia y el diagrama de colaboración muestran los diferentes objetos y relaciones que pueden tener entre ellos, así como los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción (Castillo, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de interacción (Figura 13).

Diagramas de Interacción - UML

Diagramas de Colaboración

Diagrama de Secuencia

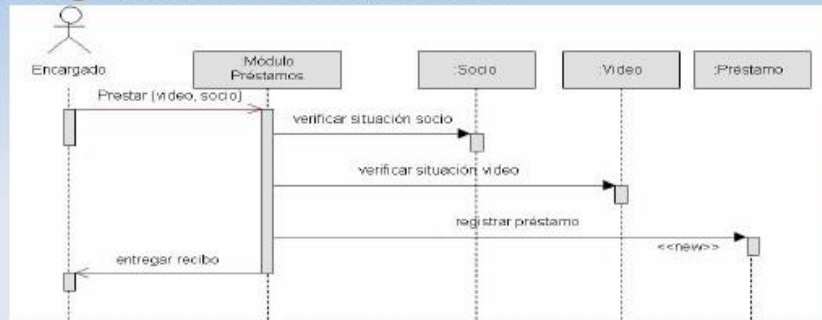


Diagrama de Colaboración

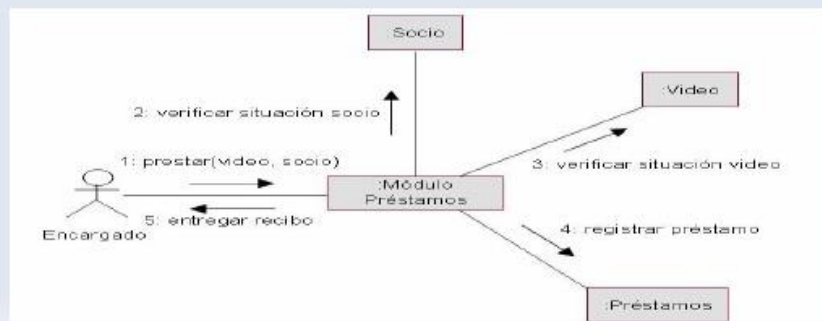


Figura 13. Ejemplo de diagrama de interacción.

3.1.13 Diagrama de tiempos

Un diagrama de tiempos o cronograma es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás (Clara, 2015).

En la siguiente figura se muestra un ejemplo del diagrama de tiempos (Figura 14).

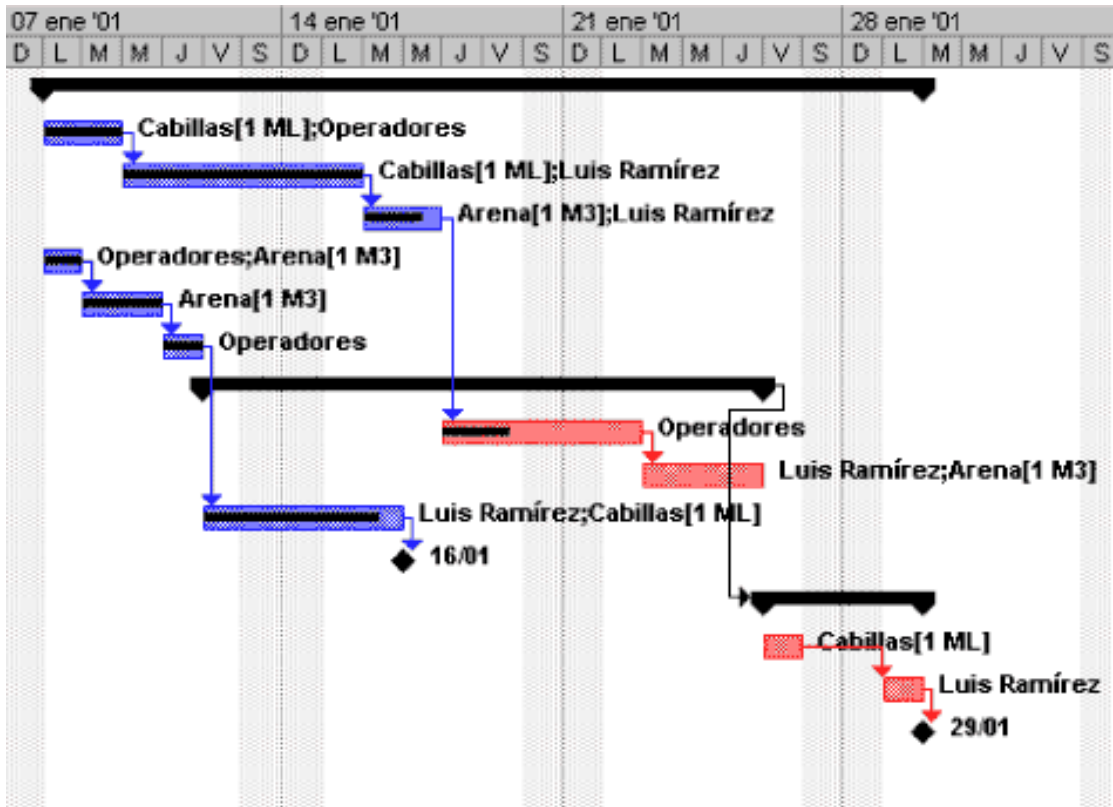


Figura 14. Ejemplo de diagrama de tiempos.

3.2 BASES DE DATOS

3.2.1 Principios de una base de datos

Una base de datos es una colección de información, organizada en función de una estructura que permite recuperar y actualizar sus datos. Esa colección de información representa una o varias funciones del proceso de negocios al que asiste y entrega información que concierne sólo a este proceso para la toma de decisiones, el registro de sus operaciones y el análisis de sus procesos.

Esta definición puede parecer algo abstracta, de su interpretación se concluye que una base de datos contiene información de gran valor para una empresa o negocio, también permite registrar las operaciones de interés, almacenando únicamente la información relevante y ayuda a la toma de decisiones (Rosa, 2005).

Las principales ventajas que ofrecen las bases de datos relacionales son; evitar la redundancia de información, mantener la consistencia de los datos almacenados, conservar y asegurar la integridad de los datos almacenados, permitir el acceso concurrente a los datos, garantizando la integridad ante la concurrencia de actualizaciones, y por último, proporcionar la oportunidad de mantener las estructuras para desarrollar aplicaciones con mayor facilidad (Rosa, 2005).

Normalización

El proceso de normalización consiste en una serie de pasos por aplicar sobre el diseño de una tabla, de manera que se pueda evitar la redundancia de información, las inconsistencias por eliminación de registros, la pérdida de dependencias funcionales por descomposición, la creación de relaciones o dependencias funcionales inexistentes, y la ineficiencia en el mantenimiento y el acceso a relaciones (Rosa, 2005).

Modelo Entidad/Relación

Las entidades son objetos, reales o abstractos, que existen o pueden llegar a existir en un contexto determinado y de las cuales se desea guardar información. Pueden clasificarse de la siguiente manera:

- Entidades regulares: Son aquellas que existen por sí solas y no dependen de la existencia de instancias de otras entidades.
- Entidades débiles: Son aquellas en las que es necesario la existencia de instancias de otras entidades distintas para que puedan existir ejemplares en esta entidad (Rosa, 2005).

Atributos

Las entidades están compuestas por atributos o propiedades que aseguran que cada instancia de la tabla sea única. Los atributos pueden ser obligatorios; estos deben tener un valor asignado, por lo tanto no les está permitido al registrar que su valor sea nulo. También hay atributos opcionales; son aquellos que pueden o no tener un valor asignado, su valor puede ser nulo y no habrá ningún problema al registrarlo (Rosa, 2005).

Formas Normales

Primer forma normal

Una entidad se encuentra en primera forma normal (1FN) si y sólo si cada uno de los campos contiene un único valor para un registro determinado y la tabla tiene definida una clave primaria (Rosa, 2005).

Segunda forma normal

Una tabla se encuentra en segunda forma normal (2FN) si comparando todos y cada uno de los campos de la tabla con la clave definida todos dependen directamente de ésta. Por otra parte, para aplicar la 2FN, la entidad ya debe estar en 1FN (Rosa, 2005).

Tercera forma normal

Se dice que una tabla está en tercera forma normal (3FN) si y sólo si los campos de la tabla dependen únicamente de la clave, es decir que los campos de las tablas no dependen unos de los otros (Rosa, 2005).

Cuarta forma normal

Una entidad está en cuarta forma normal (4FN) si y sólo si para cualquier combinación clave-campo no existen valores duplicados (Rosa, 2005).

Quinta forma normal

Una tabla se encuentra en la 5FN si cumple con la 4FN, si y sólo si, cada relación de dependencia se encuentra definida por claves candidatas (Cruz, 2014).

Cuando una entidad no ha sido normalizada, decimos que está denormalizada (Cruz, 2014).

3.3 SQL SERVER

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact -SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas (Rouse, SQL Server, 2016).

SQL Server ofrece la posibilidad de realizar desarrollos a través de una herramienta gráfica, o interfaz de desarrollo, denominada SQL Server Management Studio. En esta interfaz gráfica, podemos realizar nuestro trabajo agrupando los archivos de scripts dentro de soluciones.

T-SQL es un lenguaje relacional, orientando a datos, que se ocupa más por definir el conjunto de resultados que se pretende obtener que por los aspectos físicos de su recolección (Rosa, 2005).

SQL Server 2005 fue lanzado en noviembre de 2005. Se dice que el producto 2005 proporcionó una mayor flexibilidad, escalabilidad, confiabilidad y seguridad a las aplicaciones de base de datos, y permitió que fueran más fáciles de crear y desplegar, lo que reduce la complejidad y el tedio involucrado en la gestión de bases de datos. SQL Server 2005 también incluía más soporte administrativo. El código original de SQL Server ha sido desarrollado por Sybase; a finales de 1980, Microsoft, Sybase y Ashton-Tate colaboraron para producir la primera versión del

producto, SQL Server 4.2 para OS/2. Posteriormente, tanto Sybase como Microsoft ofrecieron productos de SQL Server. Sybase cambió después el nombre de su producto a Adaptive Server Enterprise (Rouse, SQL Server, 2016).

Sentencias DDL

Las sentencias DDL (Data Definition Language) se utilizan para manipular objetos en el ámbito de una base de datos y se basan, generalmente, en sentencias GRANT, CREATE, ALTER, DROP y DENY.

Sentencia GRANT

La sentencia GRANT de SQL se utiliza para otorgar privilegios SQL SELECT, UPDATE, INSERT, DELETE y otros en tablas o vistas.

La palabra clave PUBLIC se utiliza para otorgar privilegios SQL a todos los usuarios locales (IBM, 2016).

Sentencia CREATE

En la siguiente tabla se muestran las sentencias DDL CREATE cuyo ámbito de ejecución es la base de datos. Para todas las sentencias CREATE existen las correspondientes ALTER y DROP, así como también se cuenta con DENY como contrapartida de las sentencias GRANT (Rosa, 2005).

Tabla 1. Sentencias CREATE.

SENTENCIA	FUNCIÓN
CREATE_TABLE	Crea una tabla.
CREATE_VIEW	Crea una vista.
CREATE_SYNONYM	Crea sinónimos.
CREATE_USER	Crea usuarios.
CREATE_ROLE	Crea roles.
CREATE_APPLICATION_ROLE	Crea roles de aplicación.
CREATE_SCHEMA	Crea esquemas.
CREATE_MESSAGE_TYPE	Crea un tipo de mensaje.

CREATE_CONTRACT	Crea un contrato que define los tipos de mensajes que se van a usar en una conversación.
CREATE_QUEUE	Crea una cola de almacenamiento de mensajes.
CREATE_SERVICE	Crea servicios para ejecutar un conjunto de tareas.
CREATE_ROUTE	Crea una nueva ruta en la tabla de rutas de la base de datos.
CREATE_REMOTE_SERVICE_BINDING	Crea vínculos que relacionan un servicio remoto con credenciales de usuario.
CREATE_DATABASE	Crea permisos de acceso.
CREATE_XML_SCHEMA_COLLECTION	Incorpora los componentes del esquema dentro de la base de datos.
CREATE_PARTITION_FUNCTION	Crea una función en la base de datos que vincula las filas de una tabla o índices dentro de particiones basadas en los valores de una columna específica.
CREATE_PARTITION_SCHEME	Crea un esquema dentro de la base de datos que relaciona las particiones de una tabla particionada o de un índice a filegroups.

Fuente: (Rosa, 2005)

Sentencia ALTER

En SQL, ALTER es un comando de la categoría DDL (Data Definition Language) y como su nombre lo indica sirve para alterar objetos ya creados en una base de datos, su uso principal es la modificación de tablas (LINUXTOTAL, 2016).

Tabla 2. Sentencias ALTER.

REFERENCIA DE USO DE 'ALTER TABLE'
SOBRE LA TABLA
ALTER TABLE ejemplo ENGINE = InnoDB Cambiar el tipo de motor (engine) de la tabla 'ejemplo'
ALTER TABLE personas RENAME usuarios Cambia el nombre de la tabla 'personas' a 'usuarios'
ALTER TABLE ejemplo AUTO_INCREMENT=1000 En la tabla 'ejemplo' cualquiera que sea la columna que tenga 'AUTO_INCREMENT' en sus propiedades (solo puede haber una), los nuevos registros comenzarán a partir de '1000' o cualquier número indicado, no es posible utilizar un valor ya existente.
ALTER TABLE ejemplo CONVERT TO CHARACTER SET latin1 La tabla 'ejemplo' ahora almacenará sus valores en base al juego de caracteres 'latin1' (iso-8859-1).
OPERACIONES CON DROP
ALTER TABLE ejemplo DROP COLUMN nombre Elimina la columna 'nombre' de la tabla 'ejemplo'.
ALTER TABLE ejemplo DROP COLUMN nombre, DROP COLUMN paterno Elimina más de una columna.
ALTER TABLE ejemplo DROP COLUMN nombre, DROP COLUMN paterno Elimina más de una columna.
ALTER TABLE ejemplo DROP INDEX usuario Elimina el índice 'usuario'.
ALTER TABLE ejemplo DROP PRIMARY KEY Elimina la llave primaria de la tabla 'ejemplo'
ALTER TABLE ejemplo DROP FOREIGN KEY id_usuario Elimina de la tabla 'ejemplo' la llave foranea 'id_usuario'.
OPERACIONES CON CHANGE Y MODIFY
ALTER TABLE ejemplo CHANGE monto cantidad FLOAT(8,2) Cambia el nombre de la columna 'monto' al nuevo nombre 'cantidad' con la definición del tipo de datos.
ALTER TABLE ejemplo CHANGE cantidad cantidad FLOAT(10,2) Cambia solo el tipo de datos de la columna, conservando el mismo nombre.

ALTER TABLE ejemplo MODIFY cantidad FLOAT(10,2) Cambia solo el tipo de datos de la columna, conservando el mismo nombre. (Igual que el anterior)
ALTER TABLE ejemplo MODIFY cantidad FLOAT(6,2) NOT NULL Cambia el tipo de datos de la columna 'cantidad' y especifica que no admite nulos.
ALTER TABLE ejemplo MODIFY paterno VARCHAR(30) Modifica el tamaño de la columna 'paterno'.
ALTER TABLE ejemplo MODIFY correo CONVERT TO CHARACTER SET utf8 Es posible convertir solo una columna, en este caso 'correo' a un juego de caracteres distinto al de todo el resto de la tabla.
OPERACIONES CON ADD
ALTER TABLE ejemplo ADD fecha DATE Añade una columna llamada 'fecha' del tipo 'DATE' al final de todas las demás existentes.
ALTER TABLE ejemplo ADD INDEX(categoria) Añade un índice a la columna 'categoria'.
ALTER TABLE ejemplo ADD INDEX(categoria), ADD PRIMARY KEY(clave) Añade un índice a la columna 'categoria' y además crea la llave primaria en la columna 'clave'.
ALTER TABLE ejemplo ADD UNIQUE(email) Añade a la columna 'email' un índice del tipo único, no puede haber dos iguales.
ALTER TABLE ejemplo ADD consecutivo BIGINT AUTO_INCREMENT, ADD INDEX(consecutivo) Añade la columna 'consecutivo' con la característica de auto incremento y además genera un índice sobre la misma.
ALTER TABLE ejemplo ADD materno VARCHAR(20) AFTER paterno Añade la columna 'materno' después de la columna 'paterno'.
ALTER TABLE ejemplo ADD id INT FIRST Añade la columna 'id' en primer lugar con respecto a las existentes.
ALTER TABLE usuarios ADD FOREIGN KEY(id) REFERENCES entradas(id_user) Añade un 'Foreign key' en la columna 'id' de la tabla 'usuarios' que apunta a la columna 'id_user' de la tabla 'entradas'.

Fuente: (LINUXTOTAL, 2016)

Sentencia DROP

La sentencia DROP sirve para borrar definitivamente una base de datos, una tabla o un índice.

Sintaxis con sentencia DROP

Borrar una base de datos: DROP DATABASE nombre_base_de_datos.

Borrar una tabla: DROP TABLE nombre_de_la_tabla.

Borrar un índice: DROP INDEX nombre_de_indice (Java, 2014).

Sentencia DENY

La sentencia DENY, como su nombre lo dice; deniega un permiso a una entidad de seguridad. Evita que la entidad de seguridad herede permisos por su pertenencia a grupos o funciones (Compitiendo, 2016).

Sintaxis:

```
DENY { ALL [ PRIVILEGES ] }  
    | permission [ ( column [ ,...n ] ) ] [ ,...n ]  
    [ ON [ class :: ] securable ] TO principal [ ,...n ]  
    [ CASCADE ] [ AS principal ]
```

ALL: Esta opción no deniega todos los permisos posibles. Al denegar ALL se deniegan los permisos siguientes.

- Si el elemento protegible es una base de datos, ALL significa BACKUP DATABASE, BACKUP LOG, CREATE DATABASE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE RULE, CREATE TABLE y CREATE VIEW.
- Si el elemento protegible es una función escalar, ALL significa EXECUTE y REFERENCES.
- Si el elemento protegible es una función con valores de tabla, ALL significa DELETE, INSERT, REFERENCES, SELECT y UPDATE.
- Si el elemento protegible es un procedimiento almacenado, ALL significa EXECUTE.
- Si el elemento protegible es una tabla, ALL significa DELETE, INSERT, REFERENCES, SELECT y UPDATE.
- Si el elemento protegible es una vista, ALL significa DELETE, INSERT, REFERENCES, SELECT y UPDATE.

PRIVILEGES:

Incluido por compatibilidad con ISO. No cambia el comportamiento de ALL.

Permission

Es el nombre de un permiso.

Column:

Especifica el nombre de una columna de una tabla para la que se deniegan los permisos. Los paréntesis () son obligatorios.

Class:

Especifica la clase de elemento protegible para el que se deniega el permiso. Se requiere el calificador de ámbito.

Securable:

Especifica el elemento protegible para el que se deniega el permiso.

TO principal:

Es el nombre de una entidad de seguridad. Las entidades de seguridad para las que se pueden denegar permisos sobre un elemento protegible varían en función de este elemento protegible.

CASCADE:

Indica que el permiso se deniega para la entidad de seguridad especificada y para el resto de entidades de seguridad a las que ésta concedió el permiso. Es obligatorio cuando la entidad de seguridad tiene el permiso con GRANT OPTION.

AS principal:

Especifica una entidad de seguridad a partir de la cual la entidad de seguridad que ejecuta esta instrucción obtiene su derecho a denegar el permiso (TechNet, DENY (Transact-SQL), 2016).

Sentencia TRUNCATE

La instrucción TRUNCATE TABLE es un método rápido y eficiente para eliminar todas las filas de una tabla. TRUNCATE TABLE es equivalente a la instrucción DELETE sin una cláusula WHERE. Sin embargo, TRUNCATE TABLE es más rápida y utiliza menos recursos de registro de sistema y de transacciones.

En comparación con la instrucción DELETE, TRUNCATE TABLE ofrece las siguientes ventajas:

- Utiliza menos espacio de registro de transacciones.

La instrucción DELETE quita una a una las filas y graba una entrada en el registro de transacciones por cada fila eliminada. TRUNCATE TABLE quita los datos al cancelar la asignación de las páginas de datos utilizadas para almacenar los datos de la tabla y sólo registra la página de asignaciones anuladas en el registro de transacciones.

- Suele utilizar menos bloqueos.

Cuando la instrucción DELETE se ejecuta mediante un bloqueo de fila, cada fila de la tabla se bloquea para su eliminación. TRUNCATE TABLE siempre bloquea la tabla y la página pero no cada fila.

- Sin excepción, las páginas vacías permanecen en la tabla.

Después de ejecutar una instrucción DELETE, la tabla todavía puede contener páginas vacías. Por ejemplo, las páginas vacías de un montón no se pueden desasignar sin por lo menos un bloqueo de tabla exclusivo. Si la operación de eliminación no utiliza un bloqueo de tabla, la tabla (montón) contendrá muchas páginas vacías. Para los índices, la operación de eliminación puede dejar páginas vacías, aunque éstas se desasignarán rápidamente por medio de un proceso de limpieza en segundo plano (TechNet, TRUNCATE, 2014).

Sentencias DML

Las sentencias DML (Data Manipulation Language) conforman el conjunto de instrucciones necesario para manipular datos en T-SQL (Rosa, 2005).

El Lenguaje de Manipulación de Datos (DML, por sus siglas en inglés) se utiliza para obtener, insertar y modificar información de una base de datos (Garay, 2015).

Sentencia INSERT

La instrucción INSERT de SQL permite añadir registros a una tabla. Con ella es posible ir añadiendo registros uno a uno, o añadir de golpe tantos registros como devuelva una instrucción SELECT.

Sintaxis: INSERT INTO NombreTabla [(Campo1, ..., CampoN)] VALUES (Valor1, ..., ValorN)

Siendo:

- **NombreTabla:** La tabla en la que se van a insertar las filas.
- **(Campo1, ..., CampoN):** Representa el campo o campos en los que se va a introducir valores.
- **(Valor1, ..., ValorN):** Representan los valores que se van a almacenar en cada campo.

En realidad la lista de campos es opcional especificarla (por eso está entre corchetes en la sintaxis general). Si no se indica campo alguno se considera que por defecto se introducirá información en todos los campos de la tabla, y por lo tanto se deben introducir valores para todos ellos y en el orden en el que han sido definidos. En la práctica se suelen especificar siempre por claridad y para evitar errores.

Por otro lado, los valores se deben corresponder con cada uno de los campos que aparecen en la lista de campos, tanto en el tipo de dato que contienen como en el orden en el que se van a asignar. Es decir, si se indican una serie de campos en un orden determinado, la lista de valores debe especificar los valores a almacenar en dichos campos, en el mismo orden exactamente. Si un campo no está en la lista, se almacenará dentro de éste el valor NULL.

Nota: Si un campo está definido como NOT NULL (es decir, que no admite nulos o valores vacíos), se debe especificar siempre en la lista de campos a insertar. De no ser así se producirá un error al ejecutar la instrucción correspondiente de INSERT (campusMVP, Fundamentos de SQL: Inserción de datos - INSERT, 2014).

Sentencia SELECT

La sentencia SELECT describe una consulta a la base de datos relacional permitiendo la recuperación de filas y columnas de una o varias tablas. La ejecución de este tipo de consultas no modifica ningún dato. Los resultados de las consultas

siempre serán tuplas, es decir, conjuntos de resultados ordenados visualmente como la intersección de filas por columnas (Rosa, 2005).

La instrucción **SELECT** muestra información sobre los datos almacenados en la base de datos. Dicha información puede pertenecer a una o varias tablas.

Para realizar consultas sobre las tablas de las bases de datos se dispone de la instrucción **SELECT**. Con ella se puede consultar una o varias tablas. Es sin duda el comando más versátil del lenguaje SQL (campusMVP, **SELECT**, 2014).

Existen muchas cláusulas asociadas a la sentencia **SELECT** (**GROUP BY**, **ORDER**, **HAVING**, **UNION**). También es una de las instrucciones en la que con más frecuencia los motores de bases de datos incorporan cláusulas adicionales al estándar (campusMVP, **SELECT**, 2014).

El resultado de una consulta **SELECT** devuelve una tabla lógica. Es decir, los resultados son una relación de datos, que tiene filas/registros, con una serie de campos/columnas. Igual que cualquier tabla de la base de datos. Sin embargo esta tabla está en memoria mientras se utiliza, y luego se descarta. Cada vez que se ejecuta la consulta se vuelve a calcular el resultado (campusMVP, **SELECT**, 2014).

La sintaxis básica de una consulta **SELECT** es la siguiente (los valores opcionales van entre corchetes):

SELECT [**ALL / DISTINCT**] [*] / [ListaColumnas_Expresiones] AS [Expresion]

FROM Nombre_Tabla_Vista

WHERE Condiciones

ORDER BY ListaColumnas [**ASC / DESC**]

SELECT: Permite seleccionar las columnas que se van a mostrar y en el orden en que lo van a hacer. Simplemente es la instrucción que la base de datos interpreta como que se va a solicitar información (campusMVP, **SELECT**, 2014).

ALL / DISTINCT:

ALL es el valor predeterminado, especifica que el conjunto de resultados puede incluir filas duplicadas. Por regla general nunca se utiliza.

DISTINCT especifica que el conjunto de resultados sólo puede incluir filas únicas. Es decir, si al realizar una consulta hay registros exactamente iguales que aparecen más de una vez, éstos se eliminan. Muy útil en muchas ocasiones (campusMVP, SELECT, 2014).

Nombres de campos: Se debe especificar una lista de nombres de campos de la tabla de interés y que por lo tanto se quiere devolver. Normalmente habrá más de uno, en cuyo caso se separa cada nombre de los demás mediante comas.

Se puede anteponer el nombre de la tabla al nombre de las columnas, utilizando el formato Tabla.Columna. Además de nombres de columnas, en esta lista se pueden poner constantes, expresiones aritméticas, y funciones, para obtener campos calculados de manera dinámica.

Si se requiere devolver todos los campos de la tabla se utiliza el comodín “*” (asterisco).

Los nombres indicados deben coincidir exactamente con los campos de la tabla, pero si se quiere que en la tabla lógica de resultados tenga un nombre diferente se puede utilizar AS (campusMVP, SELECT, 2014).

AS: Permite renombrar columnas si se utiliza en la cláusula SELECT, o renombrar tablas si se utiliza en la cláusula FROM. Es opcional. Con ello se puede crear diversos alias de columnas y tablas (campusMVP, SELECT, 2014).

FROM: Esta cláusula permite indicar las tablas o vistas de las cuales se va a obtener la información (campusMVP, SELECT, 2014).

WHERE: Especifica la condición de filtro de las filas devueltas. Se utiliza cuando no se desea que se devuelvan todas las filas de una tabla, sino sólo las que cumplen ciertas condiciones. Lo habitual es utilizar esta cláusula en la mayoría de las consultas (campusMVP, SELECT, 2014).

Condiciones: Son expresiones lógicas a comprobar para la condición de filtro, que tras su resolución devuelven para cada fila TRUE o FALSE, en función de que se cumplan o no. Se puede utilizar cualquier expresión lógica y en ella utilizar diversos operadores como:

- > (Mayor)
- >= (Mayor o igual)
- < (Menor)
- <= (Menor o igual)
- = (Igual)
- <> o != (Distinto)
- IS [NOT] NULL (para comprobar si el valor de una columna es o no es nula, es decir, si contiene o no contiene algún valor)

Nota: Se dice que una columna de una fila es NULL si está completamente vacía. Hay que tener en cuenta que si se ha introducido cualquier dato, incluso en un campo alfanumérico si se introduce una cadena en blanco o un cero en un campo numérico, deja de ser NULL (campusMVP, SELECT, 2014).

- **LIKE:** Para la comparación de un modelo. Se utilizan los caracteres comodines especiales: “%” y “_”. Con el primero se indica que en su lugar puede ir cualquier cadena de caracteres, y con el segundo que puede ir cualquier carácter individual (un solo carácter). Con la combinación de estos caracteres se pueden obtener múltiples patrones de búsqueda. Por ejemplo:
 - El nombre empieza con A: Nombre LIKE ‘A%’
 - El nombre acaba con A: Nombre LIKE ‘%A’
 - El nombre contiene la letra A: Nombre LIKE ‘%A%’
 - El nombre empieza con A y después contiene un solo carácter cualquiera: Nombre LIKE ‘A_’
 - El nombre empieza con una A, después cualquier carácter, luego una E y al final cualquier cadena de caracteres: Nombre LIKE ‘A_E%’
- **BETWEEN:** Para un intervalo de valores. Por ejemplo:
 - Clientes entre el 30 y el 100: CodCliente BETWEEN 30 AND 100
 - Clientes nacidos entre 1970 y 1979: FechaNac BETWEEN ‘19700101’ AND ‘19791231’
- **IN():** Para especificar una relación de valores concretos. Por ejemplo: Ventas de los Clientes 10, 15, 30 y 75: CodCliente IN(10, 15, 30, 75)

Por supuesto es posible combinar varias condiciones simples de los operadores anteriores utilizando los operadores lógicos **OR**, **AND** y **NOT**, así como el uso de paréntesis para controlar la prioridad de los operadores (como en matemáticas). Por ejemplo: ... (Cliente = 100 AND Provincia = 30) OR Ventas > 1000 ... que sería para los clientes de las provincias 100 y 30 o cualquier cliente cuyas ventas superen 1000 (campusMVP, SELECT, 2014).

ORDER BY: Define el orden de las filas del conjunto de resultados. Se especifica el campo o campos (separados por comas) por los cuales queremos ordenar los resultados (campusMVP, SELECT, 2014).

ASC / DESC:

ASC es el valor predeterminado, especifica que la columna indicada en la cláusula ORDER BY se ordenará de forma ascendente, o sea, de menor a mayor. Si por el contrario se especifica DESC se ordenará de forma descendente (de mayor a menor).

Por ejemplo, para ordenar los resultados de forma ascendente por ciudad, y los que sean de la misma ciudad de forma descendente por nombre, se utilizaría esta cláusula de ordenación: ... ORDER BY Ciudad, Nombre DESC ...

Como a la columna Ciudad no se le ha puesto ASC o DESC se usará para la misma el valor predeterminado (que es ASC) (campusMVP, SELECT, 2014).

Sentencia UPDATE

La sentencia UPDATE actualiza los valores de las columnas especificadas en filas de una tabla, vista o apodo, o en las tablas, apodos o vistas subyacentes de la selección completa especificada. La actualización de una fila de una vista actualiza una fila de su tabla base, si no se ha definido ningún activador INSTEAD OF para la operación de actualización en esta vista. Si se ha definido un activador de este tipo, en su lugar se ejecutará el activador. La actualización de una fila utilizando un apodo actualiza una fila del objeto de fuente de datos al que hace referencia el apodo.

Las formas de esta sentencia son:

- La forma UPDATE con búsqueda se utiliza para actualizar una o varias filas (determinadas opcionalmente por la condición de búsqueda).
- La forma de UPDATE con posición se utiliza para actualizar exactamente una fila (tal como determina la posición actual de un cursor).

Invocación.

Una sentencia UPDATE puede incorporarse en un programa de aplicación o emitirse mediante la utilización de sentencias de SQL dinámico. Es una sentencia ejecutable que puede prepararse de forma dinámica.

Autorización.

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Privilegio UPDATE para la tabla, vista o apodo de destino.
- Privilegio UPDATE para cada una de las columnas que deben actualizarse.
- Privilegio CONTROL para la tabla, vista o apodo de destino.
- Autorización SYSADM o DBADM. (Knowledge, UPDATE, 2016)

WHERE: Introduce una condición que indica qué filas se actualizan. Puede omitir la cláusula, proporcionar una condición de búsqueda o nombrar un cursor. Si se omite la cláusula, se actualizan todas las filas de la tabla, vista o apodo. (Knowledge, UPDATE, 2016)

Sentencia DELETE

La sentencia DELETE suprime filas de una tabla, un apodo o una vista o las tablas, los apodos o las vistas subyacentes de la selección completa especificada. La supresión de una fila de un apodo suprime la fila del objeto de fuente de datos al que hace referencia el apodo. La supresión de una fila de una vista suprime la fila de la tabla en la que se basa la vista si no se ha definido ningún activador INSTEAD OF para la operación de supresión en esta vista. Si se ha definido un activador de este tipo, en su lugar se ejecutará el activador (Knowledge, DELETE, 2016)

Existen dos formas de esta sentencia:

- La forma DELETE con búsqueda se utiliza para suprimir una o varias filas (determinadas opcionalmente mediante una condición de búsqueda).
- La forma DELETE con posición se utiliza para suprimir una fila exactamente (determinada por la posición actual del cursor) (Knowledge, DELETE, 2016).

Invocación.

Una sentencia DELETE puede incorporarse en un programa de aplicación o emitirse mediante la utilización de sentencias de SQL dinámico. Se trata de una sentencia ejecutable que puede prepararse de forma dinámica (Knowledge, DELETE, 2016).

Autorización.

Para ejecutar cualquiera de las dos formas de esta sentencia, el ID de autorización de la sentencia debe poseer como mínimo los siguientes privilegios:

- Privilegio DELETE para la tabla, vista o apodo de donde van a suprimirse filas.
- Privilegio CONTROL sobre la tabla, vista o apodo de donde van a suprimirse filas.
- Autorización SYSADM o DBADM (Knowledge, DELETE, 2016).

Para ejecutar una sentencia DELETE buscada, los privilegios del ID de autorización de la sentencia también deben incluir, como mínimo, uno de los siguientes para cada tabla, vista o apodo al que haga referencia una subconsulta:

- Privilegio SELECT
- Privilegio CONTROL
- Autorización SYSADM o DBADM

Ejemplo:

Suprimir el departamento (DEPTNO) 'D11' de la tabla DEPARTMENT

Sintaxis:

DELETE FROM DEPARTMENT

WHERE DEPTNO = 'D11'

FROM *nombre-tabla*, *nombre-vista*, *apodo*, o (*selección completa*): Identifica el objeto de la operación de supresión. El nombre debe identificar una tabla o vista que exista en el catálogo, pero no debe identificar una tabla de catálogo, una vista de catálogo, una tabla de consultas materializadas mantenida por el sistema o una vista de sólo lectura.

Si *nombre-tabla* es una tabla con tipo, la sentencia puede suprimir filas de la tabla o cualquiera de sus subtablas correspondientes.

Si *nombre-vista* es una vista con tipo, la sentencia puede que elimine las filas de la vista subyacente o de las vistas subyacentes de las subvistas correspondientes de la vista. Si *nombre-vista* es una vista normal con una tabla subyacente que es una tabla con tipo, puede que la sentencia suprima filas de la tabla con tipo o cualquiera de sus propias subtablas.

Si el objeto de la operación de supresión es una selección completa, esta debe ser suprimible, según lo definido en el elemento de Notas "Vistas suprimibles" de la descripción de la sentencia CREATE VIEW.

Sólo puede hacerse referencia a las columnas de la tabla especificada en la cláusula WHERE. Para una sentencia DELETE con posición, el cursor asociado también debe haber especificado la tabla o vista en la cláusula FROM sin utilizar ONLY (Knowledge, DELETE, 2016).

FROM ONLY (*nombre-tabla*): Aplicable a las tablas con tipo, la palabra clave ONLY especifica que la sentencia sólo se debe aplicar a los datos de la tabla especificada y no puede suprimir las filas de las subtablas correspondientes. Para una sentencia DELETE con posición, el cursor asociado también debe haber especificado la tabla en la cláusula FROM utilizando ONLY. Si *nombre-tabla* no es una tabla con tipo, la palabra clave ONLY no tiene efecto en la sentencia (Knowledge, DELETE, 2016).

FROM ONLY (*nombre-vista*): Aplicable a las vistas con tipo, la palabra clave ONLY especifica que la sentencia sólo se debe aplicar a los datos de la vista especificada y no puede suprimir las filas de las subvistas correspondientes. Para una sentencia

DELETE con posición, el cursor asociado también debe haber especificado la vista en la cláusula FROM utilizando ONLY. Si *nombre-vista* no es una vista con tipo, la palabra clave ONLY no tiene efecto en la sentencia (Knowledge, DELETE, 2016).

WHERE: Especifica una condición que selecciona las filas que se deben suprimir. Puede omitirse la cláusula, se puede especificar una condición de búsqueda o nombrar un cursor. Si se omite la cláusula, se suprimen todas las filas de la tabla o vista (Knowledge, DELETE, 2016).

SENTENCIAS DE CONTROL DE FLUJO

En la siguiente tabla se describen las principales sentencias de control de flujo dentro de procedimientos o procesos batch (Rosa, 2005).

Tabla 3. Sentencias de control de flujo.

CONSTRUCCIÓN	DESCRIPCIÓN
BEGIN...END	Se utiliza para entrecerrar un conjunto de sentencias unidas lógicamente (ramas de IF, WHILE, etc.).
GOTO	Continúa con el proceso a partir de la etiqueta definida.
IF..ELSE	Define una ejecución condicional.
WHILE 4	Construcción de bucle que se repite mientras se cumpla la condición.
RETURN n	Salida incondicional. Termina la ejecución del proceso y devuelve n al proceso que lo invoca.
WAITFOR	Indica el tiempo para la ejecución de una instrucción. Puede ser una hora o un retardo de tiempo en horas.
BREAK	Salida de un bucle WHILE interno.
CONTINUE	Reinicia un bucle WHILE.
CASE	Instrucción condicional de varias ramas, que permite un manejo de condiciones mejor que el IF y aplica a cada fila de un conjunto de resultados

Fuente: (Rosa, 2005)

CAPÍTULO 4. METODOLOGÍA.

4.1 ANÁLISIS

Se realizó la recopilación de información necesaria para obtener los requerimientos del sistema a través de la aplicación de encuestas, así como de necesidades previamente detectadas, lo cual sirvió para analizar los procesos que se requerían, identificar cada uno de los datos de entrada y detectar otros requerimientos funcionales y no funcionales del sistema.

4.1.1 Antecedentes

Se revisaron registros en Excel y en diferentes formatos sobre documentación del sistema que se disponía, con la finalidad de comprender cada uno de los procesos involucrados para el desarrollo de la investigación.

4.1.2 Entradas

- Registro de personal.
- Registro de proveedores.
- Registro de materiales.
- Registro de unidades.
- Registro de productos.
- Registro de clasificación.
- Registro de clientes.

4.1.3 Salidas

- Relación de pagos pendientes ordenados por fecha de compra.
- Relación de abonos pendientes ordenados por fecha de venta.
- Relación de abonos vencidos.
- Relación de abonos próximos a vencer.
- Relación de pagos vencidos.
- Relación de pagos próximos a vencer.
- Relación de clientes con adeudos.
- Relación de ventas de contado.
- Relación de ventas de crédito.

- Relación de productos terminados por categoría.
- Relación de producción en un lapso de fechas.
- Relación de pedidos y detalle de pedido por fecha de entrega.
- Relación de compras de contado en un lapso de fechas.
- Relación de compras de crédito en un lapso de fechas.
- Relación de proveedores con saldo > 0.
- Relación de personal.
- Relación de clientes.
- Relación de proveedores.
- Relación de materiales.
- Impresión de nota.
- Impresión de factura.

4.1.4 Políticas

- Aplicar un descuento aquellos clientes que cubran como anticipo el 80% del costo total del producto/pedido.
- Si la venta se hace a menudeo no se aplica ningún tipo de descuento.
- Si la venta se hace a mayoreo se le aplicará un descuento.
- Aplicar IVA al cliente por pago vencido sobre la compra a crédito.

4.1.5 Entidades externas

- Personal de la empresa
- Proveedores
- Clientes

4.1.6 Procesos

- Compras
- Ventas
- Gastos de producción
- Material sobrante de producción
- Productos terminados

- Pedidos
- Abonos
- Pagos
- Baja de personal
- Reactivación de personal

4.1.7 Almacenes de datos

- Personal
- BajaPersonal
- ReactivarBaja
- Proveedores
- Pagos
- Compras
- DetalleCompra
- Unidades
- Materiales
- Pedidos
- DetallePedido
- Produccion
- DetalleProduccion
- DetalleMateProdu
- DetalleSobrantes
- Clasificacion
- Productos
- Clientes
- Abonos
- Ventas
- DetalleVenta

4.1.8 Diagramas del sistema de información

En la figura No. 15 se muestra la primera parte del diagrama de Warnier/Orr.

Diagrama de Warnier/Orr. Parte 1

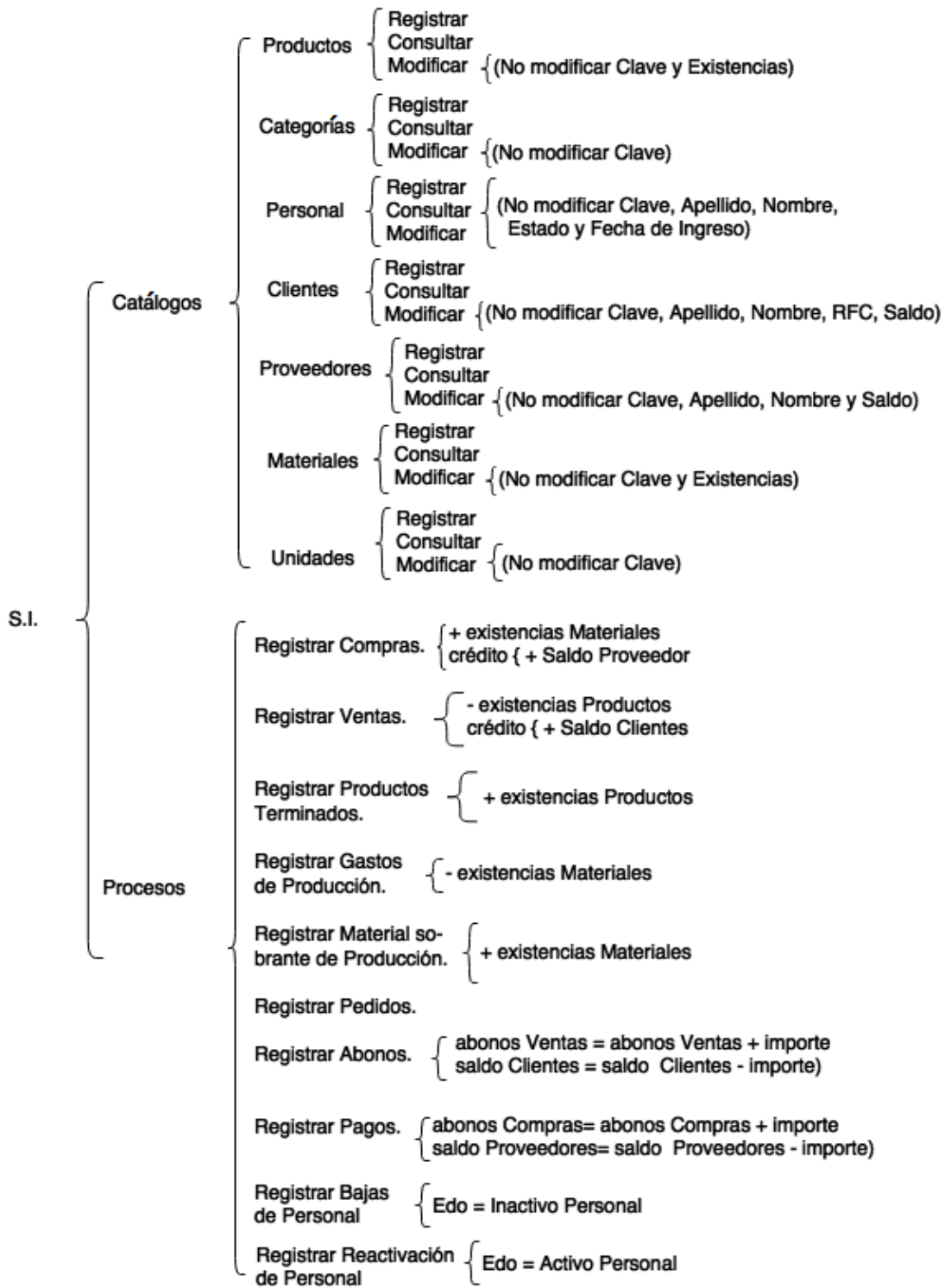


Figura 15. Diagrama de Warnier/Orr. Parte 1.

En la figura No. 16 se muestra la segunda parte del diagrama de Warnier/Orr.

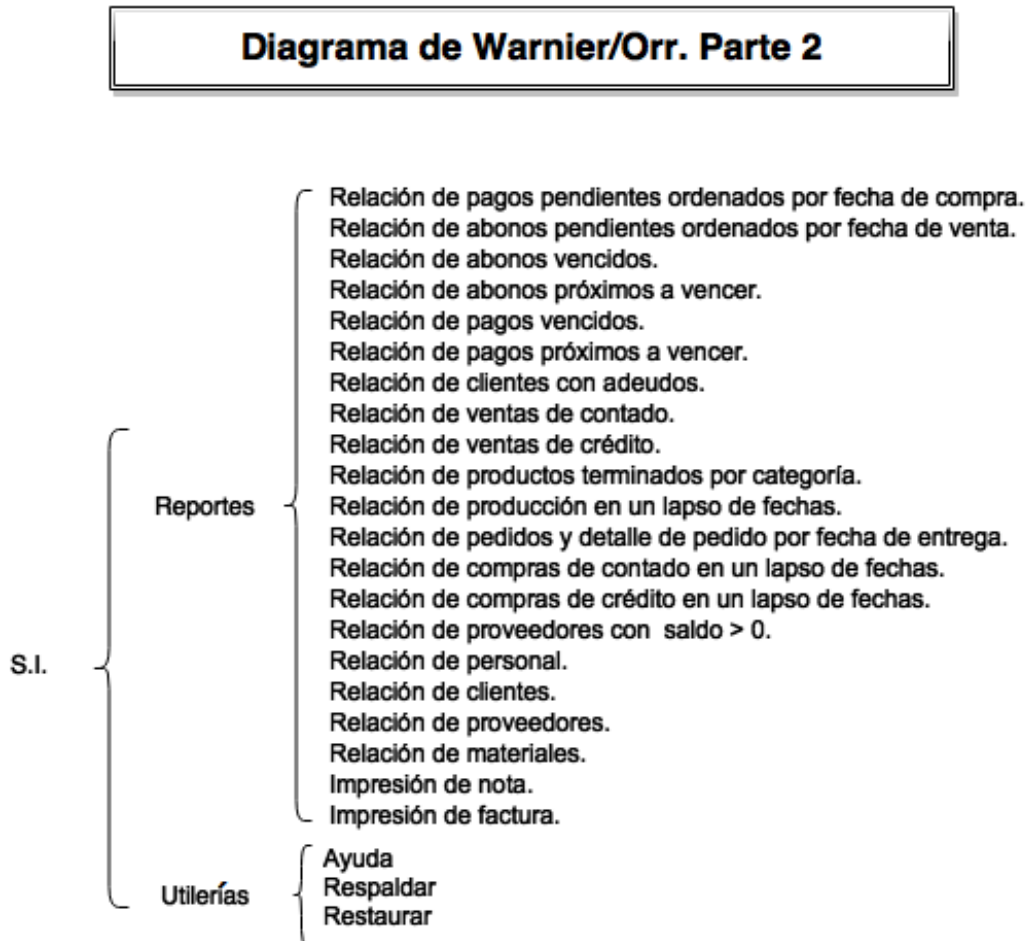


Figura 16. Diagrama Warnier/Orr. Parte 2.

En la figura No. 17 se muestra el diagrama de casos de uso general.

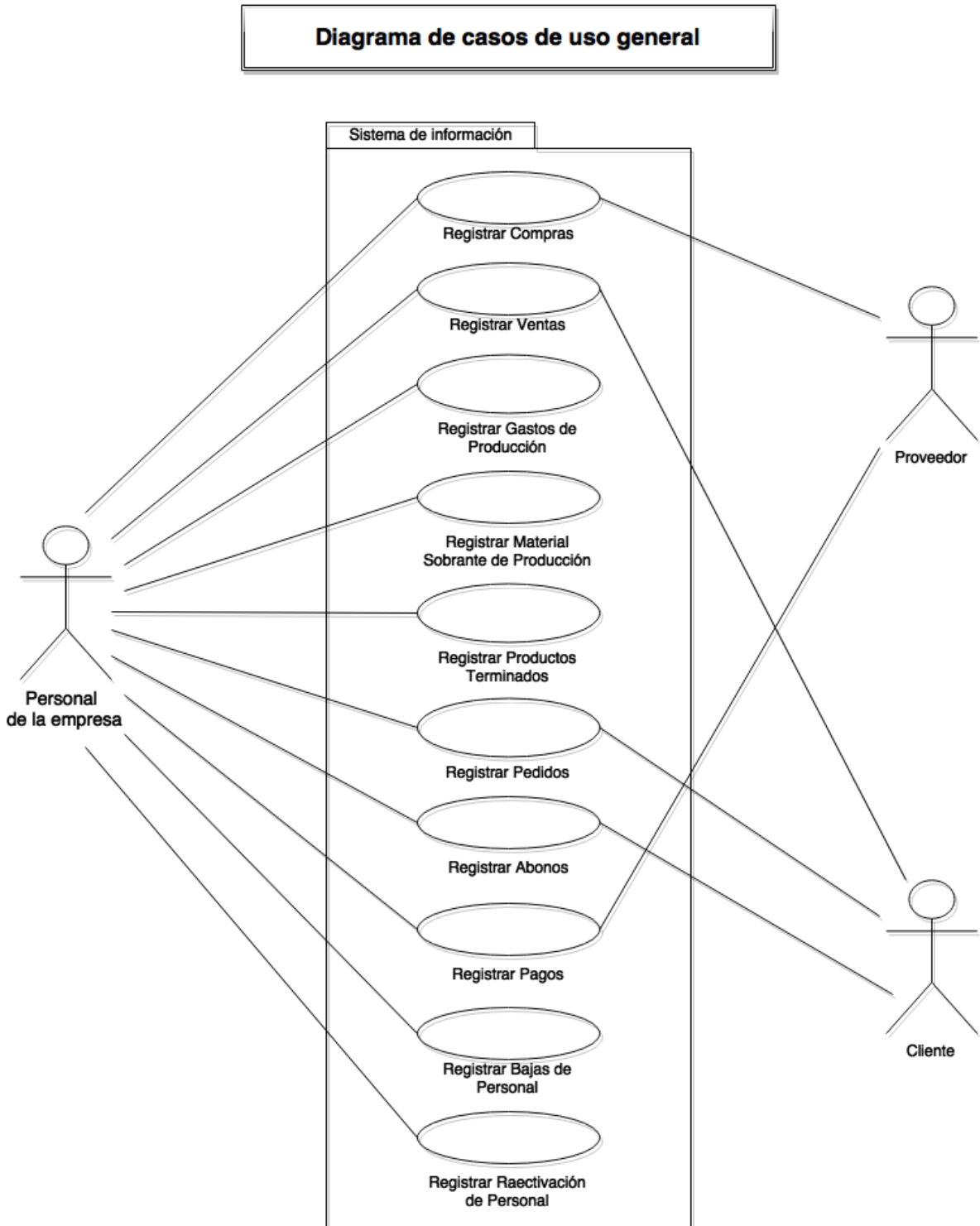


Figura 17. Diagrama de casos de uso general.

En la figura No. 18 se muestra el diagrama de contexto.

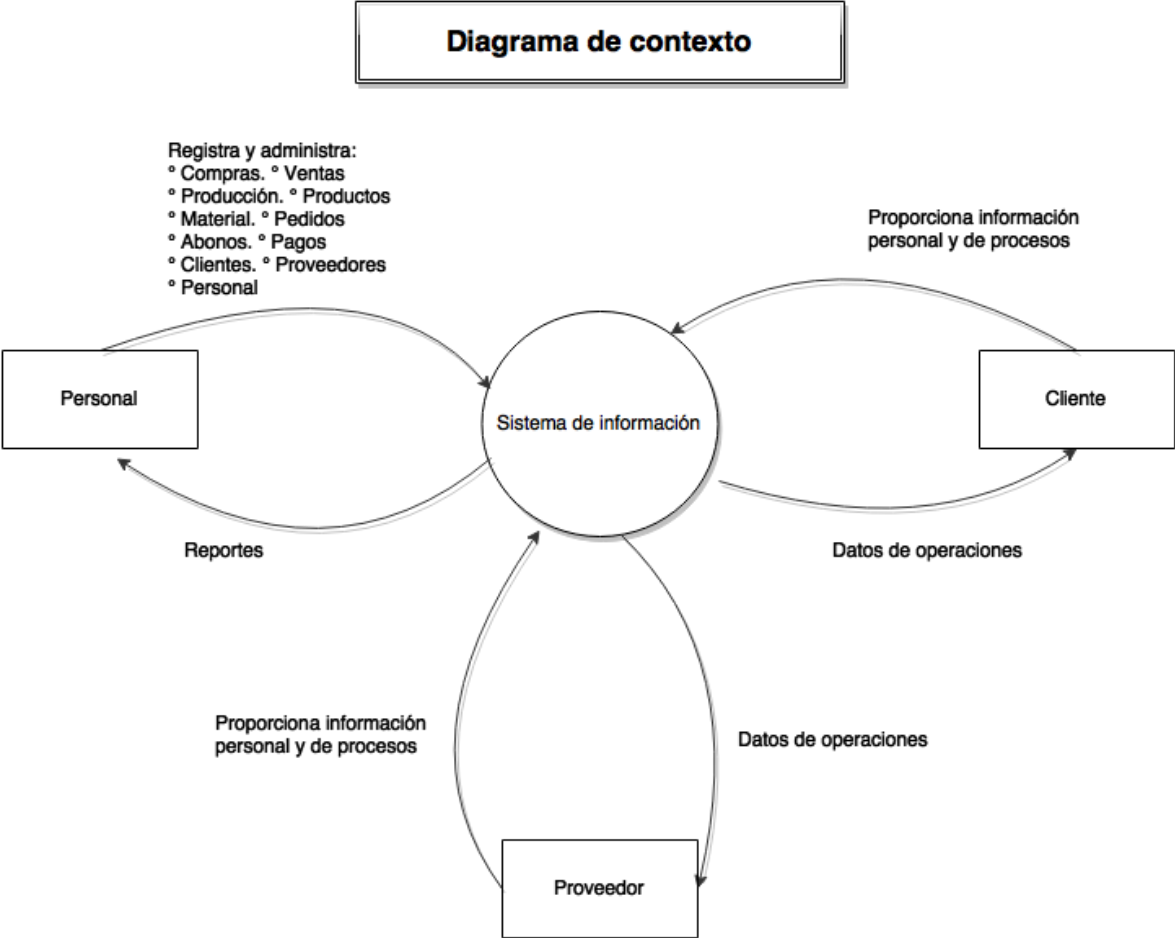


Figura 18. Diagrama de contexto.

4.1.9 Diagrama de flujo de datos de registrar compra

En la figura No. 19 se muestra el diagrama correspondiente:

Diagrama de flujo de datos: Registrar compra

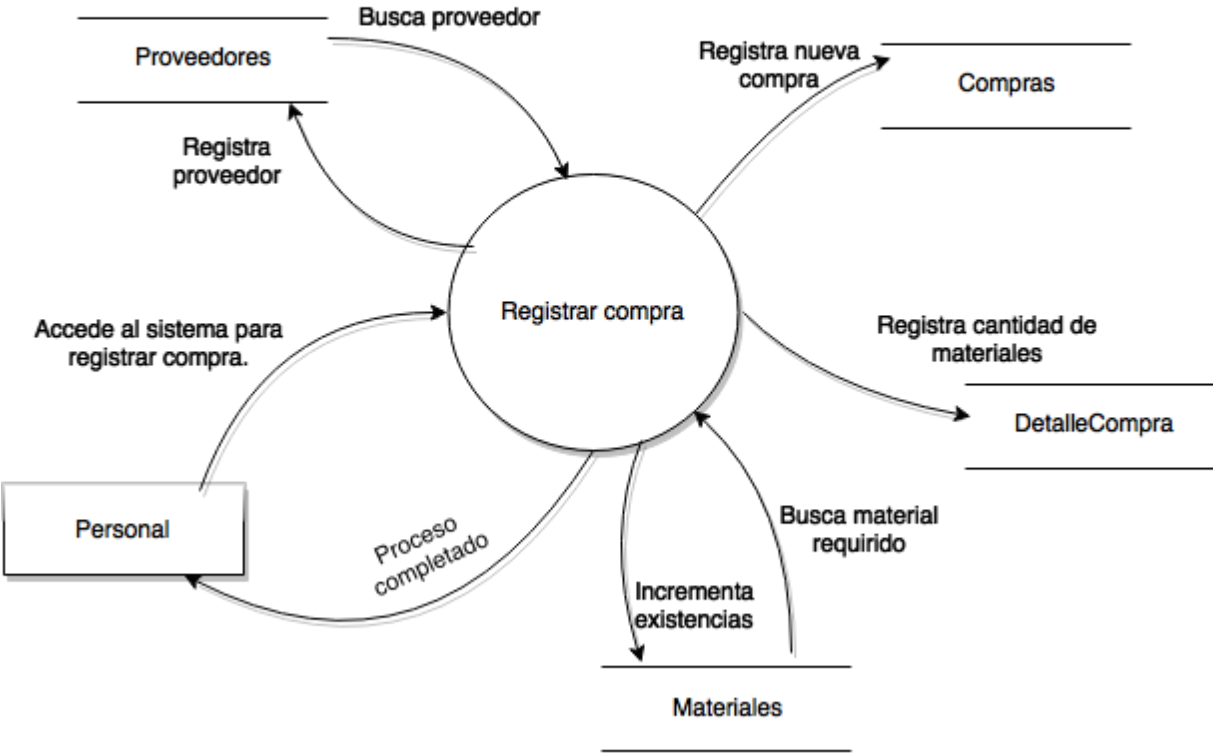


Figura 19. Diagrama de flujo de datos: Registrar compra.

4.1.10 Diagrama de estados de registrar compra

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del proveedor de la compra a registrar y visualiza sus datos. Posteriormente selecciona la descripción del material que va a registrar y visualiza los datos, enseguida proporciona cantidad y costo del material, después agrega los datos a la rejilla y de la misma manera realizará el registro para todos los materiales de la compra. Finalmente selecciona la condición de la compra si es a crédito o contado y en seguida graba la compra.

Precondición: La empresa recibe el material de una compra.

Caminos alternos: Registrar más artículos, compra a crédito, registrar en catálogo el material, registrar el catálogo al proveedor.

Postcondición: Termina cuando el personal graba la compra, actualizando automáticamente la existencia y costo en el inventario de Materiales y generando el registro en Compras con su respectivo detalle. (Ver figura No. 20).

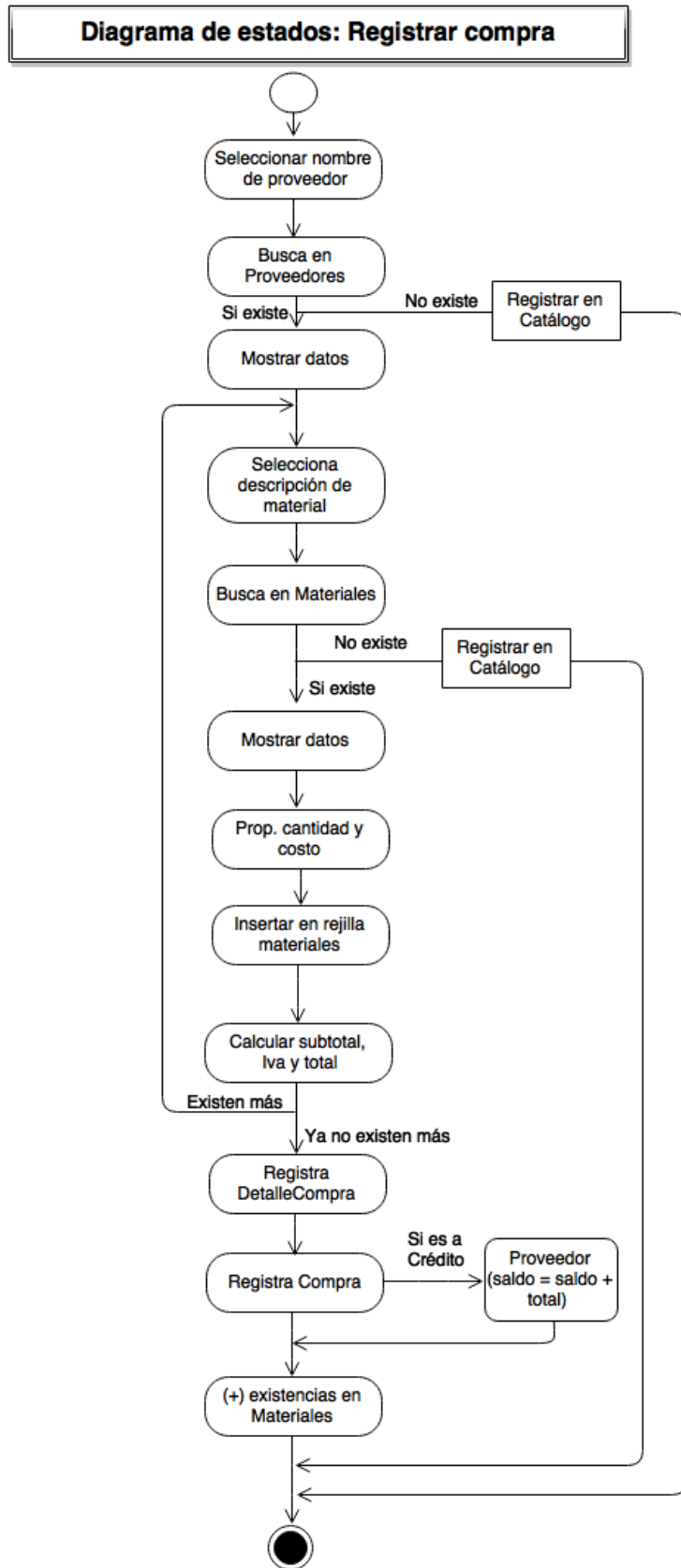


Figura 20. Diagrama de estados: Registrar compra.

4.1.11 Diagrama de colaboración de registrar compra

En la figura No. 21 se muestra el diagrama correspondiente.

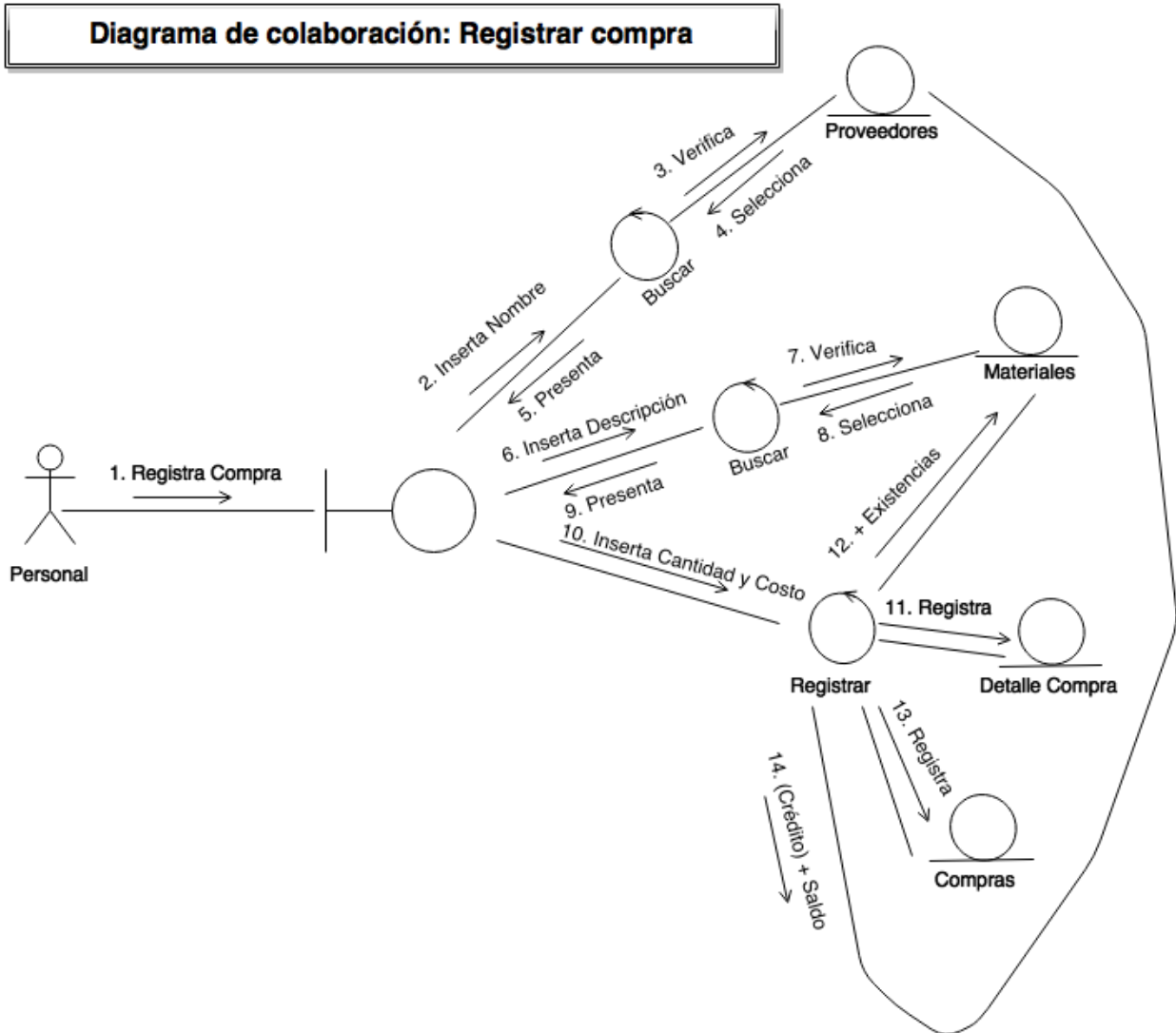


Figura 21. Diagrama de colaboración: Registrar compra.

4.1.12 Diagrama de caso de uso expandido de registrar compra

Nombre del caso: Registrar compra

Actores: Personal, Proveedor.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar una compra, busca al proveedor correspondiente y lo selecciona. Posteriormente elije la condición de la compra si es a crédito o contado. Finalmente actualiza automáticamente las existencias en Materiales y si la compra fue a crédito incrementa el saldo al proveedor.

En la figura No. 22 se muestra el diagrama de caso de uso expandido de registrar.

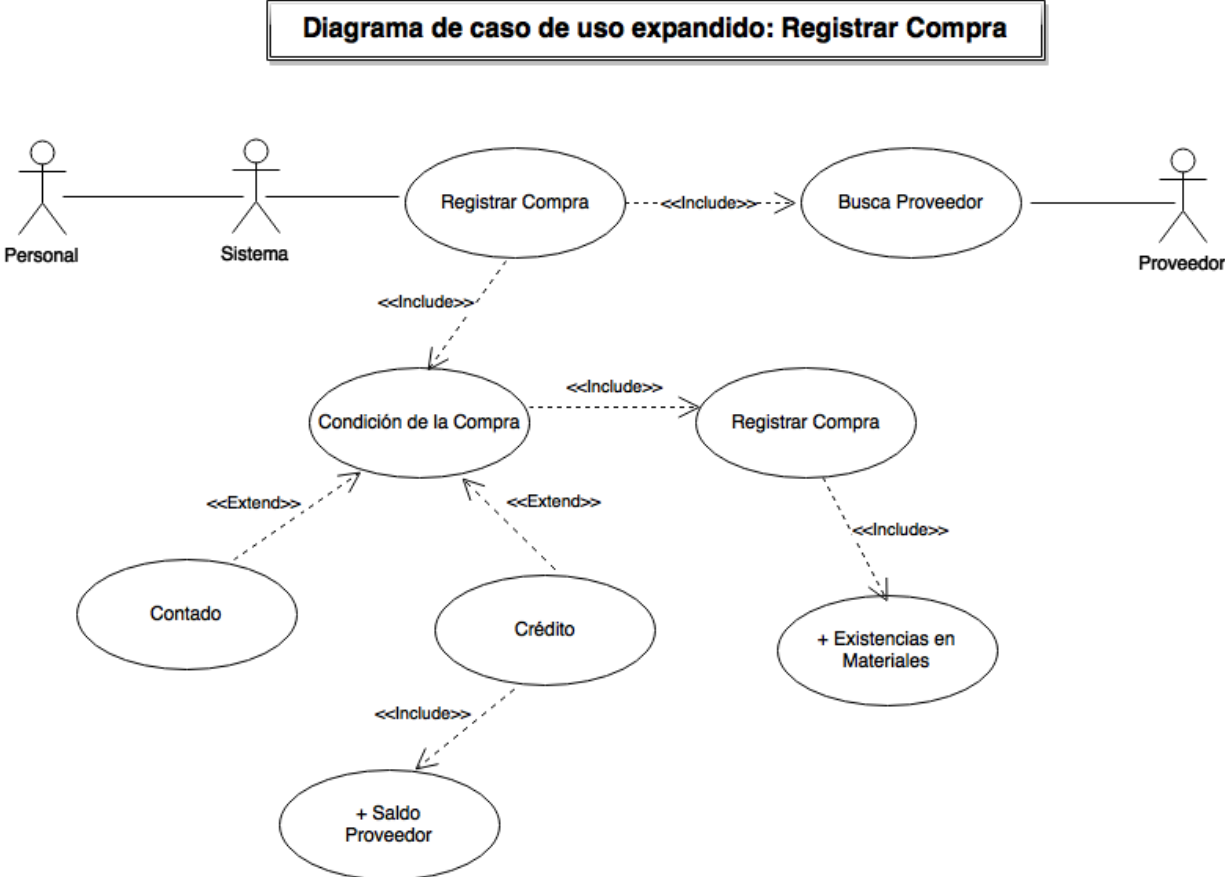


Figura 22. Diagrama de caso de uso expandido: Registrar compra.

4.1.13 Diagrama de flujo de datos de registrar pago

En la figura No. 23 se muestra el diagrama correspondiente.

Diagrama de flujo de datos: Registrar pago

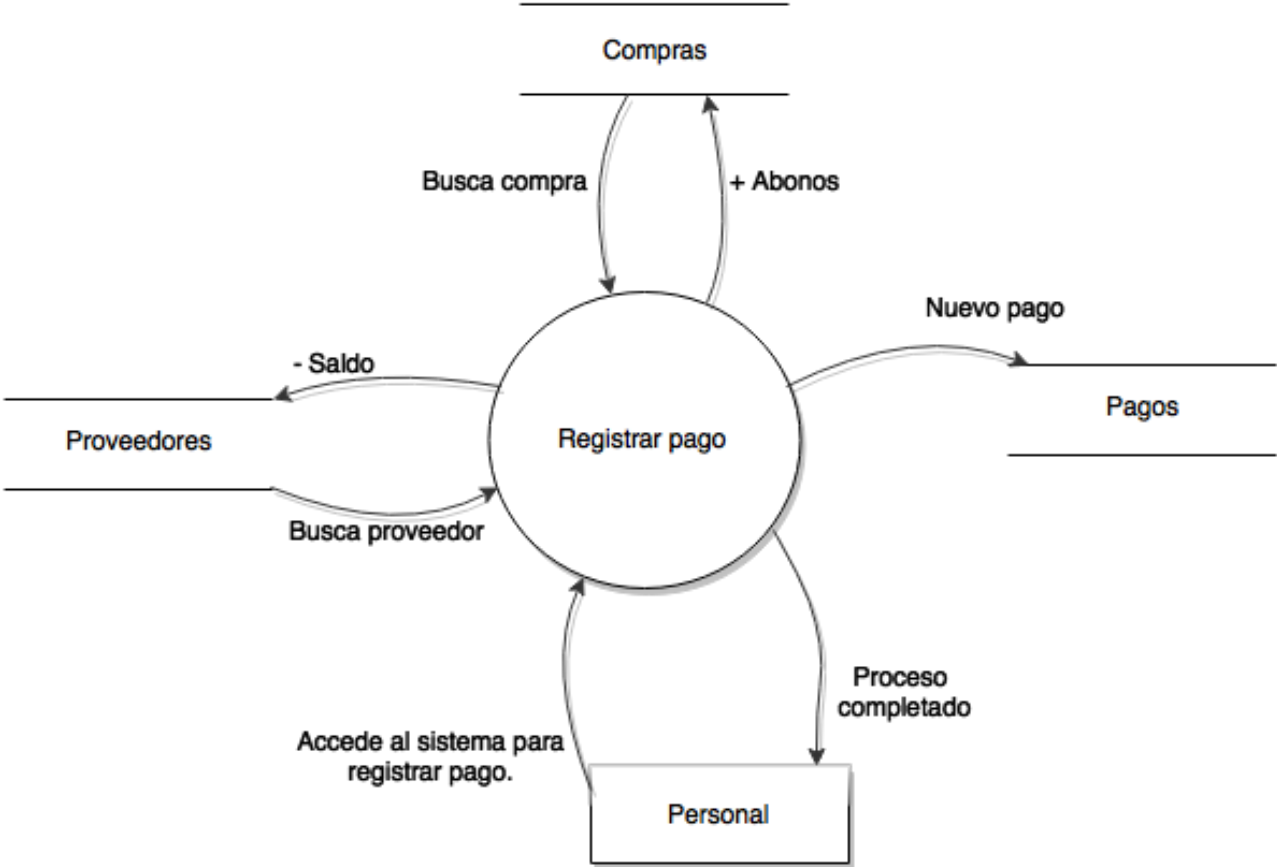


Figura 23. Diagrama de flujo de datos: Registrar pago.

4.1.14 Diagrama de estados de registrar pago

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del proveedor y visualiza sus datos. Posteriormente busca compras con saldo pendiente, selecciona la compra, visualiza el saldo y proporciona la cantidad, finalmente graba el pago.

Precondición: El personal decide hacer un pago a un proveedor.

Caminos alternos: Solo consulta el saldo de los proveedores o pagos próximos a vencer.

Postcondición: Termina cuando el personal graba el pago, generando respectivamente el registro y actualizando automáticamente el saldo del proveedor. (Ver figura No. 24).

Diagrama de estados: Registrar pago

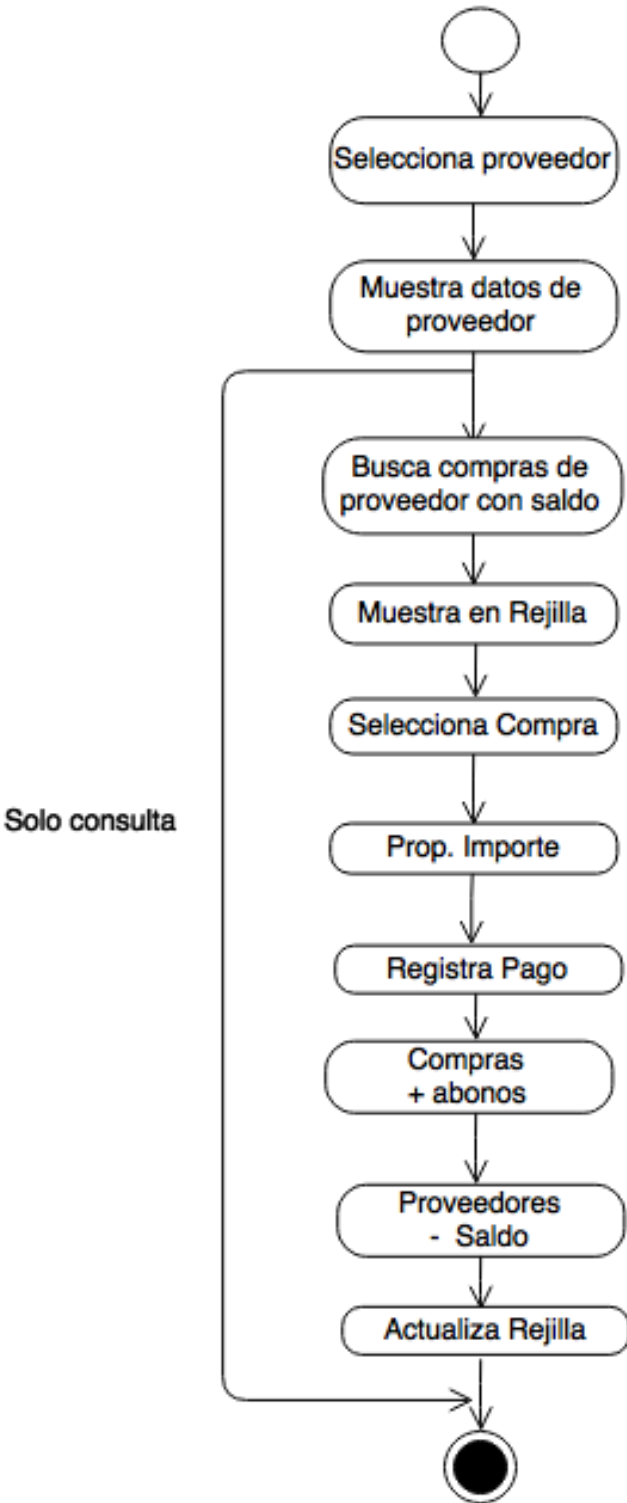


Figura 24. Diagrama de estados: Registrar pago.

4.1.15 Diagrama de colaboración de registrar pago

En la figura No. 25 se muestra el diagrama correspondiente.

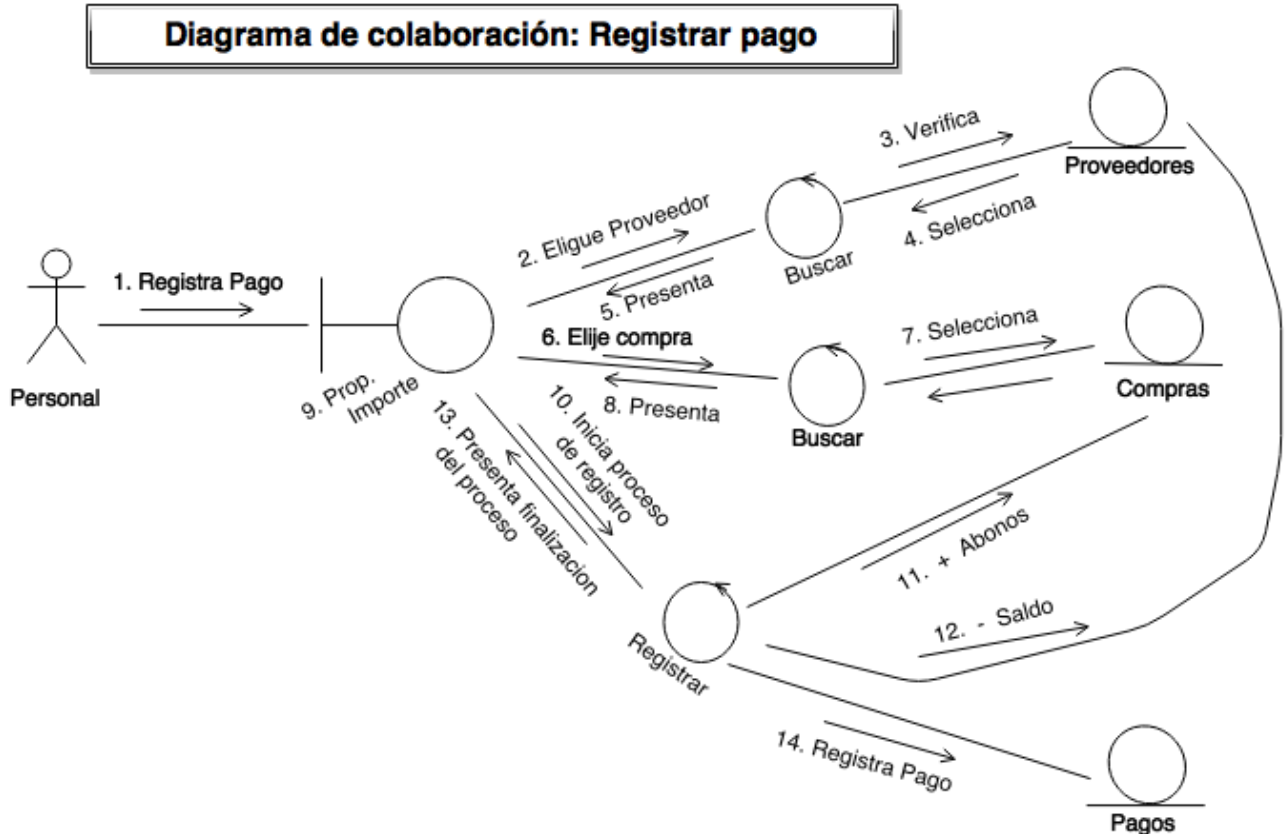


Figura 25. Diagrama de colaboración: Registrar pago.

4.1.16 Diagrama de caso de uso expandido de registrar pago

Nombre del caso: Registrar pago

Actores: Personal, Proveedor.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar un pago, busca el proveedor correspondiente y visualiza los datos. Posteriormente busca las compras que tengan saldo pendiente, decide entre realizar pago o solo consultar el saldo, si decide realizar pago entonces lo registra, actualizando automáticamente el saldo del proveedor y los abonos en la compra correspondiente.

En la figura No. 26 se muestra el diagrama correspondiente.

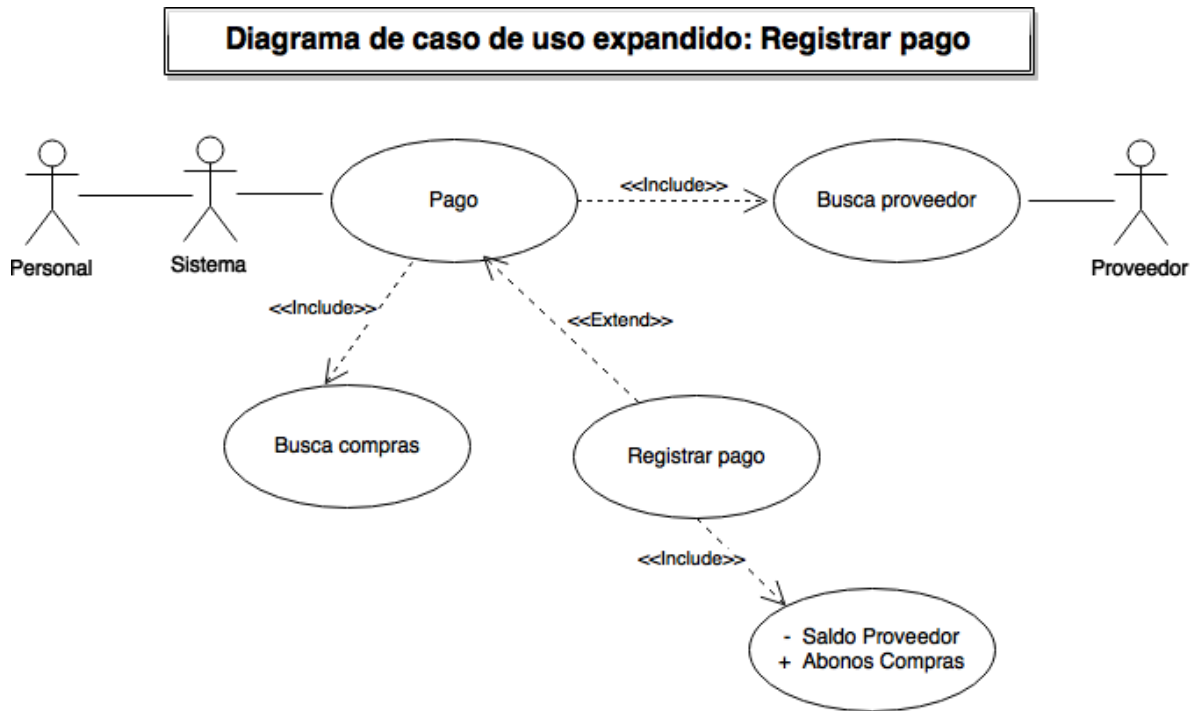


Figura 26. Diagrama de caso de uso expandido: Registrar pago.

4.1.17 Diagrama de flujo de datos de registrar venta

En la figura No. 27 se muestra el diagrama de flujo de datos correspondiente.

Diagrama de flujo de datos: Registrar venta

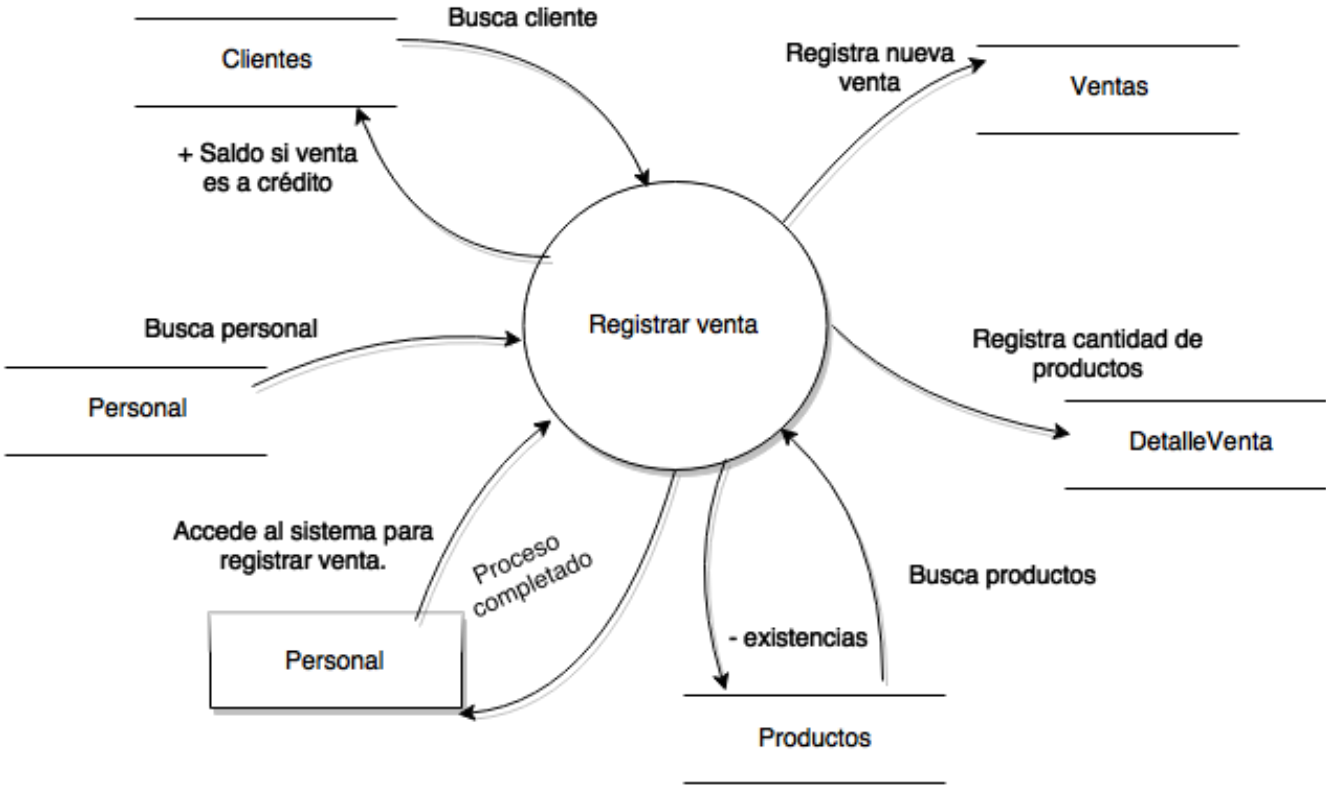


Figura 27. Diagrama de flujo de datos: Registrar venta.

4.1.18 Diagrama de estados de registrar venta

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del cliente y visualiza sus datos. Posteriormente selecciona la descripción del producto a vender y visualiza los datos, proporciona la cantidad, agrega los datos a la rejilla de productos y de la misma manera para cada uno de los productos a vender. Finalmente selecciona la condición de la venta si es a crédito o contado y en seguida graba la venta.

Precondición: Registra más artículos, registra el cliente en catálogo, registra producto en catálogo, actualiza saldo del cliente si es crédito.

Caminos alternos: No tiene.

Postcondición: Termina cuando el personal graba la venta, actualizando automáticamente la existencia en el inventario de Productos y generando el registro en Ventas con su respectivo detalle. (Ver figura No. 28).

Diagrama de estados: Registrar venta

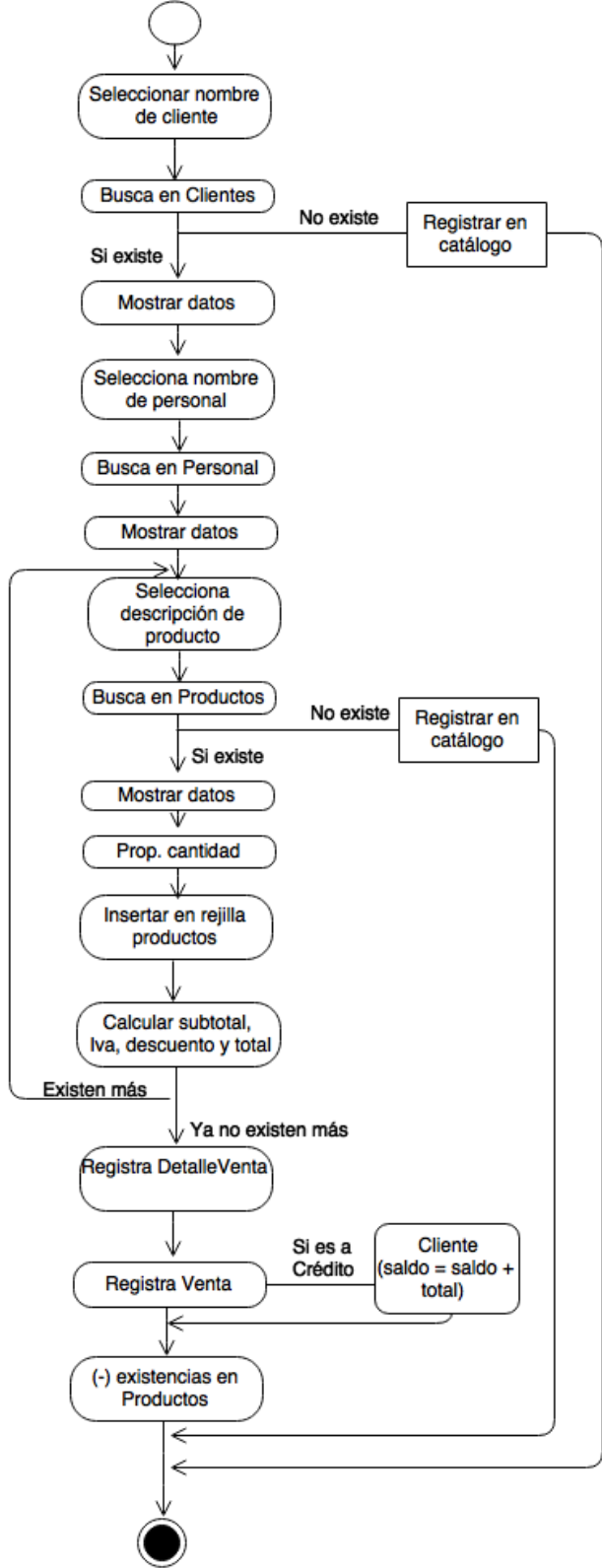


Figura 28. Diagrama de estados: Registrar venta.

4.1.19 Diagrama de colaboración de registrar venta

En la figura No. 29 se muestra el diagrama de colaboración correspondiente.

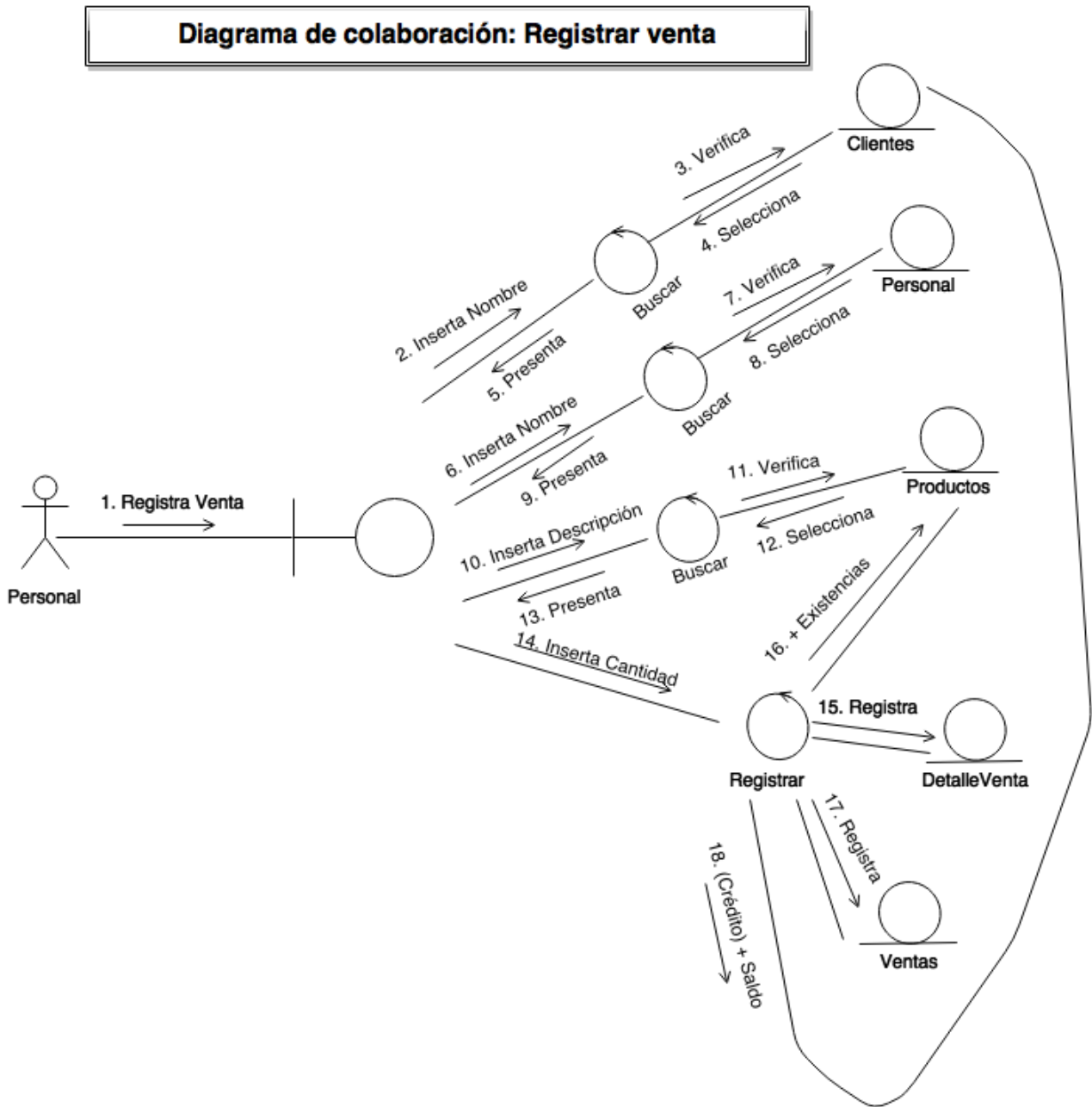


Figura 29. Diagrama de colaboración: Registrar venta.

4.1.20 Diagrama de caso de uso expandido de registrar venta

Nombre del caso: Registrar venta

Actores: Personal, Cliente.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar una venta, busca al cliente correspondiente y lo selecciona. Posteriormente elige la condición de la venta si es a crédito o contado. Finalmente al guardar actualiza automáticamente las existencias en Productos y si la venta fue a crédito incrementa el saldo del cliente.

En la figura No. 30 se muestra el diagrama de caso de uso expandido correspondiente.

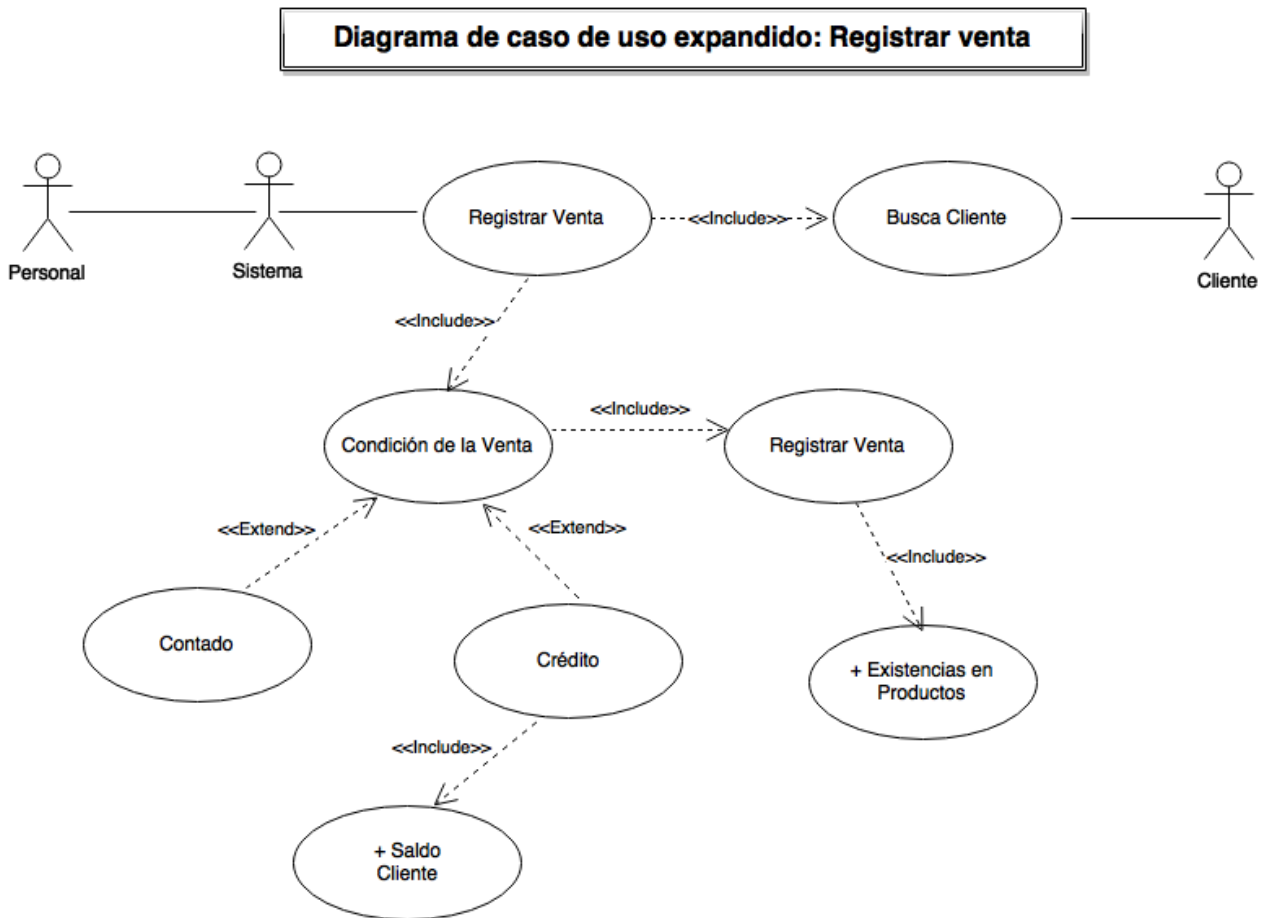


Figura 30. Diagrama de caso de uso expandido: Registrar compra.

4.1.21 Diagrama de flujo de datos de registrar abono

En la figura No. 31 se muestra el diagrama de flujo de datos de registrar abono.

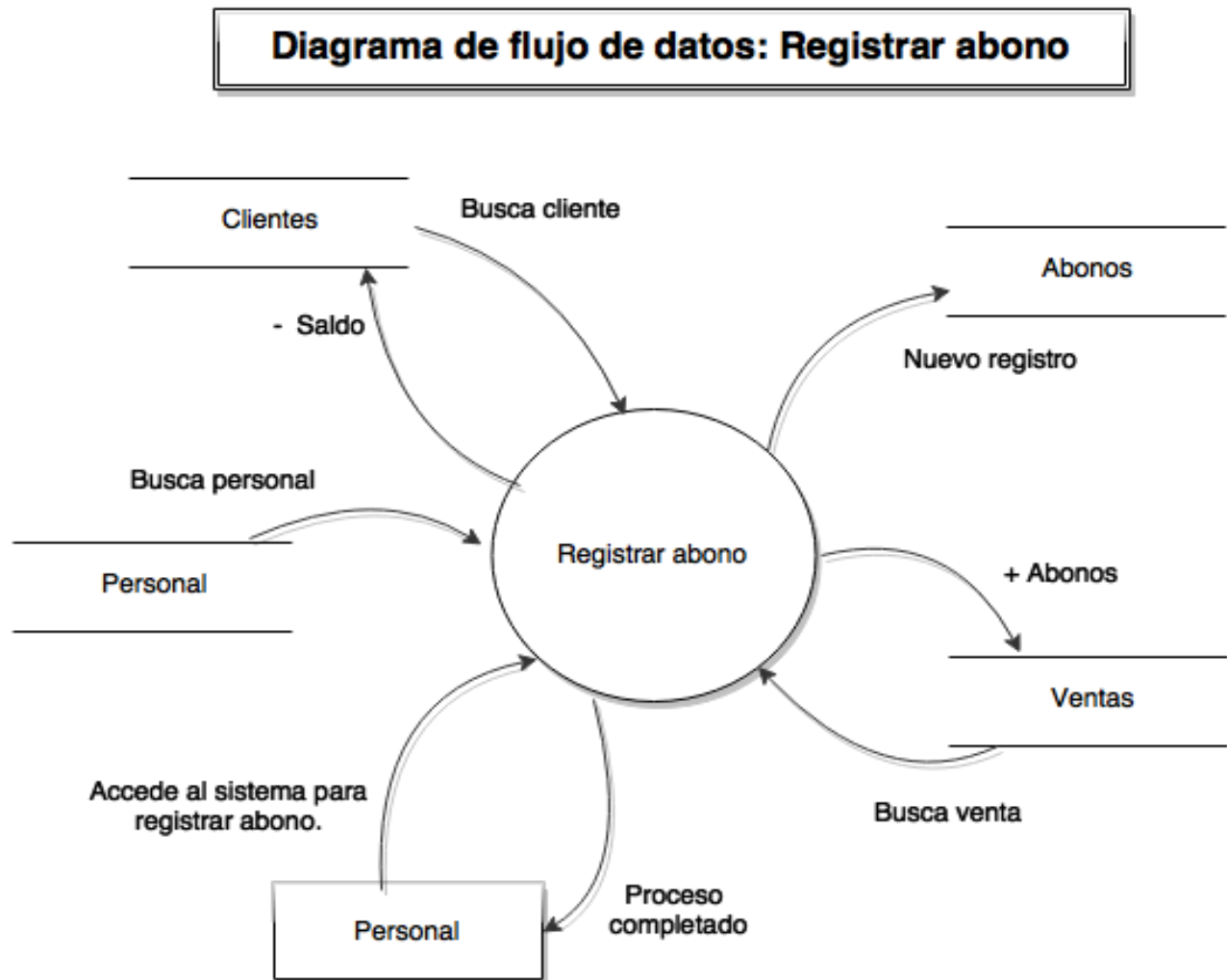


Figura 31. Diagrama de flujo de datos: Registrar abono.

4.1.22 Diagrama de estados de registrar abono

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del cliente y visualiza sus datos. Posteriormente busca las ventas con saldo y las visualiza, selecciona la venta, visualiza el saldo y proporciona la cantidad del abono, finalmente graba.

Precondición: El cliente llega a realizar un abono a la empresa.

Caminos alternos: Solo consulta lo que debe el cliente o abonos próximos a vencer.

Postcondición: Termina cuando el personal graba el abono, generando respectivamente el registro y actualizando automáticamente los abonos de la venta y el saldo del cliente. Ver figura No. 32.

Diagrama de estados: Registrar abono

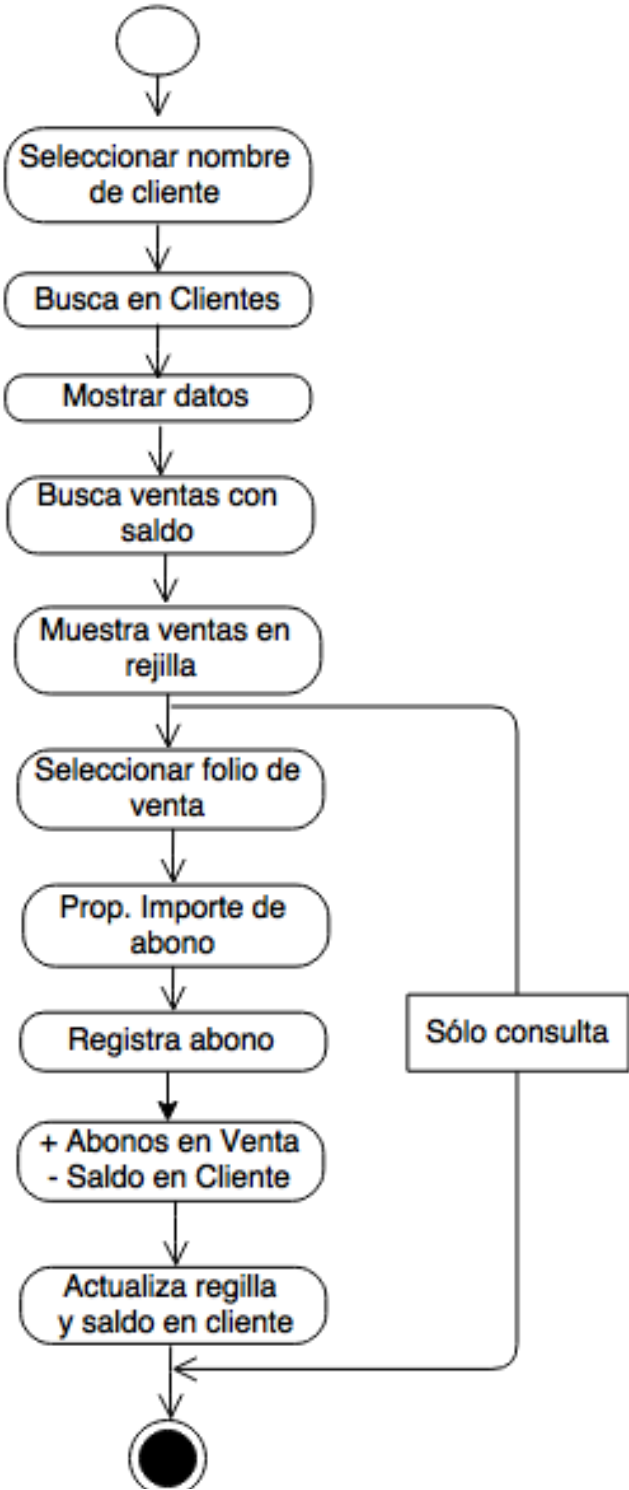


Figura 32. Diagrama de estados: Registrar abono.

4.1.23 Diagrama de colaboración de registrar abono

En la figura No. 33 se muestra el diagrama de colaboración correspondiente.

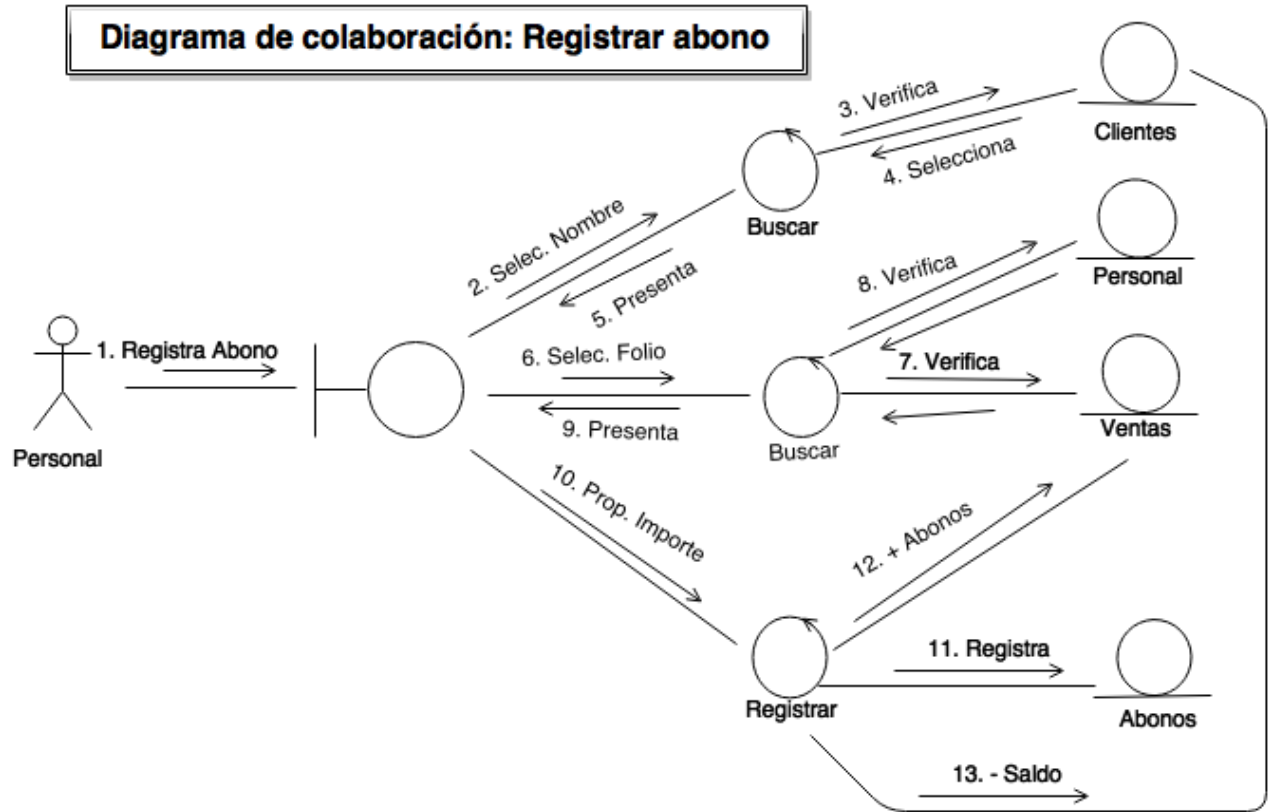


Figura 33. Diagrama de colaboración: Registrar abono.

4.1.24 Diagrama de caso de uso expandido de registrar abono

Nombre del caso: Registrar abono

Actores: Personal, Cliente.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar un abono, busca al cliente que lo solicitó y visualiza los datos. Posteriormente busca las ventas con saldo pendiente, el cliente decide entre realizar el abono o solo consultar lo que debe (su saldo), si decide realizar abono entonces lo registra, actualizando automáticamente los abonos en la venta correspondiente y el saldo del cliente.

En la figura No. 34 se muestra el diagrama de caso de uso expandido de registrar abono.

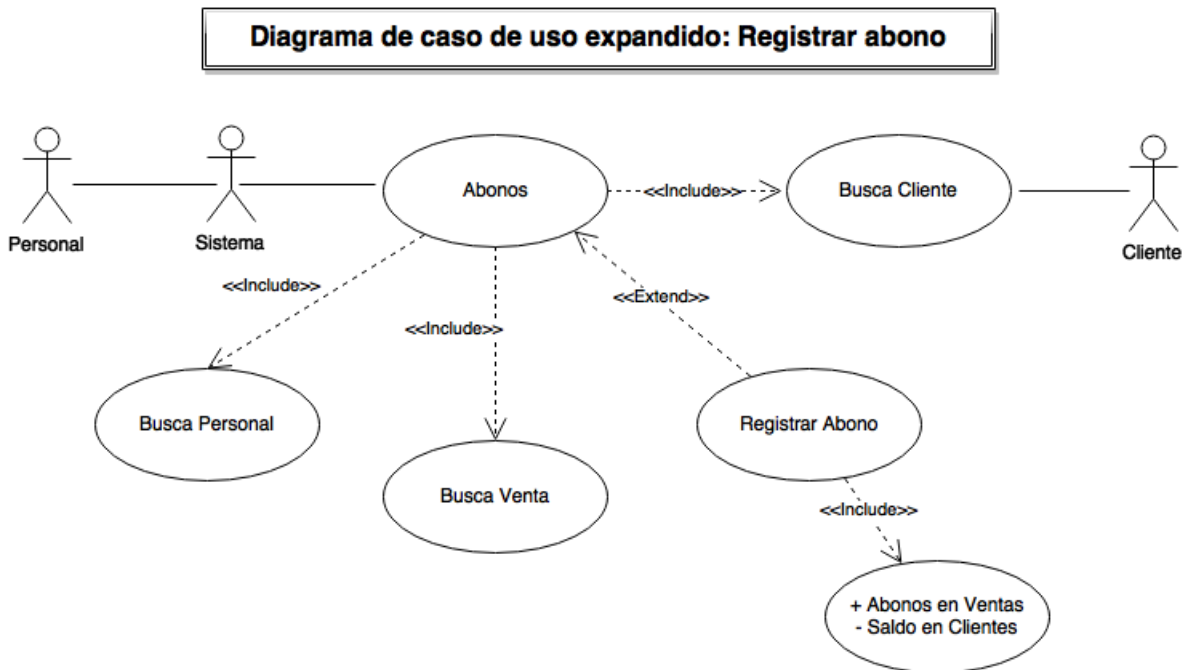


Figura 34. Diagrama de caso de uso expandido: Registrar abono.

4.1.25 Diagrama de flujo de datos de registrar pedido

En la figura No. 35 se muestra el diagrama de flujo de datos correspondiente.

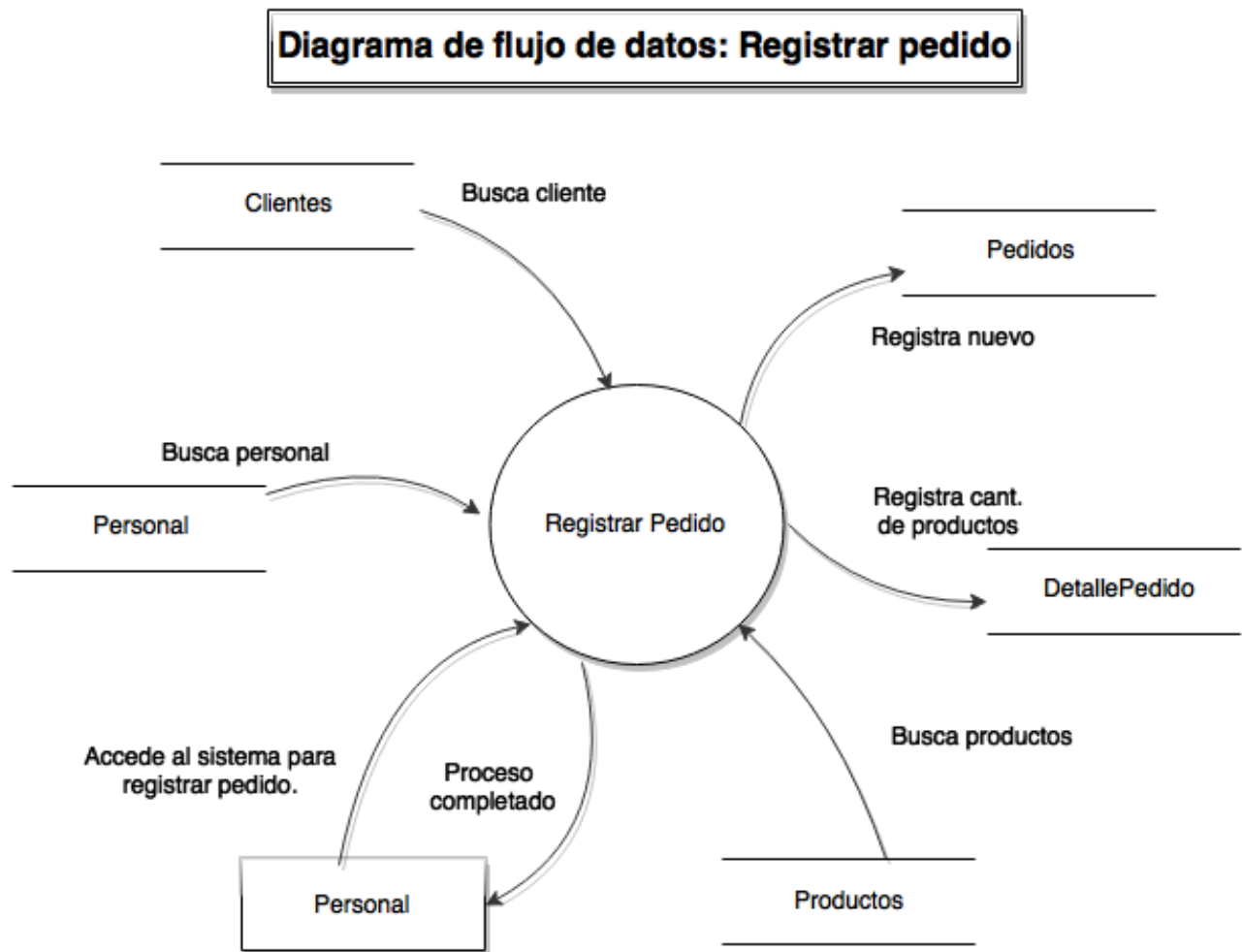


Figura 35. Diagrama de flujo de datos: Registrar pedido.

4.1.26 Diagrama de estados de registrar pedido

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del cliente, el personal busca su nombre, lo selecciona y visualiza sus datos. Posteriormente selecciona la descripción del producto a registrar en pedido y visualiza los datos, proporciona la cantidad, agrega los datos a la rejilla de productos y de la misma manera para cada uno de los productos a registrar. Finalmente graba el pedido.

Precondición: El cliente llega a solicitar un pedido.

Caminos alternos: Registra más productos, registra clientes en catálogo, registra producto en catálogo.

Postcondición: Termina cuando el personal graba el pedido, generando el registro en Pedidos con su respectivo detalle. (Ver figura No. 36).

Diagrama de estados: Registrar pedido

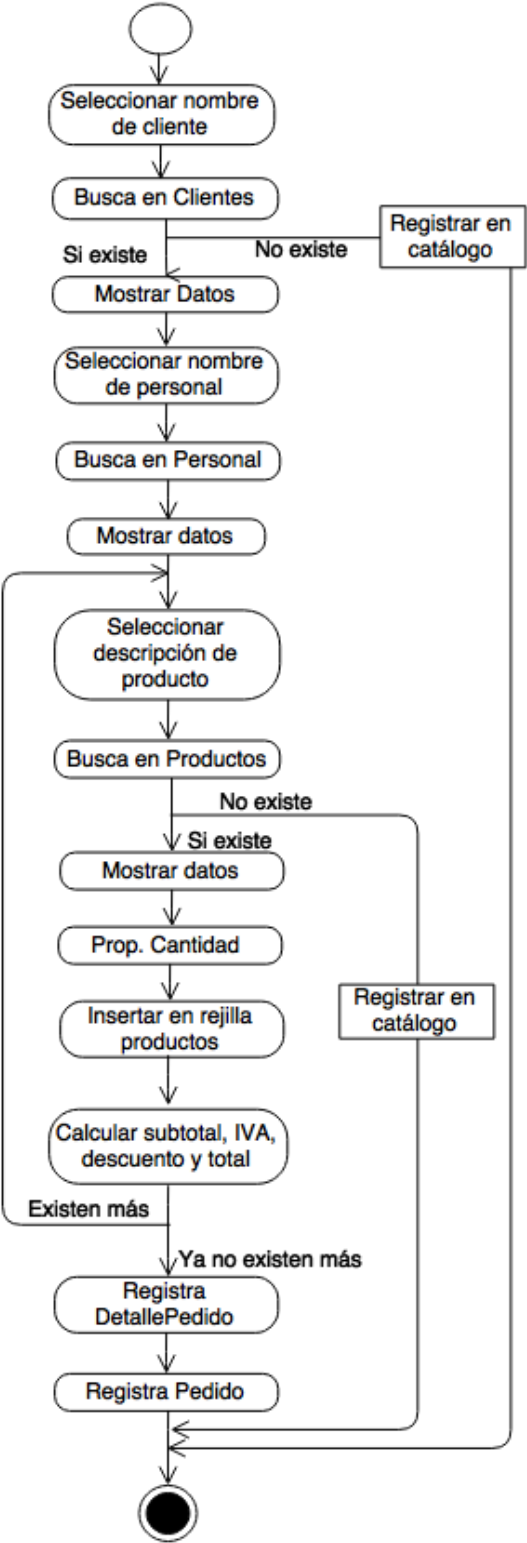


Figura 36. Diagrama de estados: Registrar pedido.

4.1.27 Diagrama de colaboración de registrar pedido

En la figura No. 37 se muestra el diagrama de colaboración correspondiente.

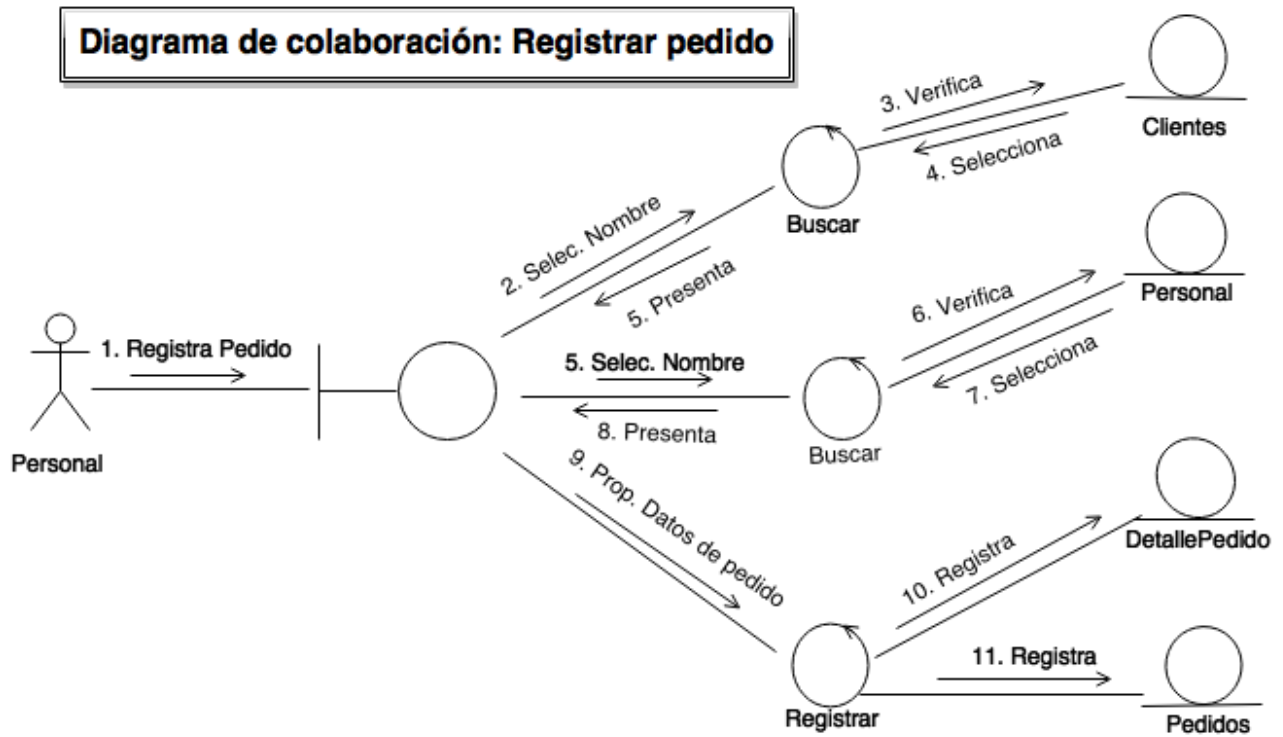


Figura 37. Diagrama de colaboración: Registrar pedido.

4.1.28 Diagrama de caso de uso expandido de registrar pedido

Nombre del caso: Registrar pedido

Actores: Personal, Cliente.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar un pedido, busca al cliente correspondiente y lo selecciona, el personal busca su nombre y también lo elije. Finalmente graba el pedido y se genera el registro en Pedidos con su respectivo detalle.

En la figura No. 38 se muestra el diagrama de caso de uso expandido correspondiente.

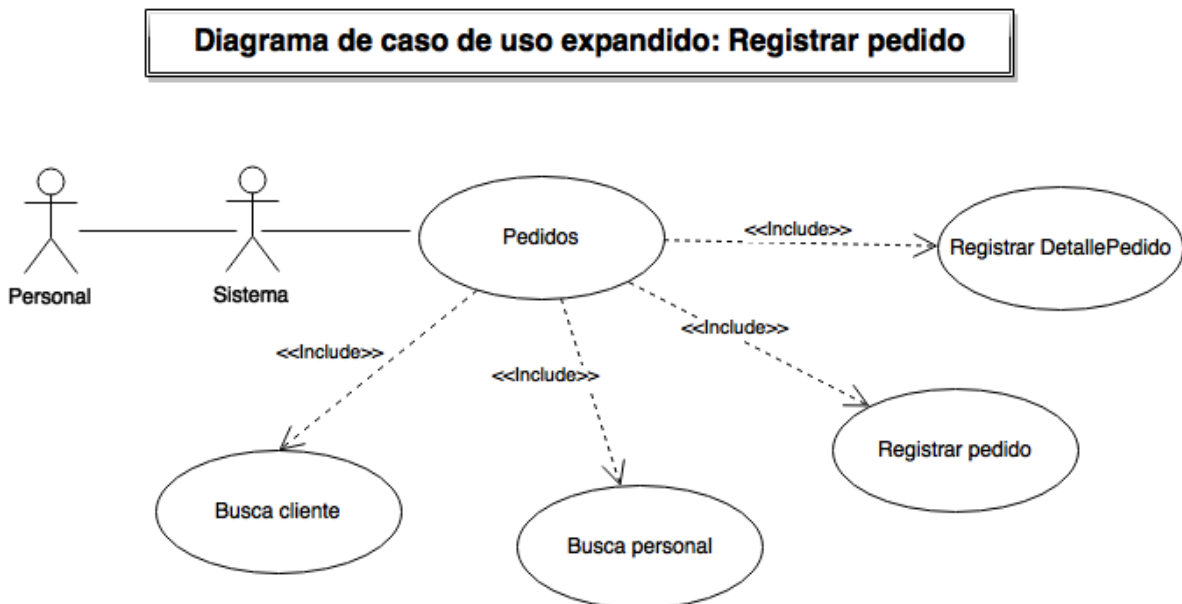


Figura 38. Diagrama de caso de uso expandido: Registrar pedido.

4.1.29 Diagrama de flujo de datos de registrar gastos de producción

En la figura No. 39 se muestra el diagrama de flujo de datos correspondiente.

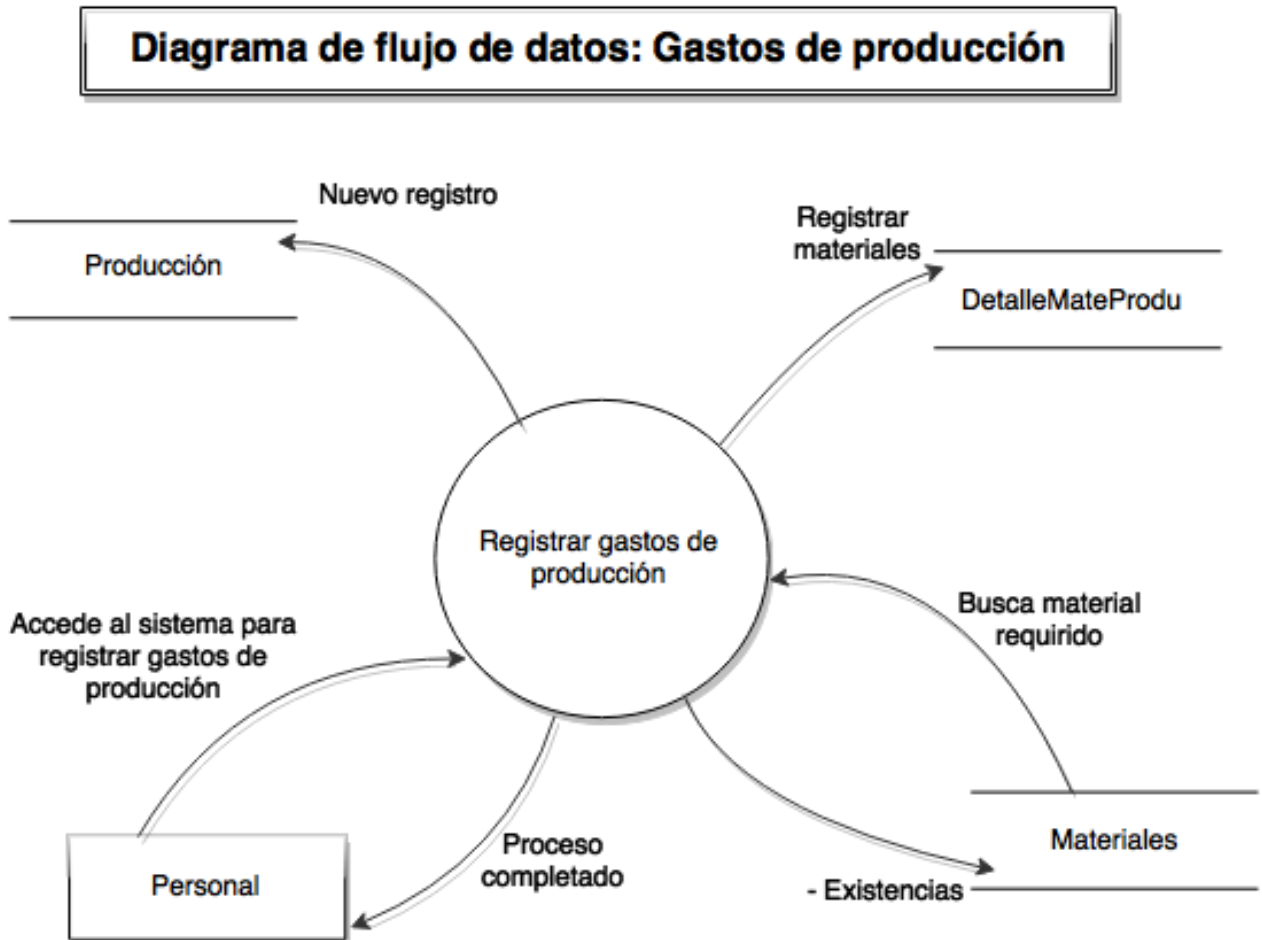


Figura 39. Diagrama de flujo de datos: Registrar gastos de producción.

4.1.30 Diagrama de estados de registrar gastos de producción

Flujo de eventos: El personal de la empresa asigna la fecha inicial y final de la producción, luego describe la producción que se llevará a cabo. Posteriormente selecciona la descripción del material que va a utilizar y visualiza los datos, proporciona la cantidad de material, agrega los datos a la rejilla y de la misma manera realizará la agregación de todos los materiales necesarios para la producción. Finalmente registra el gasto de producción con todos los materiales necesarios.

Precondición: El personal de la empresa solicita material para fabricar los muebles de un pedido.

Caminos alternos: Ingresar más materiales.

Postcondición: Termina cuando el personal graba el gasto de producción, actualizando automáticamente la existencia en el inventario de Materiales y generando los registros correspondientes. Ver figura No. 40.

Diagrama de estados: Gastos de producción

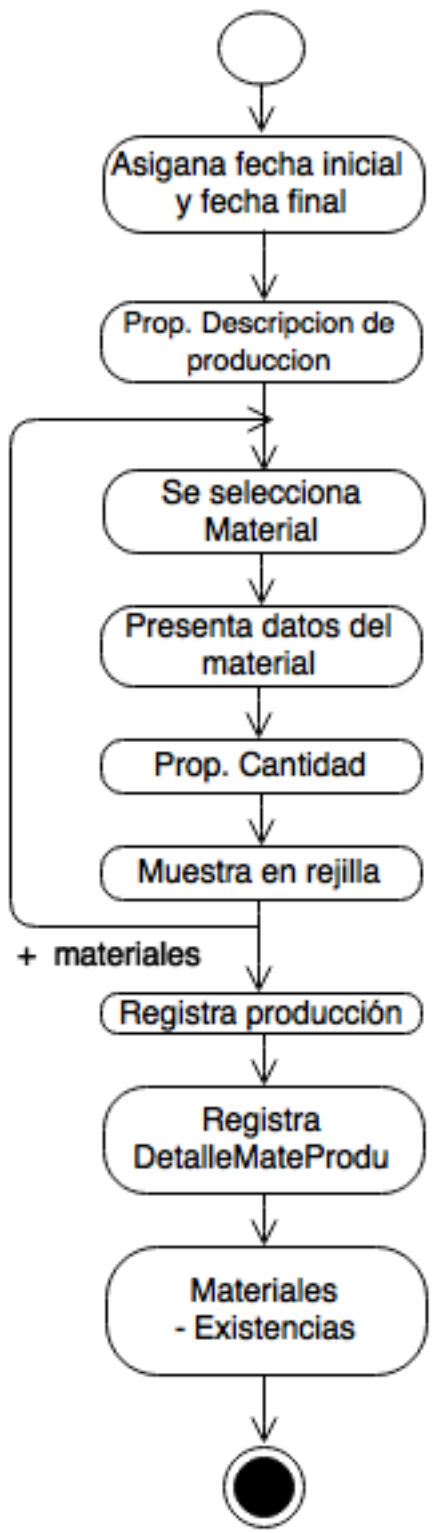


Figura 40. Diagrama de estados: Registrar gastos de producción.

4.1.31 Diagrama de colaboración de registrar gastos de producción

En la figura No. 41 se muestra el diagrama de colaboración correspondiente.

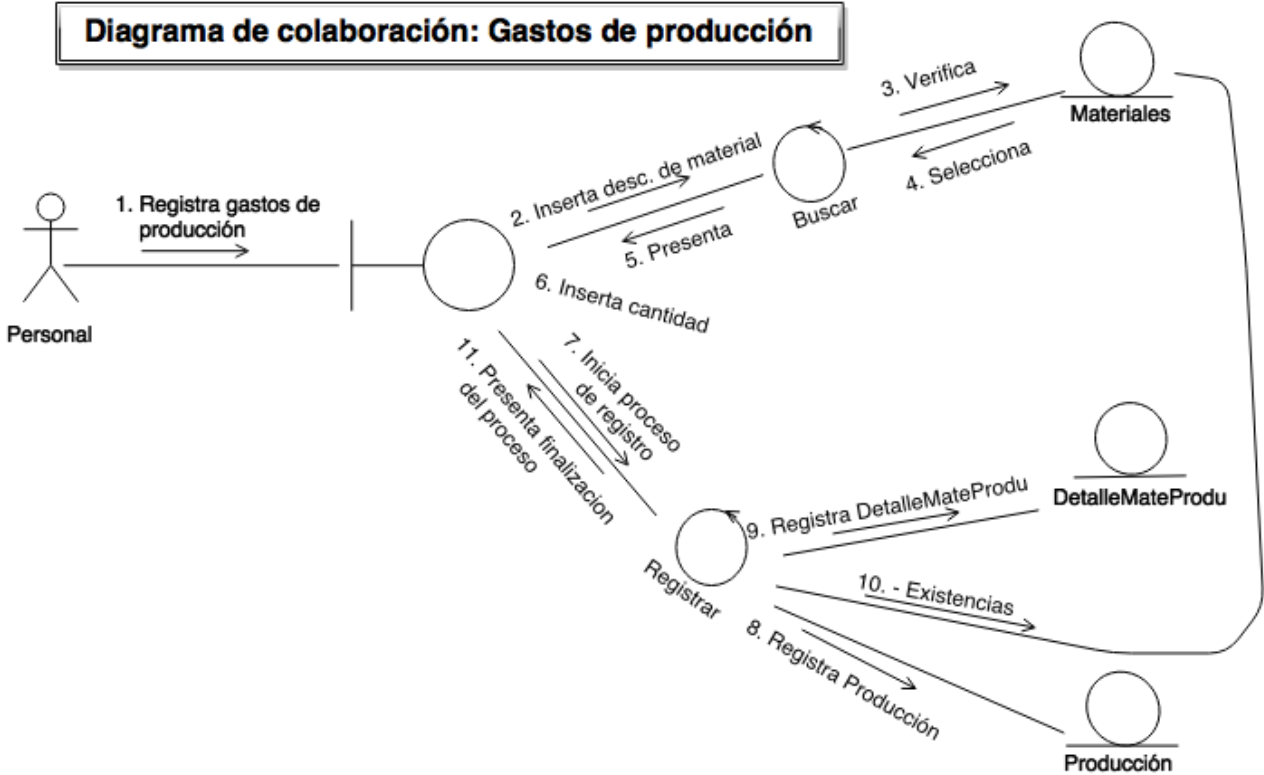


Figura 41. Diagrama de colaboración: Registrar gastos de producción.

4.1.32 Diagrama de caso de uso expandido de registrar gastos de producción

Nombre del caso: Registrar gastos de producción.

Actores: Personal.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar los gastos para una producción, busca el material necesario, lo selecciona y registra. Finalmente al guardar disminuye automáticamente las existencias en Materiales.

En la figura No. 42 se muestra el diagrama de caso de uso expandido de registrar gastos de producción.

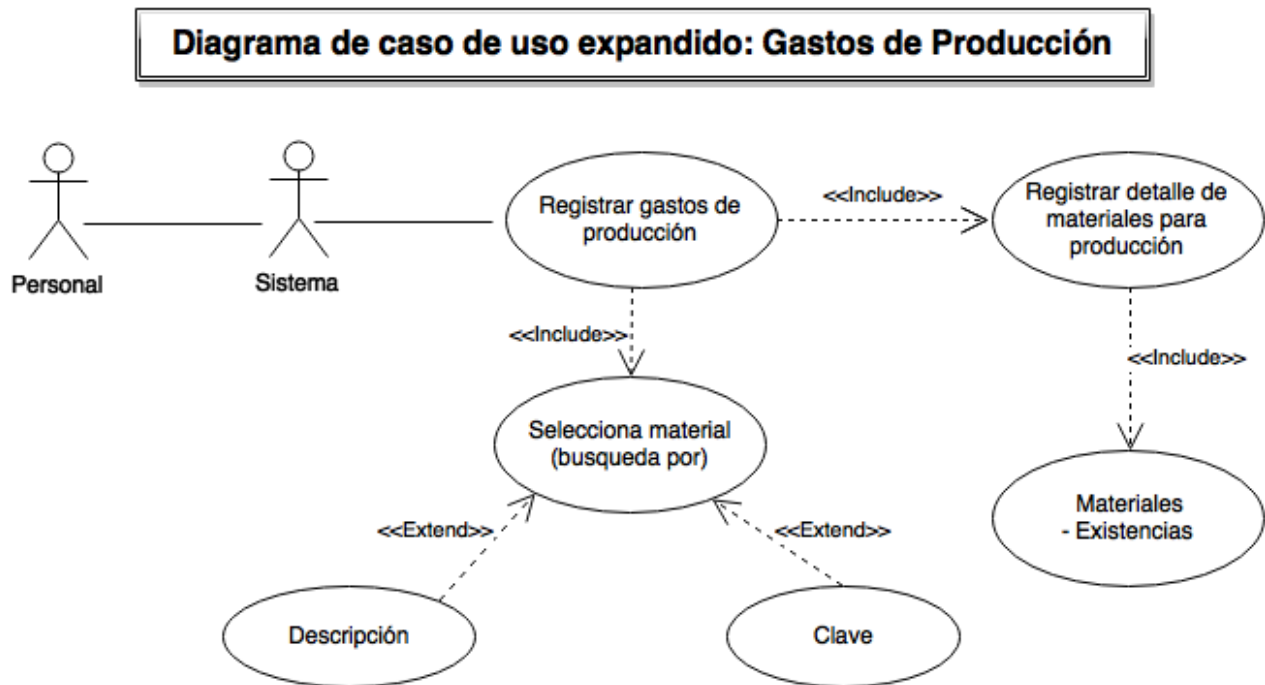


Figura 42. Diagrama de caso de uso expandido: Registrar gastos de producción.

4.1.33 Diagrama de flujo de datos de registrar material sobrante de producción

En la figura No. 43 se muestra el diagrama de flujo de datos correspondiente.

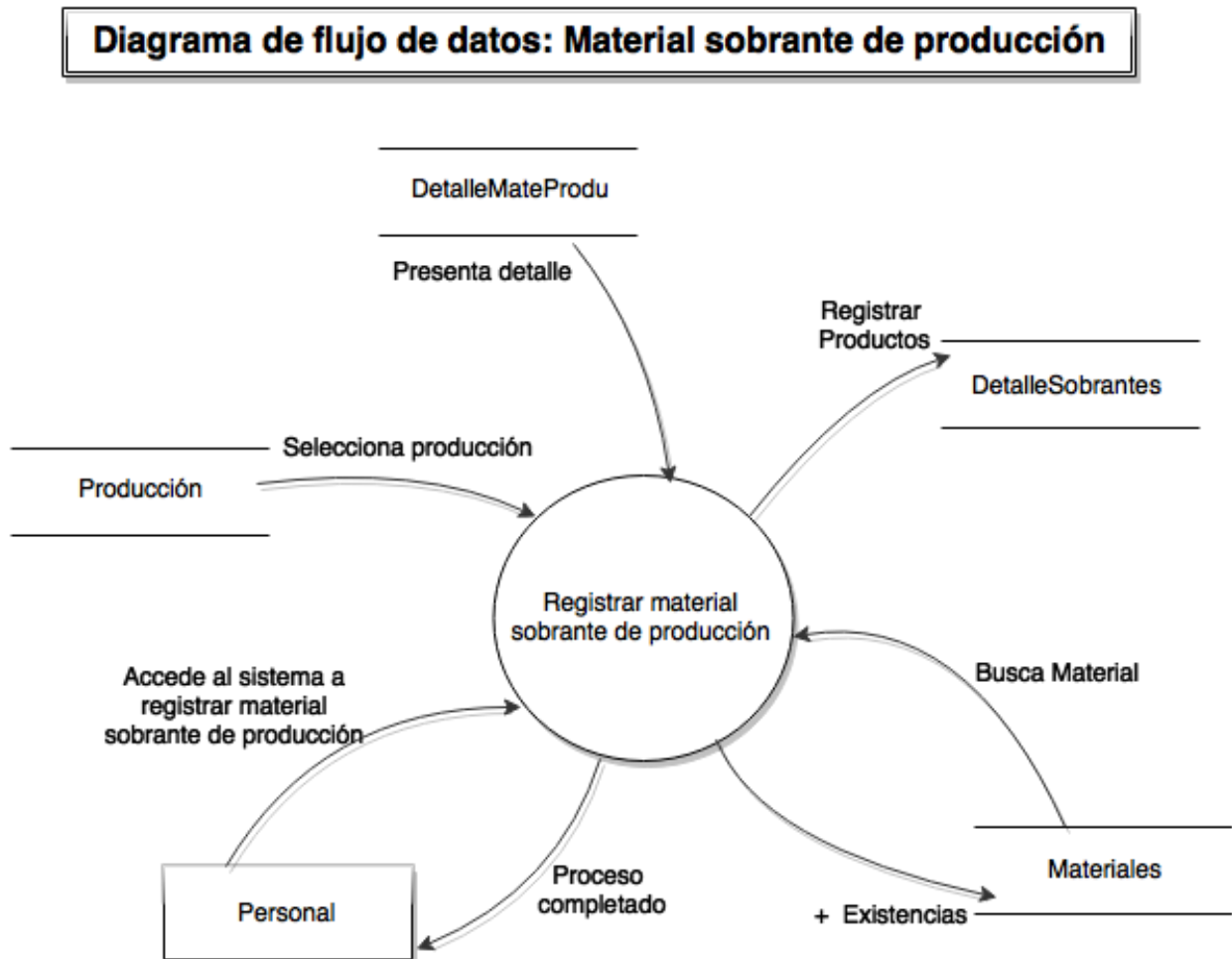


Figura 43. Diagrama de flujo de datos: Registrar material sobrante de producción.

4.1.34 Diagrama de estados de registrar material sobrante de producción.

Flujo de eventos: El personal de la empresa visualiza las producciones en pantalla, selecciona la producción de la cual se va a regresar material sobrante y presenta los materiales que se utilizaron. Posteriormente selecciona el material que corresponda al que se va a regresar e introduce la cantidad de material que se está regresando y lo agrega a la rejilla, de la misma manera realizará la agregación de todos los materiales que sobraron de esa producción. Finalmente registra todo el material sobrante de producción.

Precondición: El personal de la empresa regresa el material sobrante de una producción.

Caminos alternos: Agregar más materiales.

Postcondición: Termina cuando el personal registra el material sobrante de producción, actualizando automáticamente la existencia en el inventario de Materiales y generando los registros correspondientes. (Ver figura No. 44).

Diagrama de estados: Material sobrante de producción

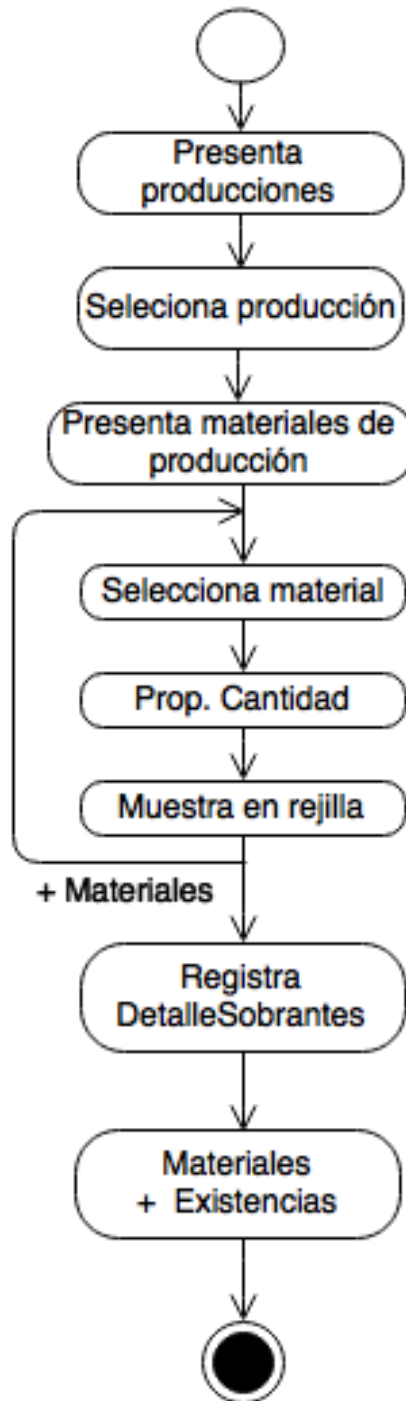


Figura 44. Diagrama de estados: Registrar material sobrante de producción.

4.1.35 Diagrama de colaboración de registrar material sobrante de producción

En la figura No. 45 se muestra el diagrama de colaboración correspondiente.

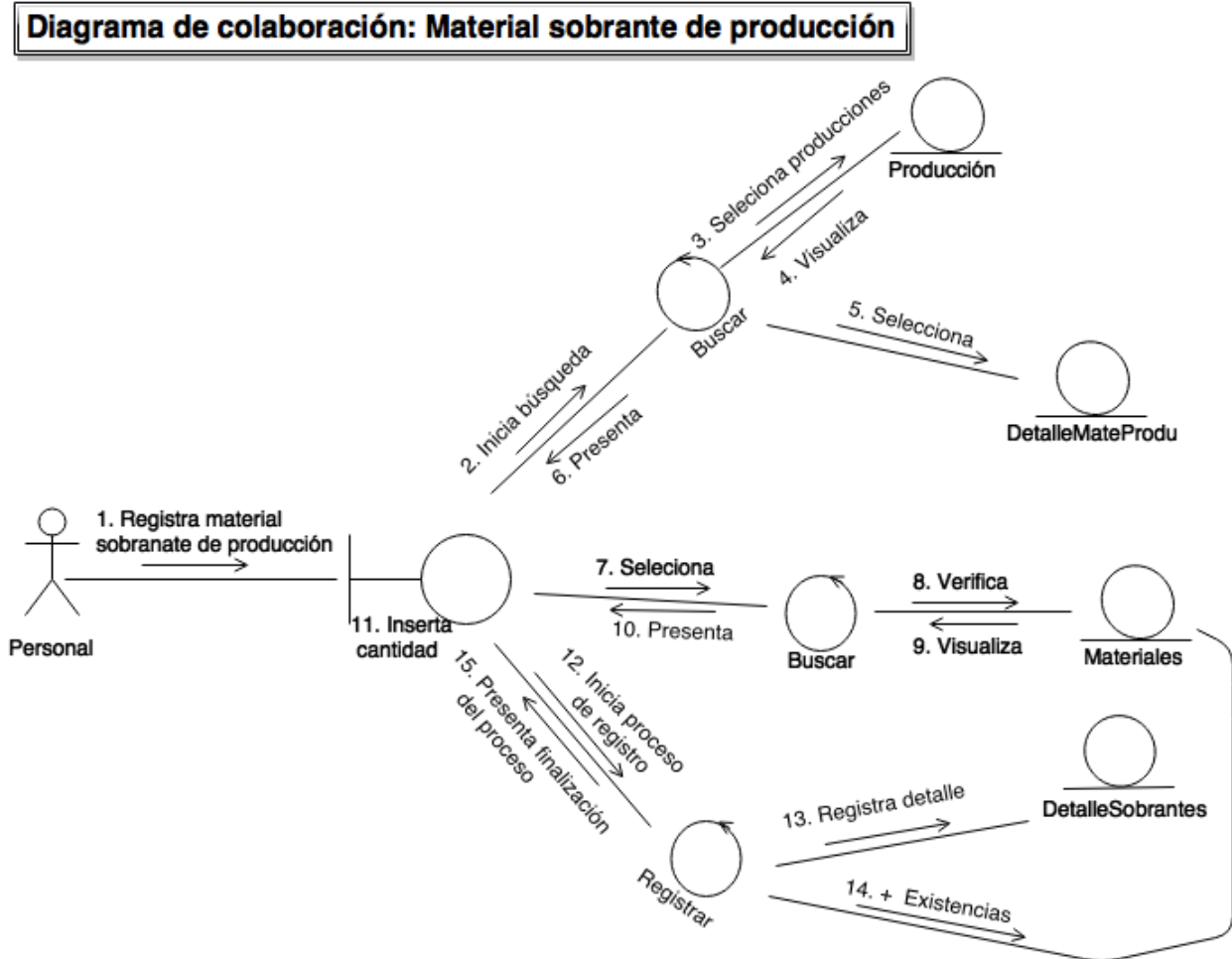


Figura 45. Diagrama de colaboración: Registrar material sobrante de producción.

4.1.36 Diagrama de caso de uso expandido de registrar material sobrante de producción

Nombre del caso: Registrar material sobrante de producción.

Actores: Personal.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar el material sobrante de una producción, busca la producción correspondiente y visualiza los materiales utilizados, selecciona el material e introduce la cantidad a regresar. Finalmente al grabar incrementa automáticamente las existencias en Materiales.

En la figura No. 46 se muestra el diagrama de caso de uso expandido de registrar material sobrante de producción.

Diagrama de caso de uso expandido: Material sobrante de producción

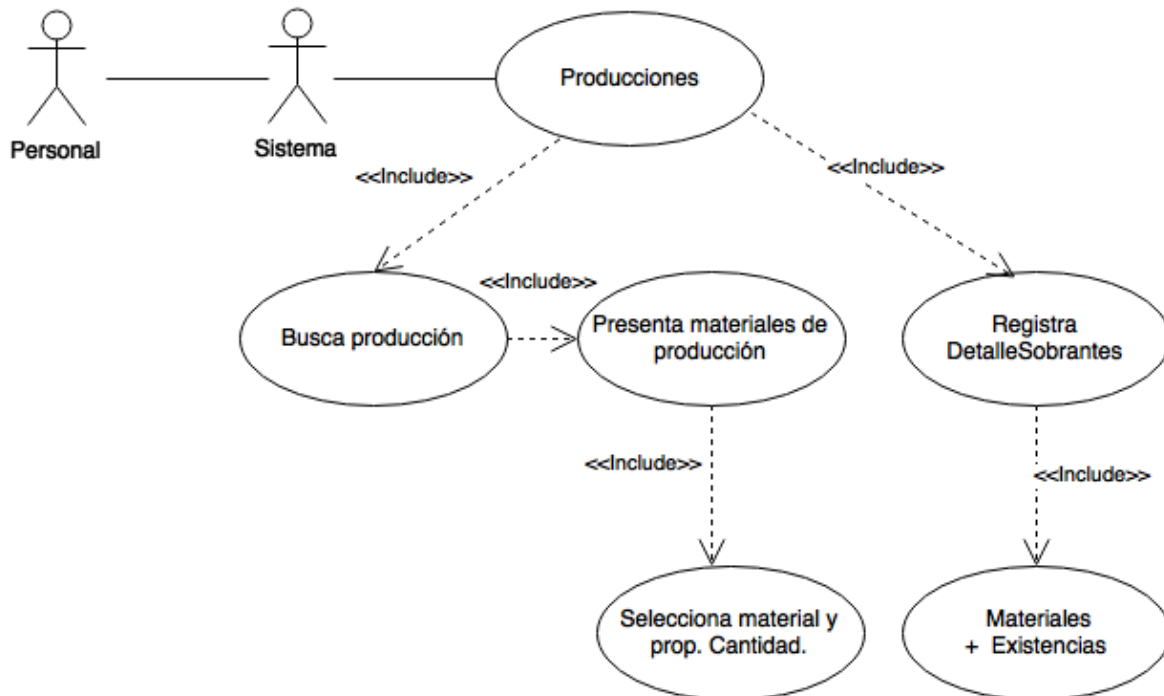


Figura 46. Diagrama de caso de uso expandido: Registrar material sobrante de producción.

4.1.37 Diagrama de flujo de datos de registrar productos terminados

En la figura No. 47 se muestra el diagrama de flujo de datos correspondiente.

Diagrama de flujo de datos: Productos terminados

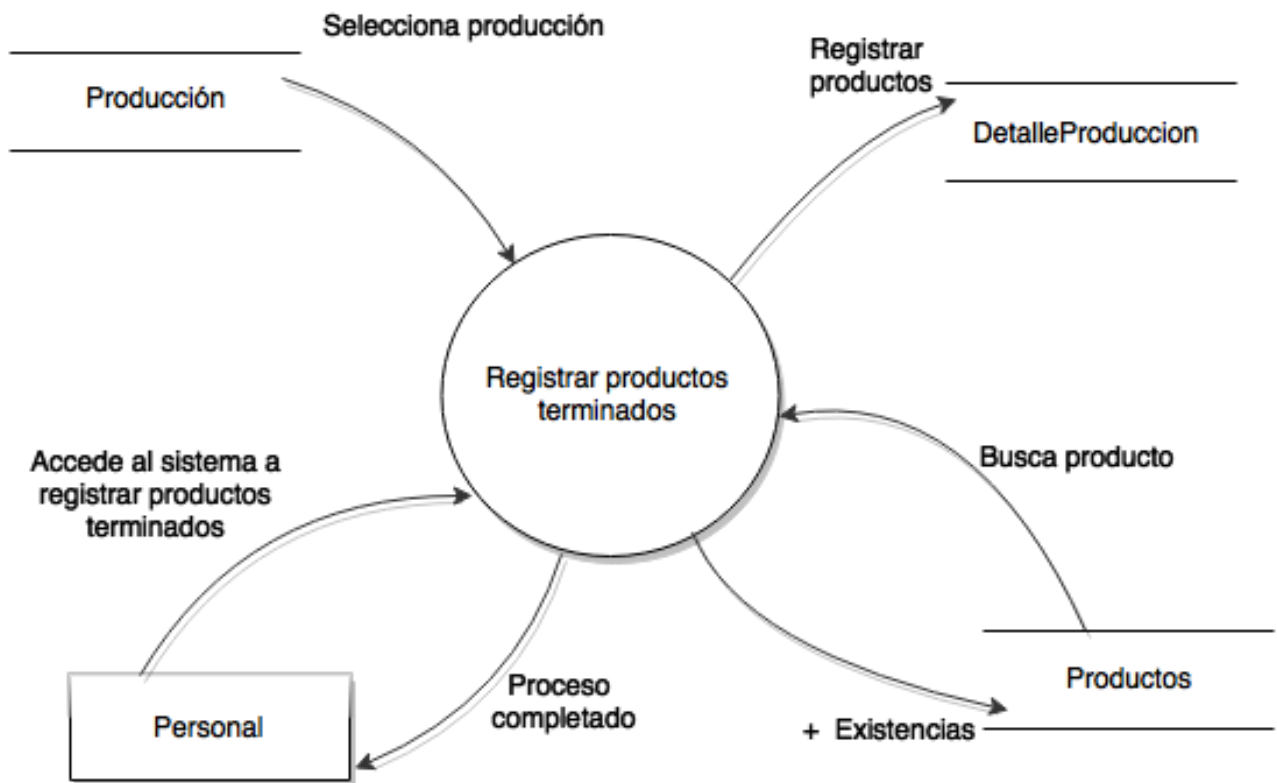


Figura 47. Diagrama de flujo de datos: Registrar productos terminados.

4.1.38 Diagrama de estados de registrar productos terminados

Flujo de eventos: El personal de la empresa visualiza las producciones en pantalla, selecciona la producción de la cual se van a registrar los productos terminados. Posteriormente selecciona el producto, introduce la cantidad y lo agrega a la rejilla, de la misma manera realizará la agregación de todos los productos terminados de esa producción. Finalmente guarda la producción.

Precondición: El personal de la empresa da el reporte de la producción realizada.

Caminos alternos: Registrar más productos.

Postcondición: Termina cuando el personal registra los productos terminados de la producción, incrementando automáticamente la existencia en el inventario de Productos y generando los registros correspondientes. (Ver figura No. 48).

Diagrama de estados: Productos terminados

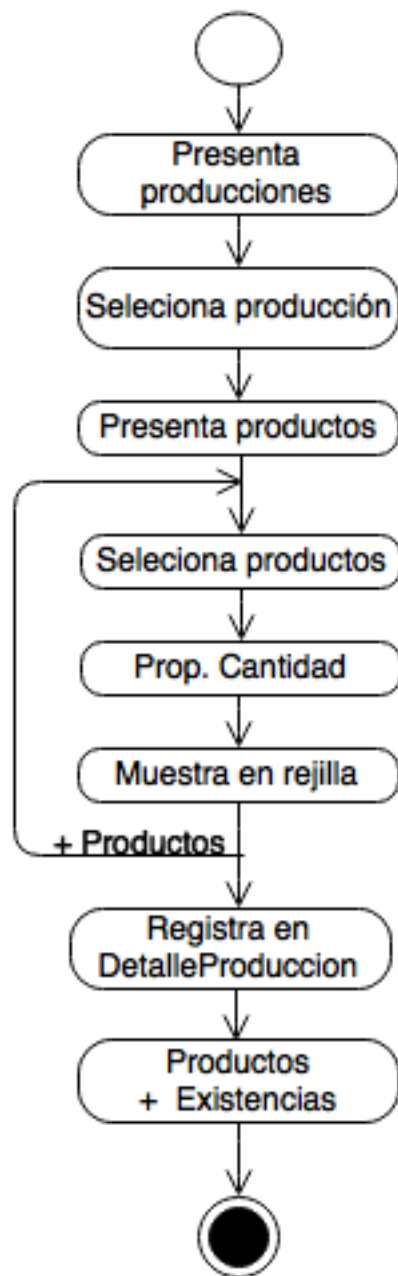


Figura 48. Diagrama de estados: Registrar productos terminados.

4.1.39 Diagrama de colaboración de registrar productos terminados

En la figura No. 49 se muestra el diagrama de colaboración correspondiente.

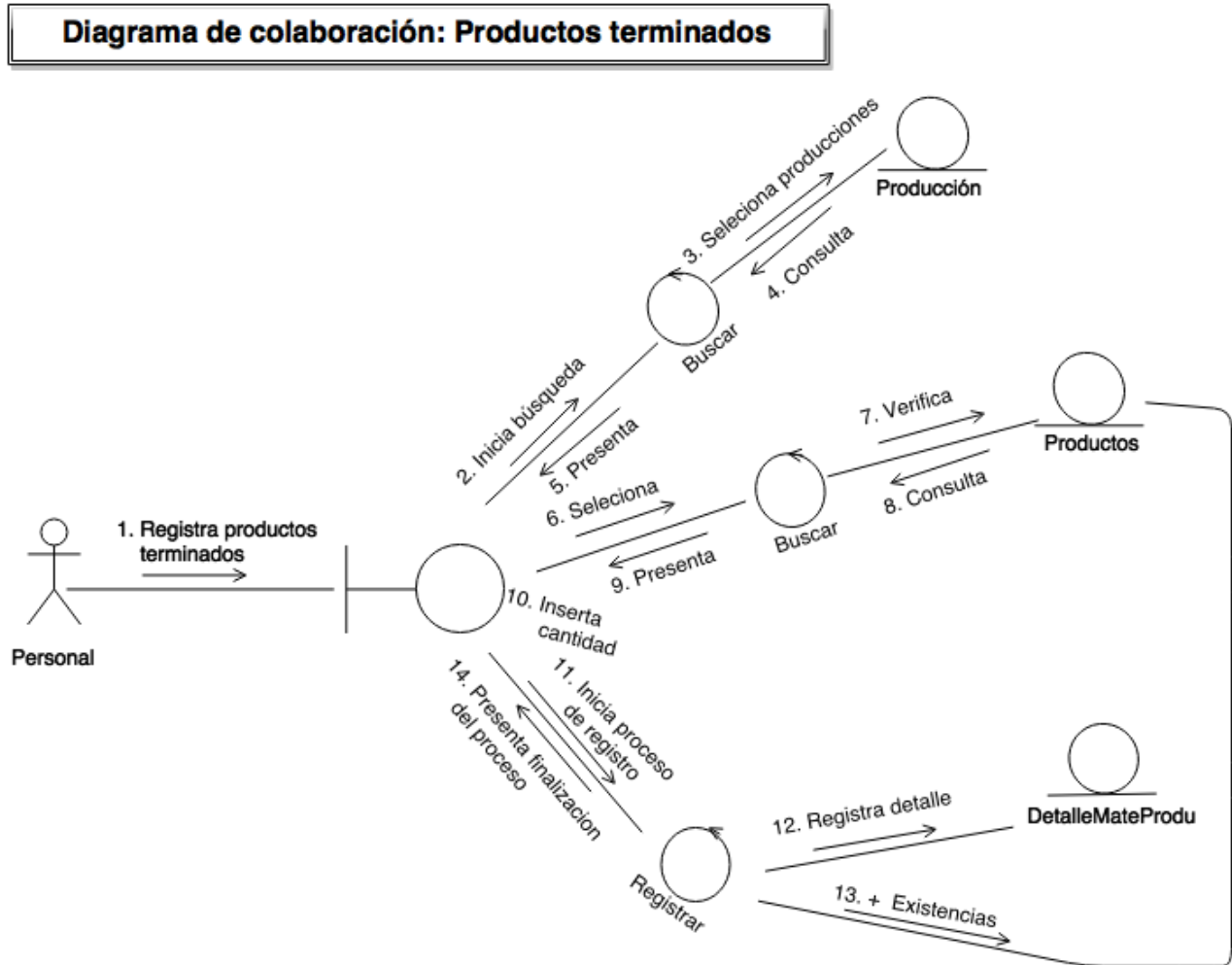


Figura 49. Diagrama de colaboración: Registrar productos terminados.

4.1.40 Diagrama de caso de uso expandido de registrar productos terminados

Nombre del caso: Registrar productos terminados.

Actores: Personal.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar productos terminados, busca la producción correspondiente, posteriormente busca el producto, lo selecciona e introduce la cantidad producida de dicho producto. Finalmente al grabar incrementa automáticamente las existencias en Productos.

En la figura No. 50 se muestra el diagrama de caso de uso expandido correspondiente.

Diagrama de caso de uso expandido: Productos terminados

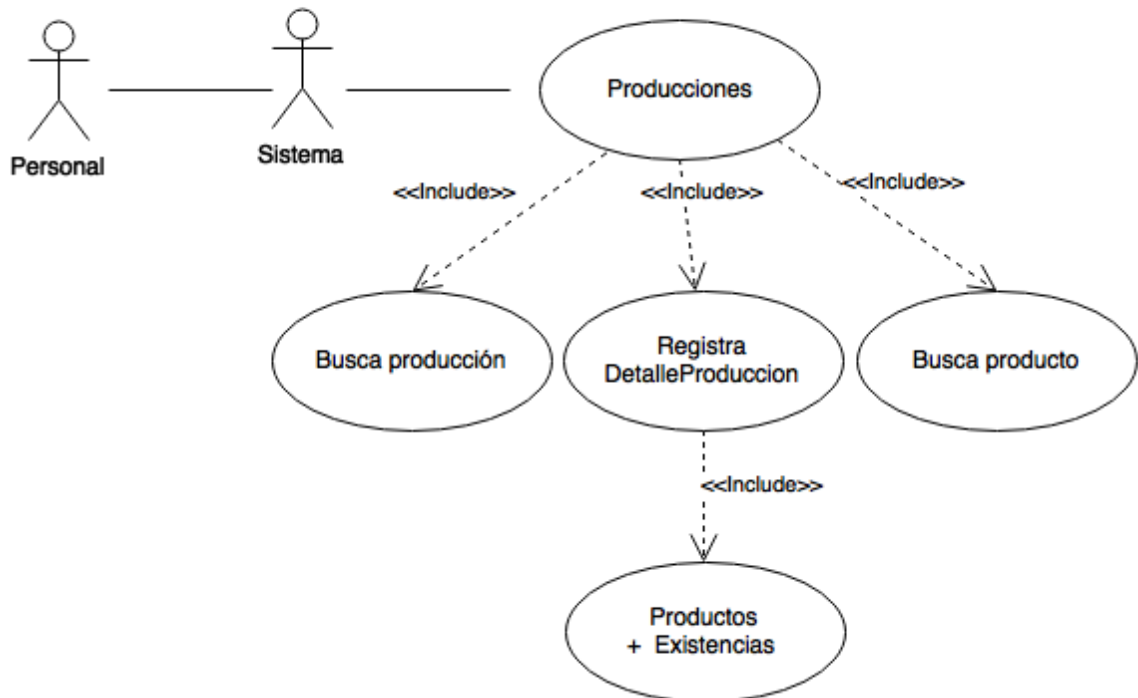


Figura 50. Diagrama de caso de uso expandido: Registrar productos terminados.

4.1.41 Diagrama de flujo de datos de registrar baja de personal

En la figura No. 51 se muestra el diagrama de flujo de datos correspondiente.

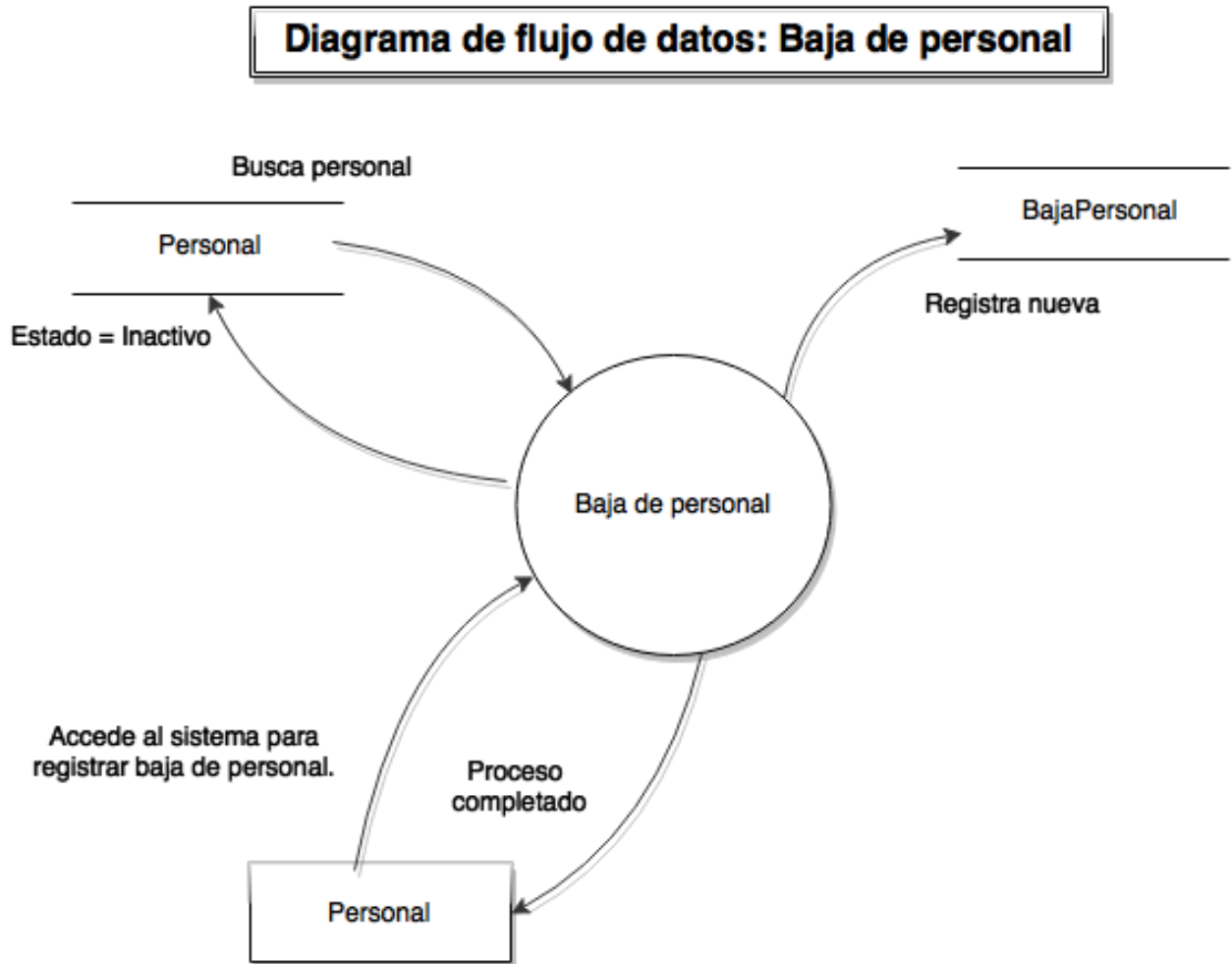


Figura 51. Diagrama de flujo de datos: Registrar baja de personal.

4.1.42 Diagrama de estados de registrar baja de personal

Flujo de eventos: El personal de la empresa selecciona en pantalla el nombre del personal a dar de baja y visualiza sus datos. Posteriormente describe el motivo por el cual se va a dar de baja y finalmente graba el registro.

Precondición: El personal (Gerente) de la empresa solicita una baja de personal.

Caminos alternos: No tiene.

Postcondición: Termina cuando el personal graba la baja, generando el registro correspondiente y cambiando automáticamente el estado del personal a inactivo. (Ver figura No. 52).

Diagrama de estados: Baja de personal

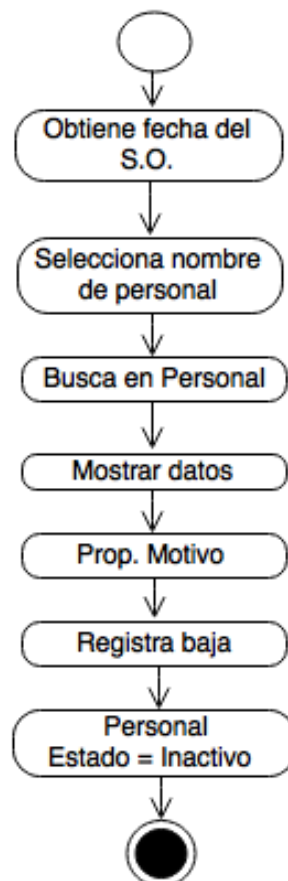


Figura 52. Diagrama de estados: Registrar baja de personal.

4.1.43 Diagrama de colaboración de registrar baja de personal

En la figura No. 53 se muestra el diagrama de colaboración correspondiente.

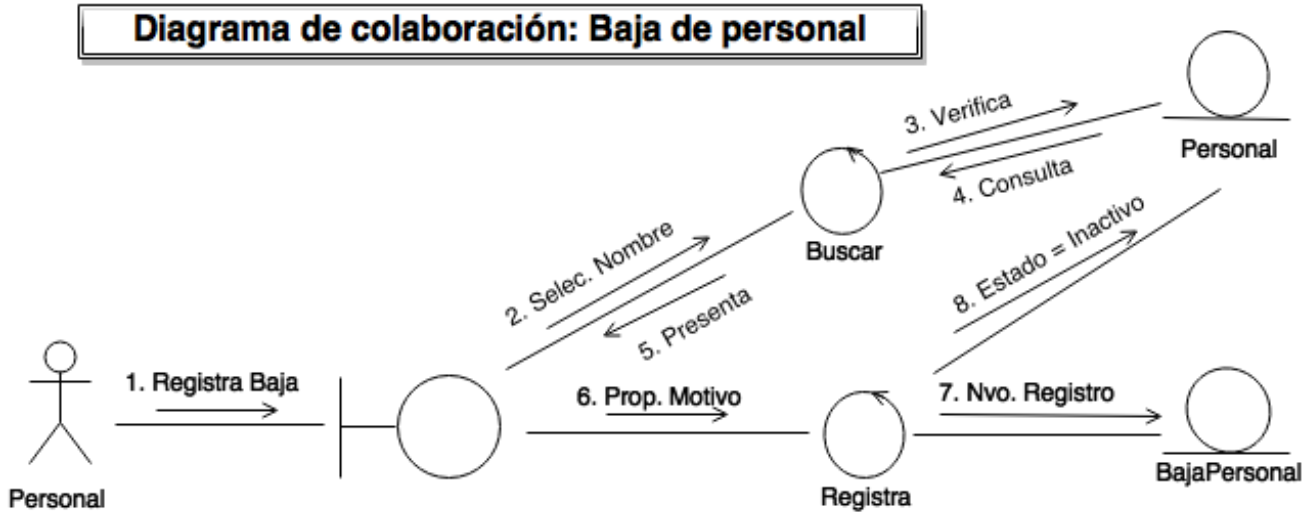


Figura 53. Diagrama de colaboración: Registrar baja de personal.

4.1.44 Diagrama de caso de uso expandido de registrar baja de personal

Nombre del caso: Registrar baja de personal

Actores: Personal.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar una baja de personal, busca el nombre del personal al cual se le dará de baja y lo selecciona. Finalmente graba el registro, actualizando el estado del personal a inactivo automáticamente.

En la figura No. 54 se muestra el diagrama de caso de uso expandido correspondiente.

Diagrama de caso de uso expandido: Baja de personal

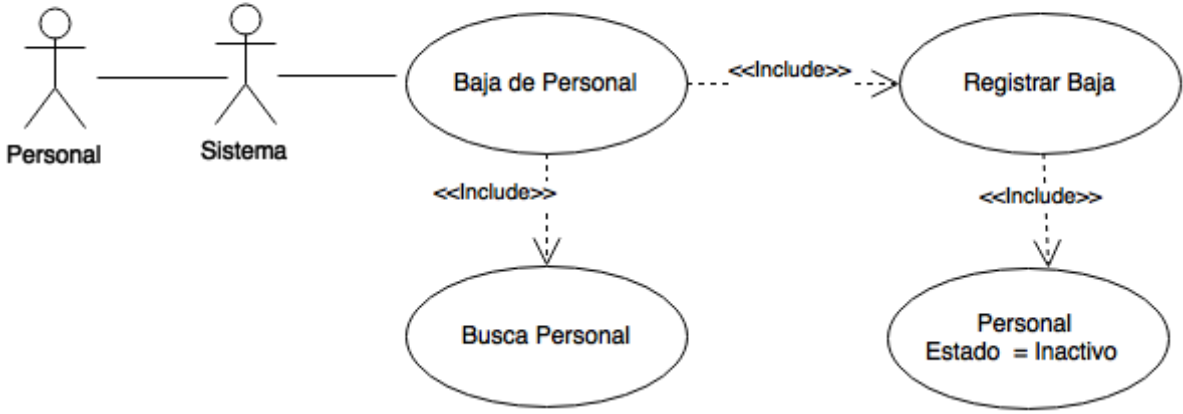


Figura 54. Diagrama de caso de uso expandido: Registrar baja de personal.

4.1.45 Diagrama de flujo de datos de reactivar personal

En la figura No. 55 se muestra el diagrama de flujo de datos correspondiente.

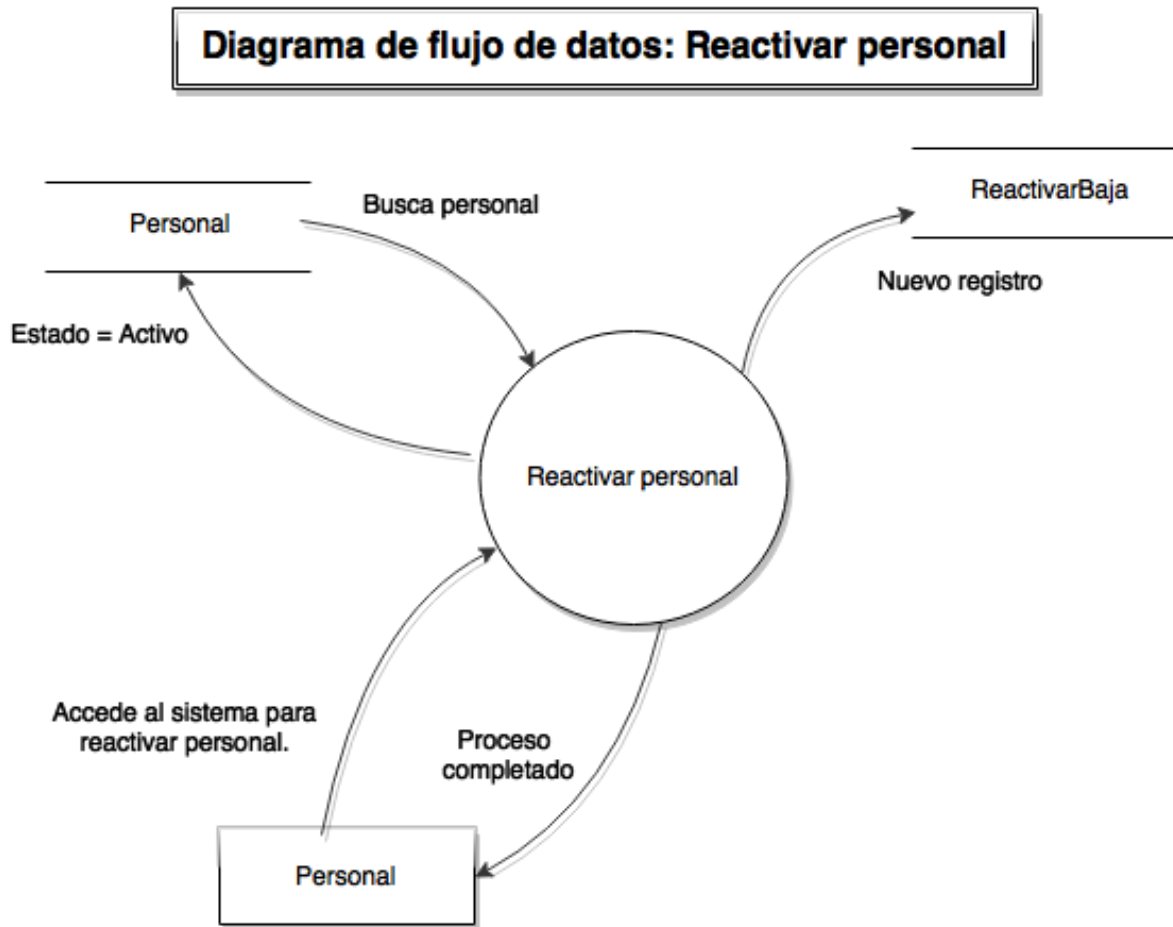


Figura 55. Diagrama de flujo de datos: Reactivar personal.

4.1.46 Diagrama de estados de reactivar personal

Flujo de eventos: El personal de la empresa busca y selecciona en pantalla el nombre del personal a reactivar, visualiza sus datos y posteriormente graba el registro.

Precondición: El personal (Gerente) de la empresa solicita una reactivación de personal.

Caminos alternos: No tiene.

Postcondición: Termina cuando el personal graba la reactivación, generando el registro correspondiente y cambiando automáticamente el estado del personal en activo. (Ver figura No. 56).

Diagrama de estados: Reactivar personal

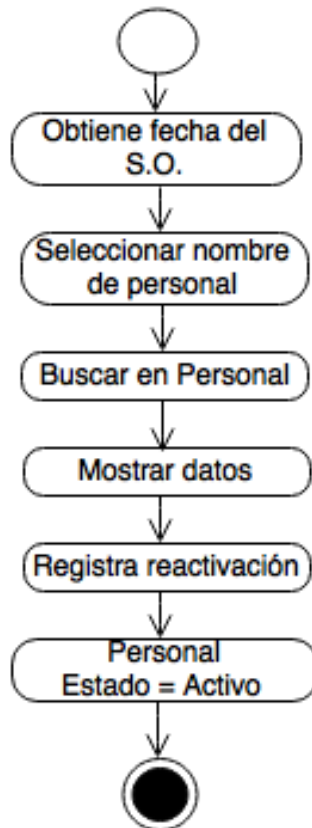


Figura 56. Diagrama de estados: Reactivar personal.

4.1.47 Diagrama de colaboración de reactivar personal

En la figura No. 57 se muestra el diagrama de colaboración correspondiente.

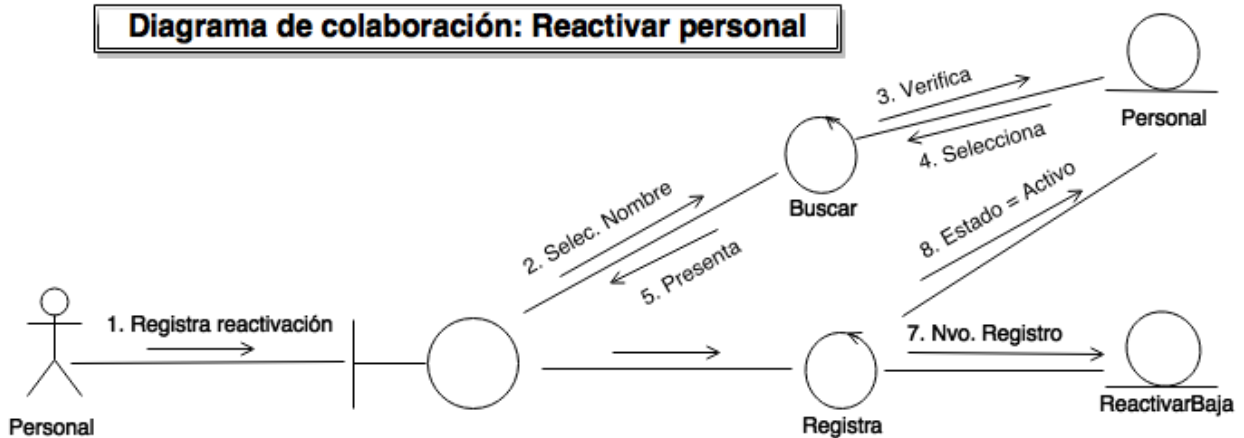


Figura 57. Diagrama de colaboración: Reactivar personal.

4.1.48 Diagrama de caso de uso expandido de reactivar personal

Nombre del caso: Reactivar personal.

Actores: Personal.

Nivel: Primario.

Descripción: El personal accede al sistema a registrar una reactivación de personal, busca el nombre del personal que se reactivará y lo selecciona. Finalmente graba el registro, actualizando automáticamente el estado del personal en activo.

En la figura No. 58 se muestra el diagrama de caso de uso expandido correspondiente.

Diagrama de caso de uso expandido: Reactivar personal

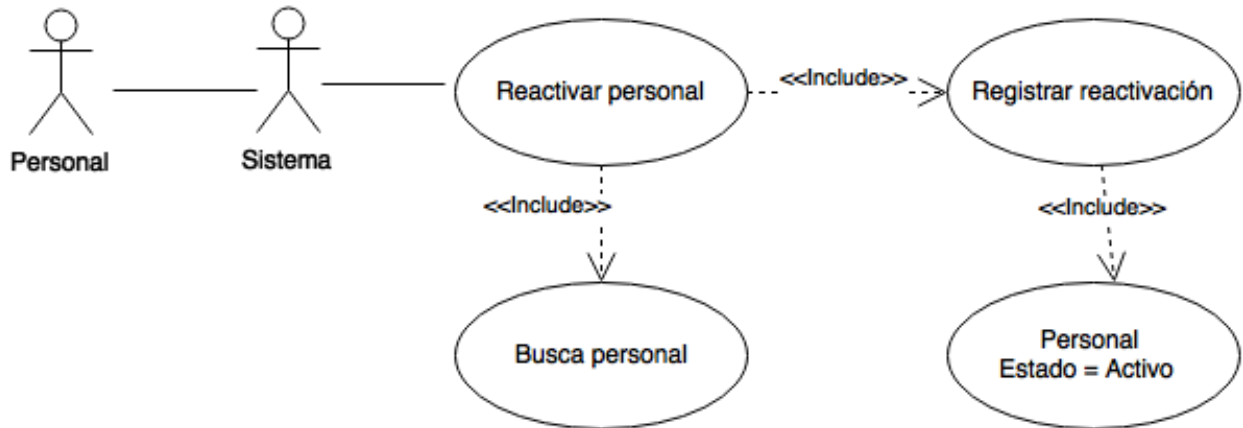


Figura 58. Diagrama de caso de uso expandido: Reactivar personal.

4.2. DISEÑO.

En el diseño se dio la continuidad con la secuencia de los diagramas UML, además se generaron los formularios para el usuario así como el diseño de la base de datos, reportes generales y estadísticos requeridos.

4.2.1 Normalización

Entidad: Personal

Campos: *IdPersonal, Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE, FechaIngreso.

Primera forma normal: La entidad Personal cumple con la primera forma normal (1FN) ya que sus atributos (IdPersonal, Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE y FechaIngreso) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdPersonal).

Segunda forma normal: La entidad Personal cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE y FechaIngreso) dependen de manera completa y no parcial del campo llave (IdPersonal).

Tercera Forma Normal: La entidad Personal cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE y FechaIngreso) dependen directamente y no de manera transitiva del campo llave (IdPersonal).

Entidad: BajaPersonal

Campos: *IdBaja, °IdPersonal, Fecha, Motivo.

Primera forma normal: La entidad BajaPersonal cumple con la primera forma normal (1FN) ya que sus atributos (IdBaja, IdPersonal, Fecha y Motivo) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdBaja).

Segunda forma normal: La entidad BajaPersonal cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no

primos (Fecha y Motivo) dependen de manera completa y no parcial del campo llave (IdBaja).

Tercera Forma Normal: La entidad BajaPersonal cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Fecha y Motivo) dependen directamente y no de manera transitiva del campo llave (IdBaja).

Entidad: ReactivarBaja

Campos: *IdReactivarBaja, °IdPersonal, Fecha.

Primera forma normal: La entidad ReactivarBaja cumple con la primera forma normal (1FN) ya que sus atributos (IdReactivarBaja, IdPersonal y Fecha) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdReactivarBaja).

Segunda forma normal: La entidad ReactivarBaja cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Fecha) dependen de manera completa y no parcial del campo llave (IdReactivarBaja).

Tercera Forma Normal: La entidad ReactivarBaja cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo (Fecha) dependen directamente y no de manera transitiva del campo llave (IdReactivarBaja).

Entidad: Proveedores

Campos: *IdProveedor, Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE, Saldo.

Primera forma normal: La entidad Proveedores cumple con la primera forma normal (1FN) ya que sus atributos (IdProveedor, Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE y Saldo) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProveedor).

Segunda forma normal: La entidad Proveedores cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no

primos (Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE y Saldo) dependen de manera completa y no parcial del campo llave (IdProveedor).

Tercera Forma Normal: La entidad Proveedores cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE y Saldo) dependen directamente y no de manera transitiva del campo llave (IdProveedor).

Entidad: Pagos

Campos: *IdPago, °FolioCompra, Fecha, Importe.

Primera forma normal: La entidad Pagos cumple con la primera forma normal (1FN) ya que sus atributos (IdPago, FolioCompra, Fecha e Importe) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdPago).

Segunda forma normal: La entidad Pagos cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Fecha e Importe) dependen de manera completa y no parcial del campo llave (IdPago).

Tercera Forma Normal: La entidad Pagos cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Fecha e Importe) dependen directamente y no de manera transitiva del campo llave (IdPago).

Entidad: Compras

Campos: *FolioCompra, °IdProveedor, Fecha, Subtotal, Iva, Condicion, FechaVencimiento, Abonos.

Primera forma normal: La entidad Compras cumple con la primera forma normal (1FN) ya que sus atributos (FolioCompra, IdProveedor, Fecha, Subtotal, Iva, Condicion, FechaVencimiento y Abonos) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (FolioCompra).

Segunda forma normal: La entidad Compras cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos

(Fecha, Subtotal, Iva, Condicion, FechaVencimiento y Abonos) dependen de manera completa y no parcial del campo llave (FolioCompra).

Tercera Forma Normal: La entidad Compras cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Fecha, Subtotal, Iva, Condicion, FechaVencimiento y Abonos) dependen directamente y no de manera transitiva del campo llave (FolioCompra).

Entidad: DetalleCompra

Campos: *FolioCompra, *IdMaterial, Cantidad, Costo.

Primera forma normal: La entidad DetalleCompra cumple con la primera forma normal (1FN) ya que sus atributos (FolioCompra, IdMaterial, Cantidad y Costo) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (FolioCompra + IdMaterial).

Segunda forma normal: La entidad DetalleCompra cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Cantidad y Costo) dependen de manera completa y no parcial del campo llave (FolioCompra + IdMaterial).

Tercera Forma Normal: La entidad DetalleCompra cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Cantidad y Costo) dependen directamente y no de manera transitiva del campo llave (FolioCompra + IdMaterial).

Entidad: Unidades

Campos: *IdUnidad, Descripcion.

Primera forma normal: La entidad Unidades cumple con la primera forma normal (1FN) ya que sus atributos (IdUnidad y Descripcion) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdUnidad).

Segunda forma normal: La entidad Unidades cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Descripcion) dependen de manera completa y no parcial del campo llave (IdUnidad).

Tercera Forma Normal: La entidad Unidades cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo (Descripcion) dependen directamente y no de manera transitiva del campo llave (IdUnidad).

Entidad: Materiales

Campos: *IdMaterial, °IdUnidad, Nombre, Costo, Stock, Existencias.

Primera forma normal: La entidad Materiales cumple con la primera forma normal (1FN) ya que sus atributos (IdMaterial, IdUnidad, Nombre, Costo, Stock y Existencias) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdMaterial).

Segunda forma normal: La entidad Materiales cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Nombre, Costo, Stock y Existencias) dependen de manera completa y no parcial del campo llave (IdMaterial).

Tercera Forma Normal: La entidad Materiales cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Nombre, Costo, Stock y Existencias) dependen directamente y no de manera transitiva del campo llave (IdMaterial).

Entidad: Pedidos

Campos: *IdPedido, °IdCliente, °IdPersonal, OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino, FechaEntrega.

Primera forma normal: La entidad Pedidos cumple con la primera forma normal (1FN) ya que sus atributos (IdPedido, IdCliente, IdPersonal, OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino y FechaEntrega) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdPedido).

Segunda forma normal: La entidad Pedidos cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino y

FechaEntrega) dependen de manera completa y no parcial del campo llave (IdPedido).

Tercera Forma Normal: La entidad Pedidos cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino y FechaEntrega) dependen directamente y no de manera transitiva del campo llave (IdPedido).

Entidad: DetallePedido

Campos: *IdPedido, *IdProducto, Cantidad, Precio.

Primera forma normal: La entidad DetallePedido cumple con la primera forma normal (1FN) ya que sus atributos (IdPedido, IdProducto, Cantidad y Precio) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdPedido + IdProducto).

Segunda forma normal: La entidad DetallePedido cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Cantidad y Precio) dependen de manera completa y no parcial del campo llave (IdPedido + IdProducto).

Tercera Forma Normal: La entidad DetallePedido cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Cantidad y Precio) dependen directamente y no de manera transitiva del campo llave (IdPedido + IdProducto).

Entidad: Produccion

Campos: *IdProduccion, FechaInicio, FechaFinal, Descripcion.

Primera forma normal: La entidad Produccion cumple con la primera forma normal (1FN) ya que sus atributos (IdProduccion, FechaInicio, FechaFinal y Descripcion) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProduccion).

Segunda forma normal: La entidad Produccion cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no

primos (FechaInicio, FechaFinal y Descripcion) dependen de manera completa y no parcial del campo llave (IdProduccion).

Tercera Forma Normal: La entidad Produccion cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (FechaInicio, FechaFinal y Descripcion) dependen directamente y no de manera transitiva del campo llave (IdProduccion).

Entidad: DetalleProduccion

Campos: *IdProduccion, *IdProducto, Cantidad.

Primera forma normal: La entidad DetalleProduccion cumple con la primera forma normal (1FN) ya que sus atributos (IdProduccion, IdProducto y Cantidad) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProduccion + IdProducto).

Segunda forma normal: La entidad DetalleProduccion cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Cantidad) dependen de manera completa y no parcial del campo llave (IdProduccion + IdProducto).

Tercera Forma Normal: La entidad DetalleProduccion cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo (Cantidad) dependen directamente y no de manera transitiva del campo llave (IdProduccion + IdProducto).

Entidad: DetalleMateProdu

Campos: *IdProduccion, *IdMaterial, Cantidad.

Primera forma normal: La entidad DetalleMateProdu cumple con la primera forma normal (1FN) ya que sus atributos (IdProduccion, IdMaterial y Cantidad) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProduccion + IdMaterial).

Segunda forma normal: La entidad DetalleMateProdu cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Cantidad) dependen de manera completa y no parcial del campo llave (IdProduccion + IdMaterial).

Tercera Forma Normal: La entidad DetalleMateProdu cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo (Cantidad) dependen directamente y no de manera transitiva del campo llave (IdProduccion + IdMaterial).

Entidad: DetalleSobrantes

Campos: *IdProduccion, *IdMaterial, Cantidad.

Primera forma normal: La entidad DetalleSobrantes cumple con la primera forma normal (1FN) ya que sus atributos (IdProduccion, IdMaterial y Cantidad) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProduccion + IdMaterial).

Segunda forma normal: La entidad DetalleSobrantes cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Cantidad) dependen de manera completa y no parcial del campo llave (IdProduccion + IdMaterial).

Tercera Forma Normal: La entidad DetalleSobrantes cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo (Cantidad) dependen directamente y no de manera transitiva del campo llave (IdProduccion + IdMaterial).

Entidad: Clasificacion

Campos: *IdClasificacion, Descripcion.

Primera forma normal: La entidad Clasificacion cumple con la primera forma normal (1FN) ya que sus atributos (IdClasificacion y Descripcion) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdClasificacion).

Segunda forma normal: La entidad Clasificacion cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y su atributo no primo (Descripcion) dependen de manera completa y no parcial del campo llave (IdClasificacion).

Tercera Forma Normal: La entidad Clasificacion cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y su atributo no primo

(Descripcion) dependen directamente y no de manera transitiva del campo llave (IdClasificacion).

Entidad: Productos

Campos: *IdProducto, °IdClasificacion, Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias, Stock.

Primera forma normal: La entidad Productos cumple con la primera forma normal (1FN) ya que sus atributos (IdProducto, IdClasificacion, Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias y Stock) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdProducto).

Segunda forma normal: La entidad Productos cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias y Stock) dependen de manera completa y no parcial del campo llave (IdProducto).

Tercera Forma Normal: La entidad Productos cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias y Stock) dependen directamente y no de manera transitiva del campo llave (IdProducto).

Entidad: Clientes

Campos: *IdCliente, Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP, Saldo.

Primera forma normal: La entidad Clientes cumple con la primera forma normal (1FN) ya que sus atributos (IdCliente, Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP y Saldo) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdCliente).

Segunda forma normal: La entidad Clientes cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP y Saldo) dependen de manera completa y no parcial del campo llave (IdCliente).

Tercera Forma Normal: La entidad Clientes cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos

(Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP y Saldo) dependen directamente y no de manera transitiva del campo llave (IdCliente).

Entidad: Ventas

Campos: *FolioVenta, °IdCliente, °IdPersonal, Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento, Abonos.

Primera forma normal: La entidad Ventas cumple con la primera forma normal (1FN) ya que sus atributos (FolioVenta, IdCliente, IdPersonal, Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento y Abonos) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (FolioVenta).

Segunda forma normal: La entidad Ventas cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento y Abonos) dependen de manera completa y no parcial del campo llave (FolioVenta).

Tercera Forma Normal: La entidad Ventas cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento y Abonos) dependen directamente y no de manera transitiva del campo llave (FolioVenta).

Entidad: DetalleVenta

Campos: *FolioVenta, *IdProducto, Cantidad, Precio.

Primera forma normal: La entidad DetalleVenta cumple con la primera forma normal (1FN) ya que sus atributos (FolioVenta, IdProducto, Cantidad y Precio) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (FolioVenta + IdProducto).

Segunda forma normal: La entidad DetalleVenta cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Cantidad y Precio) dependen de manera completa y no parcial del campo llave (FolioVenta + IdProducto).

Tercera Forma Normal: La entidad DetalleVenta cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no

primos (Cantidad y Precio) dependen directamente y no de manera transitiva del campo llave (FolioVenta + IdProducto).

Entidad: Abonos

Campos: *IdAbono, °FolioVenta, Fecha, Importe.

Primera forma normal: La entidad Abonos cumple con la primera forma normal (1FN) ya que sus atributos (IdAbono, FolioVenta, Fecha e Importe) contienen solo valores atómicos, no existen grupos repetidos y cuenta con un campo llave (IdAbono).

Segunda forma normal: La entidad Abonos cumple con la segunda forma normal (2FN), ya que cumple con la primera forma normal y todos sus atributos no primos (Fecha e Importe) dependen de manera completa y no parcial del campo llave (IdAbono).

Tercera Forma Normal: La entidad Abonos cumple con la tercera forma normal (3FN), ya que cumple con la segunda forma normal y todos sus atributos no primos (Fecha e Importe) dependen directamente y no de manera transitiva del campo llave (IdAbono).

CAPÍTULO 5. RESULTADOS.

5.1 Diagrama de clases

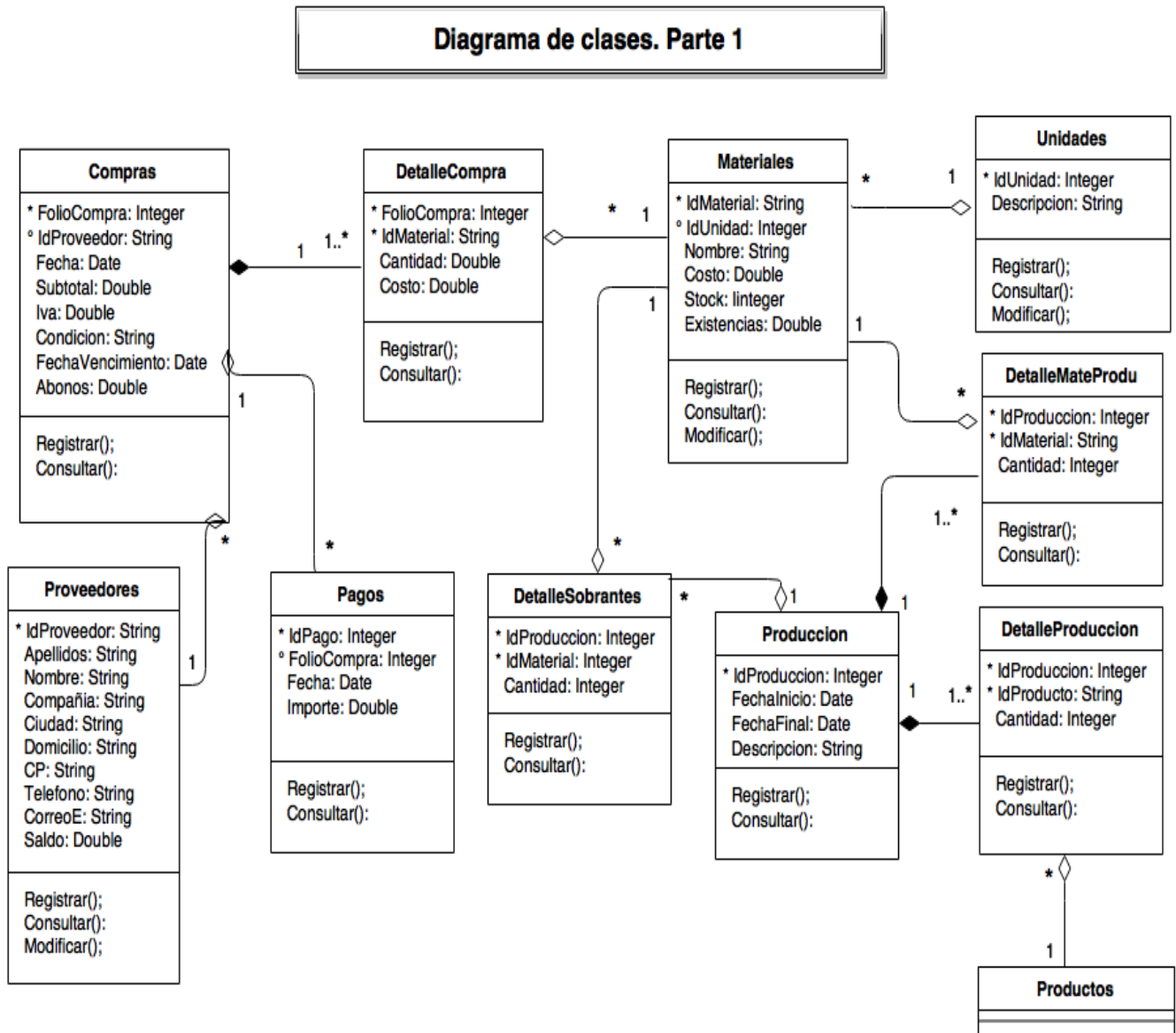


Figura 59. Diagrama de clases. Parte 1.

En la figura No. 59 se muestra la primera parte del diagrama de clases.

En la figura No. 60 se muestra la segunda parte del diagrama de clases.

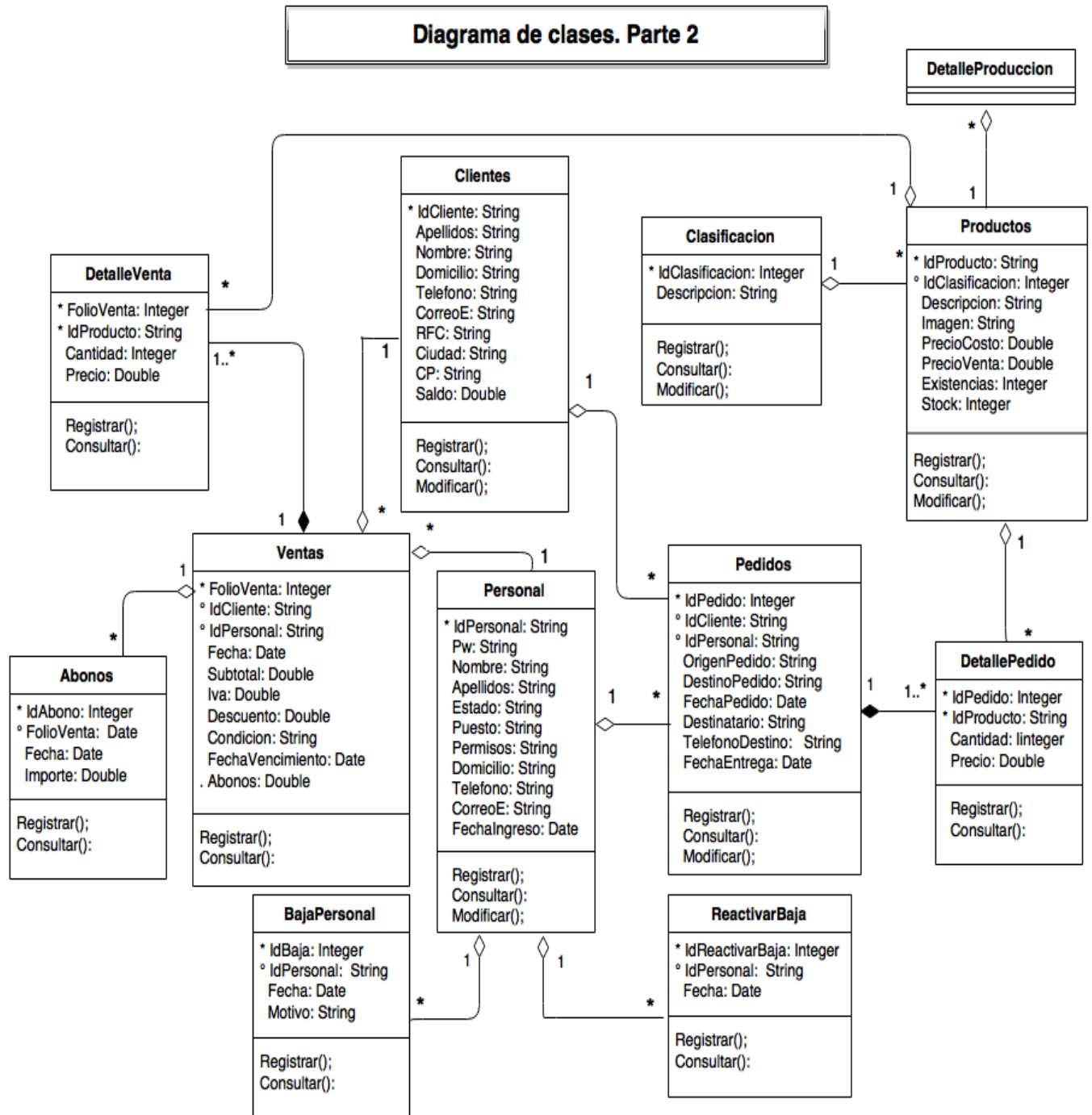


Figura 60. Diagrama de clases. Parte 2.

5.2 Diccionario de datos

Nombre de la entidad: Personal.

Objetivo: Lleva la administración de los empleados activos e inactivos de la empresa.

Número de campos: 11 (IdPersonal, Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE, FechaIngreso).

Campo llave: IdPersonal.

Campos no primos: 10 (Pw, Nombre, Apellidos, Estado, Puesto, Permisos, Domicilio, Telefono, CorreoE, FechaIngreso).

En la tabla No. 3 se muestra la tabla de Personal.

Tabla 3. Tabla de Personal.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdPersonal	varchar (Cadena)	18	Guarda el Identificador del personal.	Conjunto de dígitos válidos para formar el Id.
2	Pw	varchar (Cadena)	30	Guarda el password del personal.	Conjunto de caracteres válidos.
3	Nombre	varchar (Cadena)	40	Guarda el nombre del empleado.	Conjunto de caracteres válidos para representar el nombre.
4	Apellidos	varchar (Cadena)	40	Guarda el apellido de empleado.	Conjunto de caracteres válidos para representar los apellidos.
5	Estado	varchar (Cadena)	18	Guarda el estado actual de empleado.	Activo, Inactivo.
6	Puesto	varchar (Cadena)	50	Guarda el puesto del empleado.	Conjunto de caracteres válidos para representar el puesto.

7	Permisos	varchar (Cadena)	500	Guarda los permisos correspondientes al puesto del personal.	Conjunto de caracteres válidos para representar los permisos.
8	Domicilio	varchar (Cadena)	100	Guarda el domicilio del empleado.	Conjunto de caracteres válidos para representar el domicilio.
9	Telefono	varchar (Cadena)	20	Guarda el teléfono del empleado.	Conjunto de dígitos válidos para representar el número telefónico incluyendo (), /, -.
10	CorreoE	varchar (Cadena)	60	Guarda el correo electrónico del empleado.	Conjunto de caracteres válidos para representar el correo electrónico.
11	FechaIngreso	date	--	Guarda la fecha de ingreso del empleado.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).

Fuente: Propia.

Nombre de la entidad: BajaPersonal.

Objetivo: Controlar el estado del personal dentro de la empresa.

Número de campos: 4 (IdBaja, IdPersonal, Fecha, Motivo).

Campo llave: IdBaja.

Campos no primos: 2 (Fecha, Motivo)

En la tabla No. 4 se muestra la tabla de BajaPersonal.

Tabla 4. Tabla de BajaPersonal.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdBaja	int	9	Guarda el Identificador (Id) de la baja.	Conjunto de dígitos válidos para formar el Id.

2	IdPersonal	varchar (Cadena)	18	Guarda el Identificador (Id) del personal.	Conjunto de dígitos válidos para formar el Id.
3	Fecha	date	--	Guarda la fecha en que se efectuó la baja.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
4	Motivo	varchar (Cadena)	50	Guarda el motivo por el cual se hizo la baja.	Conjunto de caracteres válidos para representar la causa.

Fuente: Propia.

Nombre de la entidad: ReactivarBaja.

Objetivo: Llevar el control del estado del personal ejerciendo en la empresa.

Número de campos: 3 (IdReactivarBaja, IdPersonal, Fecha).

Campo llave: IdReactivarBaja.

Campos no primos: 1 (Fecha).

En la tabla No. 5 se muestra la tabla de ReactivarBaja.

Tabla 5. Tabla de ReactivarBaja.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdReactivarBaja	int	9	Guarda el Identificador (Id) de la reactivación.	Conjunto de dígitos válidos para formar el Id.
2	IdPersonal	varchar (Cadena)	18	Guarda el Identificador (Id) del empleado.	Conjunto de dígitos válidos para formar el Id.
3	Fecha	date	--	Guarda la fecha en que se realiza la reactivación.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).

Fuente: Propia.

Nombre de la entidad: Proveedores.

Objetivo: Llevar el control de los proveedores que surten la empresa.

Número de campos: 10 (IdProveedor, Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE, Saldo).

Campo llave: IdProveedor.

Campos no primos: 9 (Apellidos, Nombre, Compañía, Ciudad, Domicilio, CP, Telefono, CorreoE, Saldo).

En la tabla No. 6 se muestra la tabla de Proveedores.

Tabla 6. Tabla de Proveedores.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProveeder	varchar (Cadena)	18	Guarda el Identificador (Id) del proveedor.	Conjunto de dígitos válidos para formar el Id.
2	Apellidos	varchar (Cadena)	40	Guarda el apellido del proveedor.	Conjunto de caracteres válidos para representar los apellidos.
3	Nombre	varchar (Cadena)	40	Guarda el nombre del proveedor.	Conjunto de caracteres válidos para representar el nombre.
4	Compania	varchar (Cadena)	50	Guarda la compañía del proveedor.	Conjunto de caracteres válidos para representar la compañía.
5	Ciudad	varchar (Cadena)	30	Guarda la ciudad del proveedor.	Conjunto de caracteres válidos para representar la ciudad.
6	Domicilio	varchar (Cadena)	100	Guarda el domicilio del proveedor.	Conjunto de caracteres válidos para representar el domicilio.
7	CP	varchar (Cadena)	15	Guarda el código postal del proveedor.	Conjunto de dígitos válidos para representar el CP.
8	Telefono	varchar (Cadena)	20	Guarda el teléfono del proveedor.	Conjunto de dígitos válidos para representar el número telefónico incluyendo (), /, -.
9	CorreoE	varchar (Cadena)	50	Guarda el correo electrónico del proveedor.	Conjunto de caracteres válidos para representar el correo.
10	Saldo	decimal	(12,2)	Guarda el saldo del proveedor.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Pagos.

Objetivo: Llevar el control de los pagos a proveedores.

Número de campos: 4 (IdPago, FolioCompra, Fecha, Importe).

Campo llave: IdPago.

Campos no primos: 2 (Fecha, Importe).

En la tabla No. 7 se muestra la tabla de Pagos.

Tabla 7. Tabla de Pagos.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdPago	int	9	Guarda el Identificador (Id) del pago.	Conjunto de dígitos válidos para formar el Id.
2	FolioCompra	int	9	Guarda el folio de la compra.	Conjunto de dígitos válidos para formar el folio.
3	Fecha	date	--	Guarda la fecha en que se efectuó el pago.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
4	Importe	decimal	(12,2)	Guarda el importe del pago.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Compras.

Objetivo: Llevar el control de las compras realizadas a los proveedores que maneja la empresa.

Número de campos: 8 (FolioCompra, IdProveedor, Fecha, Subtotal, Iva, Condicion, FechaVencimiento, Abonos).

Campo llave: FolioCompra.

Campos no primos: 6 (Fecha, Subtotal, Iva, Condicion, FechaVencimiento, Abonos).

En la tabla No. 8 se muestra la tabla de Compras.

Tabla 8. Tabla de Compras.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	FolioCompra	int	9	Guarda el folio de la compra.	Conjunto de dígitos válidos para formar el folio.
2	IdProveedor	varchar (Cadena)	18	Guarda el Identificador (Id) del proveedor.	Conjunto de dígitos válidos para formar el Id.
3	Fecha	date	--	Guarda la fecha en que se efectuó la compra.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
4	Subtotal	decimal	(12,2)	Guarda el subtotal de la compra.	Conjunto de dígitos válidos.

5	IVA	decimal	(12,2)	Guarda el IVA en cuanto a la compra.	Conjunto de dígitos válidos.
6	Condicion	varchar (Cadena)	20	Guarda el tipo de compra.	Contado o Crédito.
7	FechaVencimiento	date	--	Guarda la fecha limite para liquidar la compra.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
8	Abonos	decimal	(12,2)	Guarda el acumulado de los abonos.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: DetalleCompra.

Objetivo: Llevar el control de materiales adquiridos en cada una de las compras.

Número de campos: 4 (FolioCompra, IdMaterial, Cantidad, Costo).

Campo llave: FolioCompra + IdMaterial.

Campos no primos: 2 (Cantidad, Costo).

En la tabla No. 9 se muestra la tabla de DetalleCompra.

Tabla 9. Tabla de DetalleCompra.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	FolioCompra	int	9	Guarda el Identificador (Id) de la compra.	Conjunto de dígitos válidos para formar el Id.
2	IdMaterial	varchar (Cadena)	18	Guarda el Identificador (Id) del material.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	real	9	Guarda la cantidad de material.	Conjunto de dígitos válidos.
4	Costo	decimal	(12,2)	Guarda el costo de compra del material.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Unidades.

Objetivo: Llevar el control de las unidades de los materiales.

Número de campos: 2 (IdUnidad, Descripcion).

Campo llave: IdUnidad.

Campos no primos: 1 (Descripcion).

En la tabla No. 10 se muestra la tabla de Unidades.

Tabla 10. Tabla de Unidades.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdUnidad	int	9	Guarda el Identificador (Id) de la unidad.	Conjunto de dígitos válidos para formar el Id.
2	Descripcion	varchar (Cadena)	50	Guarda el nombre de la unidad.	Conjunto de caracteres válidos para representar la unidad.

Fuente: Propia.

Nombre de la entidad: Materiales.

Objetivo: Llevar la administración de los materiales disponibles para las producciones.

Número de campos: 6 (IdMaterial, IdUnidad, Nombre, Costo, Stock, Existencias).

Campo llave: IdMaterial.

Campos no primos: 4 (Nombre, Costo, Stock, Existencias).

En la tabla No. 11 se muestra la tabla de Materiales.

Tabla 11. Tabla de Materiales.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdMaterial	varchar (Cadena)	18	Guarda el Identificador (Id) del material.	Conjunto de dígitos válidos para formar el Id.
2	IdUnidad	int	9	Guarda el Id de la unidad de medida.	Conjunto de dígitos válidos para formar el Id.
3	Descripcion	Varchar (Cadena)	100	Guarda la descripción del material.	Conjunto de caracteres válidos para representar la descripción del material.
4	Costo	decimal	(12,2)	Guarda el costo del material	Conjunto de dígitos válidos.
5	Stock	int	9	Guarda la cantidad mínima para surtir.	Conjunto de dígitos válidos.
6	Existencias	real	9	Guarda las existencias de material	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Pedidos.

Objetivo: Llevar el control de pedidos realizados por los clientes.

Número de campos: 9 (IdPedido, IdCliente, IdPersonal, OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino, FechaEntrega).

Campo llave: IdPedido.

Campos no primos: 6 (OrigenPedido, DestinoPedido, FechaPedido, Destinatario, TelefonoDestino, FechaEntrega).

En la tabla No. 12 se muestra la tabla de Pedidos.

Tabla 12. Tabla de Pedidos.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdPedido	int	9	Guarda el Identificador (Id) del pedido.	Conjunto de dígitos válidos para formar el Id.
2	IdCliente	varchar (Cadena)	18	Guarda el identificador (Id) del cliente.	Conjunto de dígitos válidos para formar el Id.
3	IdPersonal	varchar (Cadena)	18	Guarda el identificador (Id) del personal que atendió el pedido.	Conjunto de dígitos válidos para formar el Id.
4	OrigenPedido	varchar (Cadena)	100	Guarda el origen del pedido.	Conjunto de caracteres válidos para representar el origen del pedido.
5	DestinoPedido	varchar (Cadena)	100	Guarda el destino del pedido.	Conjunto de caracteres válidos para representar el destino del pedido.
6	FechaPedido	date	--	Guarda la fecha en que se realizó el pedido	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
7	Destinatario	varchar (Cadena)	100	Guarda el nombre de quién va a recibir el pedido.	Conjunto de caracteres válidos para representar el nombre del destinatario.
8	TelefonoDetino	varchar (Cadena)	20	Guarda el número telefónico del destinatario.	Conjunto de dígitos válidos para representar el número telefónico incluyendo (), /, -.
9	FechaEntrega	date	--	Guarda la fecha en que se entregar el pedido.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).

Fuente: Propia.

Nombre de la entidad: DetallePedido.

Objetivo: Llevar el control de la cantidad de productos bajo pedido.

Número de campos: 4 (IdPedido, IdProducto, Cantidad, Precio).

Campo llave: IdPedido + IdProducto.

Campos no primos: 2 (Cantidad, Precio).

En la tabla No. 13 se muestra la tabla de DetallePedido.

Tabla 13. Tabla de DetallePedido.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdPedido	int	9	Guarda el Identificador (Id) del pedido.	Conjunto de dígitos válidos para formar el Id.
2	IdProducto	varchar (Cadena)	50	Guarda el Identificador (Id) del producto.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	int	9	Guarda la cantidad de productos.	Conjunto de dígitos válidos.
4	Precio	decimal	(8,2)	Guarda el precio venta del producto.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Produccion.

Objetivo: Llevar el control de las producciones.

Número de campos: 4 (FolioCompra, IdMaterial, Cantidad, Costo).

Campo llave: IdProduccion.

Campos no primos: 2 (Cantidad, Costo).

En la tabla No. 14 se muestra la tabla de Produccion.

Tabla 14. Tabla de Produccion.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProduccion	Int	9	Guarda el Identificador (Id) de la producción.	Conjunto de dígitos válidos para formar el Id.
2	FechaInicio	date	--	Guarda la fecha de inicio de la producción.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
3	FechaFinal	date	--	Guarda la fecha final de la producción.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
4	Descripcion	varchar (Cadena)	100	Guarda una breve descripción de la producción.	Conjunto de caracteres válidos

					para representar la descripción.
--	--	--	--	--	----------------------------------

Fuente: Propia.

Nombre de la entidad: DetalleProduccion.

Objetivo: Llevar la contabilización de los productos fabricados por producción.

Número de campos: 3 (IdProduccion, IdProducto, Cantidad).

Campo llave: IdProduccion + IdProducto.

Campos no primos: 1 (Cantidad).

En la tabla No. 15 se muestra la tabla de DetalleProduccion.

Tabla 15. Tabla de DetalleProduccion.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProduccion	Int	9	Guarda el Identificador (Id) de producción.	Conjunto de dígitos válidos para forma el Id.
2	IdProducto	varchar (Cadena)	50	Guarda el Identificador (Id) del producto.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	real	9	Guarda la cantidad de productos.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: DetalleMateProdu.

Objetivo: Llevar el control de material utilizado para cada producción.

Número de campos: 3 (IdProduccion, IdMaterial, Cantidad).

Campo llave: IdProduccion + IdMaterial.

Campos no primos: 1 (Cantidad).

En la tabla No. 16 se muestra la tabla de DetalleMateProdu.

Tabla 16. Tabla de DetalleMateProdu.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProduccion	int	9	Guarda el Identificador (Id) de producción.	Conjunto de dígitos válidos para formar el Id.
2	IdMaterial	varchar (Cadena)	18	Guarda el Identificador (Id) del material.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	Real	9	Guarda la cantidad de material.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: DetalleSobrantes.

Objetivo: Llevar el control de los materiales sobrantes por cada producción.

Número de campos: 3 (IdProduccion, IdMaterial, Cantidad).

Campo llave: IdProduccion + IdMaterial.

Campos no primos: 1 (Cantidad).

En la tabla No. 17 se muestra la tabla de DetalleSobrantes.

Tabla 17. Tabla de DetalleSobrantes.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProduccion	int	9	Guarda el Identificador (Id) de producción.	Conjunto de dígitos válidos para formar el Id.
2	IdMaterial	varchar (Cadena)	10	Guarda el Identificador (Id) del material.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	int	9	Guarda la cantidad de material.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Clasificacion.

Objetivo: Llevar el control de clasificación de los diferentes productos para su fácil localización.

Número de campos: 2 (IdClasificacion, Descripcion).

Campo llave: IdClasificacion.

Campos no primos: 1 (Descripcion).

En la tabla No. 18 se muestra la tabla de Clasificacion.

Tabla 18. Tabla de Clasificacion.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdClasificacion	int	9	Guarda el Identificador (Id) de la clasificación.	Conjunto de dígitos válidos para forma el Id.
2	Descripcion	varchar (Cadena)	30	Guarda el nombre de la clasificación.	Conjunto de caracteres válidos para representar la categoría.

Fuente: Propia.

Nombre de la entidad: Productos.

Objetivo: Llevar el control de los productos que se fabrican dentro de la empresa.

Número de campos: 8 (IdProducto, IdClasificacion, Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias, Stock).

Campo llave: IdProducto.

Campos no primos: 6 (Descripcion, Imagen, PrecioCosto, PrecioVenta, Existencias, Stock).

En la tabla No. 19 se muestra la tabla de Productos.

Tabla 19. Tabla de Productos.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdProducto	varchar (Cadena)	50	Guarda el Identificador (Id) del producto.	Conjunto de dígitos válidos para formar el Id.
2	IdClasificacion	int	9	Guarda el Identificador (Id) de la clasificación.	Conjunto de dígitos válidos para formar el Id.
3	Descripcion	varchar (Cadena)	100	Guarda el nombre del producto.	Conjunto de caracteres válidos para representar la descripción.
4	Imagen	varchar (Cadena)	100	Guarda la ruta donde se encuentra almacenada la imagen	Conjunto de caracteres válidos para representar la ruta de imagen.
5	PrecioCosto	decimal	(12,2)	Guarda el precio costo del producto.	Conjunto de dígitos válidos.
6	PrecioVenta	decimal	(12,2)	Guarda el precio venta del producto.	Conjunto de dígitos válidos.
7	Existencias	Real	9	Guarda las existencias del producto.	Conjunto de dígitos válidos.
8	Stock	int	9	Guarda la cantidad mínima de productos.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Clientes.

Objetivo: Llevar el control de clientes a los que la empresa hace las ventas.

Número de campos: 10 (IdCliente, Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP, Saldo).

Campo llave: IdCliente.

Campos no primos: 9 (Apellidos, Nombre, Domicilio, Telefono, CorreoE, RFC, Ciudad, CP, Saldo).

En la tabla No. 20 se muestra la tabla de Clientes.

Tabla 20. Tabla de Clientes.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdCliente	varchar (Cadena)	18	Guarda el identificador (Id) del cliente.	Conjunto de dígitos válidos para formar el Id.
2	Nombre	varchar (Cadena)	40	Guarda el nombre del cliente.	Conjunto de caracteres válidos para representar el nombre.
3	Apellidos	varchar (Cadena)	40	Guarda el apellido del cliente.	Conjunto de caracteres válidos para representar los apellidos.
4	Domicilio	varchar (Cadena)	100	Guarda el domicilio del cliente.	Conjunto de caracteres válidos para representar el domicilio.
5	Telefono	varchar (Cadena)	20	Guarda el teléfono del cliente.	Conjunto de dígitos válidos para representar el número telefónico incluyendo (), /, -.
6	CorreoE	varchar (Cadena)	60	Guarda el correo electrónico del cliente.	Conjunto de caracteres válidos para representar el correo.
7	RFC	varchar (Cadena)	30	Guarda el RFC del cliente en caso de tener.	Conjunto de caracteres válidos para representar el RFC.
8	Ciudad	varchar (Cadena)	30	Guarda la ciudad de cliente.	Conjunto de caracteres válidos para representar la ciudad.
9	CP	varchar (Cadena)	15	Guarda el código postal de cliente.	Conjunto de dígitos válidos para representar el CP.
10	Saldo	decimal	(12,2)	Guarda el saldo del cliente.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: Abonos.

Objetivo: Llevar el control de abonos realizados por cada uno de los clientes de la empresa.

Número de campos: 4 (IdAbono, FolioVenta, Fecha, Importe).

Campo llave: IdAbono.

Campos no primos: 2 (Fecha, Importe).

En la tabla No. 21 se muestra la tabla de Abonos.

Tabla 21. Tabla de Abonos.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	IdAbono	int	9	Guarda el identificador del pago.	Conjunto de dígitos válidos para formar el Id.
2	FolioVenta	int	9	Guarda el identificador de la venta.	Conjunto de dígitos válidos para forma el Id.
3	Fecha	date	--	Almacena la fecha en que se realizó la venta.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
4	Importe	decimal	(12,2)	Guarda el importe del abono.	Conjunto de dígitos numéricos.

Fuente: Propia.

Nombre de la entidad: Ventas.

Objetivo: Llevar el control de las ventas realizadas a los clientes.

Número de campos: 10 (FolioVenta, IdCliente, IdPersonal, Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento, Abonos).

Campo llave: FolioVenta.

Campos no primos: 7 (Fecha, Subtotal, Iva, Descuento, Condicion, FechaVencimiento, Abonos).

En la tabla No. 22 se muestra la tabla de Ventas.

Tabla 22. Tabla de Ventas.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	FolioVenta	int	9	Guarda el Identificador (Id) de la venta.	Conjunto de dígitos válidos para formar el Id.
2	IdCliente	varchar (Cadena)	18	Guarda el Identificador (Id) del cliente.	Conjunto de dígitos válidos para formar el Id.
3	IdPersonal	varchar (Cadena)	18	Guarda el Identificador (Id) del empleado.	Conjunto de dígitos válidos para formar el Id.
4	Fecha	date	--	Guarda la fecha en que se realizó la venta.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
5	Subtotal	decimal	(12,2)	Guarda el subtotal de la venta.	Conjunto de dígitos válidos.
6	IVA	decimal	(12,2)	Guarda el IVA de la venta.	Conjunto de dígitos válidos.
7	Descuento	decimal	(12,2)	Guarda el descuento en caso de existir.	Conjunto de dígitos válidos.
8	Condicion	varchar (Cadena)	20	Guarda el tipo de venta.	Contado o Crédito.
9	FechaVencimiento	date	--	Guarda la fecha límite de vencimiento para liquidar la venta.	Conjunto de caracteres con formato de fecha (dd/mm/aaaa).
10	Abonos	decimal	(12,2)	Guarda el monto acumulado por los pagos del cliente.	Conjunto de dígitos válidos.

Fuente: Propia.

Nombre de la entidad: DetalleVenta.

Objetivo: Llevar el control de productos vendidos por venta.

Número de campos: 4 (FolioVenta, IdProducto, Cantidad, Precio).

Campo llave: FolioVenta + IdProducto.

Campos no primos: 2 (Cantidad, Precio).

En la tabla No. 23 se muestra la tabla de DetalleVenta.

Tabla 23. Tabla de DetalleVenta.

No.	Nombre	Tipo	Long.	Descripción	Dominio
1	FolioVenta	int	9	Guarda el Identificador (Id) de la venta.	Conjunto de dígitos válidos para formar el Id.
2	IdProducto	varchar (Cadena)	50	Guarda el identificador (Id) del producto.	Conjunto de dígitos válidos para formar el Id.
3	Cantidad	real	9	Guarda la cantidad de productos.	Conjunto de dígitos válidos.
4	Precio	decimal	(12,2)	Guarda el precio de venta.	Conjunto de dígitos válidos.

Fuente: Propia.

5.3 Diseño de interfaces

En la figura No. 61 se muestra el menú principal del sistema.

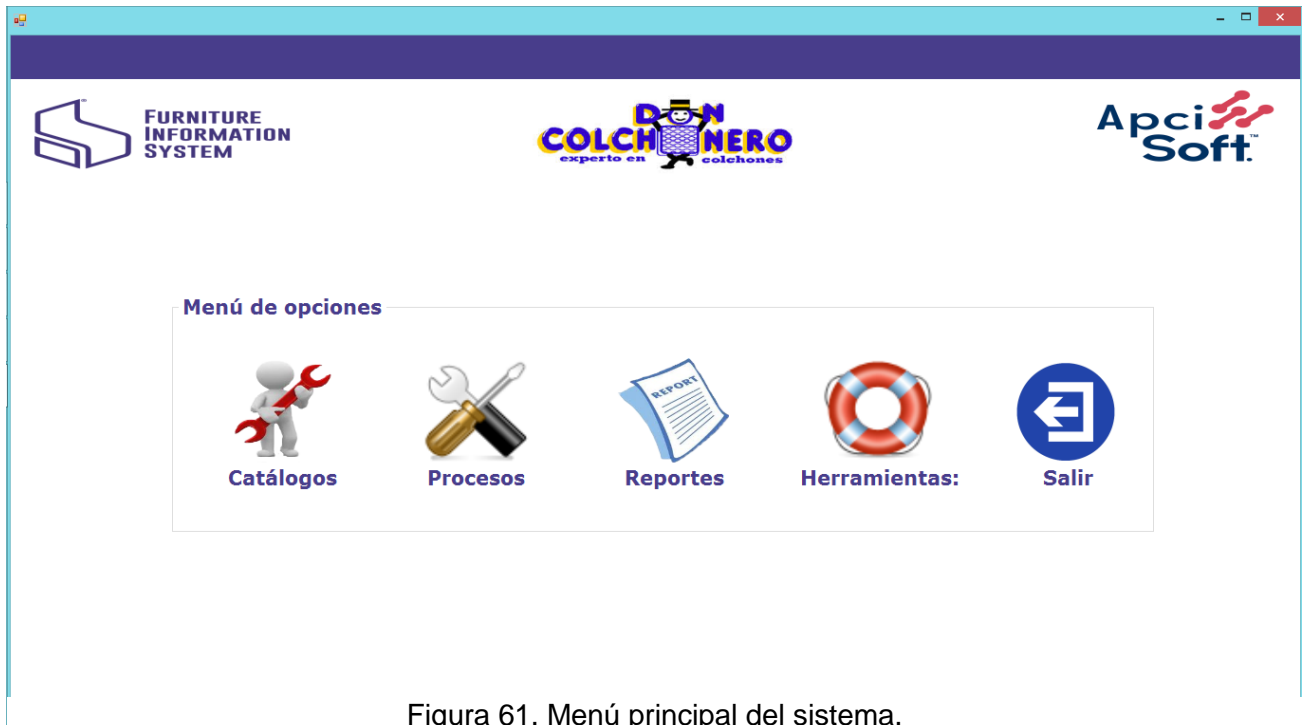


Figura 61. Menú principal del sistema.

En la figura No. 62 se muestra el catálogo de clientes.

Datos Del Cliente

ID: 10 Ciudad: Guadalajara
 Nombre: Mario Domicilio: Desconocido
 Apellidos: Castellanos Código P: 48752
 RFC: SDYR78E E-Mail: m1234567890@live.com
 Teléfono: 578-456-6561 Saldo: 0.00

Busqueda Por: ID NOMBRE

ID	Nombre	Apellidos	RFC	Teléfono	Ciudad	Domicilio	Cód. Postal	Email	Saldo
10	Mario	Castellanos	SDYR78E	578-456-6561	Guadalajara	Desconocido	48752	m1234567890@live...	0.00
20	Karla Aguirre	Jiménez Chávez	JCKA343DSD	333-312-4686	Guadalajara	Desconocido	49070	jimenez@gmail.com	0.00
21	Karen Sofía	Chávez Lázaro	CLKS345SDS	3333784686	Guadalajara	Conocido	49070	chavez@gmail.com	0.00
22	Manuel	Ramírez Cruz	RCMS3475SDS	342-378-4681	Tecalitlan	Conocido	59070	ramirez@gmail.com	0.00
34	Jóse Daniel	Orozco Martínez	OMJD3475SDS	548-002-3687	Tamazula	Conocido	30210	joseDa@gmail.com	0.00
36	Cristián	González Morán	GMCS3475SDS	332-561-0058	Mazamitla	Conocido	49500	cris@gmail.com	0.00
40	Mariana	Alcaraz Gama	ASDD09SD14	123-165-1321	Mazamitla	Conocido	49500	marianita@live.co...	0.00
50	Luis	Castellos	HJSKDH67	456-465-1987	Guzman	Conocido	78920	luiscastellos@live.c...	0.00
60	Melisa	Maya	KJHJ23H23	148-756-1132	Guzman	Conocido	87410	meli@hotmail.com	0.00
70	Natalia	Reyes Godines	48D6S64FFG	365-214-7800	Mazamitla	Conocido	49600	nata@hormail.com	11000.00

Figura 62. Catálogo de clientes.

En la figura No. 63 se muestra el catálogo de productos.

Datos Del Producto

Clave: 100 Precio de Venta: 600.00
 Descripción: Mesa de sala Existencias: 28
 Categoría: Cocina Stock: 5
 Precio de Costo: 540.00 Imagen:

Busqueda Por: DESCRIPCION

Código	Descripción	Categoría	Precio costo	Precio venta	Existencias	Stock	Imagen
100	Mesa de sala	Cocina	540.00	600.00	28	5	
101	Mesa de cocina	Sala	600.00	700.00	50	10	
102	Ropero	Recamara	700.00	750.00	28	25	
103	Sofa de Piel	Sala	300.00	400.00	60	7	
104	Silla madera	Otros	240.00	260.00	242	10	
105	Cama matrimonial	Recamara	2800.00	3000.00	63	12	
106	Banco mediano	Cocina	340.00	360.00	40	25	
107	Buró chico	Sala	620.00	680.00	40	10	
108	Cama individual	Recamara	1200.00	1400.00	320	8	

Figura 63. Catálogo de productos.

En la figura No. 64 se muestra el catálogo de proveedores.

Datos Del Proveedor

ID: 100 Domicilio: Desconocido

Nombre: Óscar Jaime Código P: 49088

Apellidos: Contreras E-Mail:

Compañía: Los héroes Teléfono: 365-214-7800

Ciudad: Guzmán Saldo: 0.00

Busqueda Por: APELLIDOS

ID	Nombre	Apellidos	Compañía	Ciudad	Domicilio	Cód. postal	Correo	Teléfono	Saldo
100	Óscar Jaime	Contreras	Los héroes	Guzmán	Desconocido	49088		365-214-7800	0.00
101	Manuel	Costelos	Los altos	Aguascalientes	Conocido	15201	manuel@live...	489-456-6545	0.00
102	Héctor Miguel	Llamas	Los inculcos	Guadalajara	Conocido	45210		215-468-1322	22000.00

Figura 64. Catálogo de proveedores.

En la figura No. 65 se muestra el catálogo de materiales.

Datos Del Material

ID: 1

Descripción: Brocha

Unidad: Piezas

Costo: 20.00

Stock: 5

Existencias: 231

Busqueda Por: DESCRIPCION

ID	Descripción	U. medida	Costo	Stock	Existencias
1	Brocha	Piezas	20.00	5	231
2	Martillo	Piezas	35.00	5	129
3	Pintura Café	Litros	58.00	5	3
4	Barnís	Litros	50.00	6	254
5	Pegamento	Litros	35.00	5	6
6	Tinta negra	Litros	220.00	25	0
7	Brocha grande	Piezas	22.00	250	0

Figura 65. Catálogo de materiales.

En la figura No. 66 se muestra el catálogo de personal.

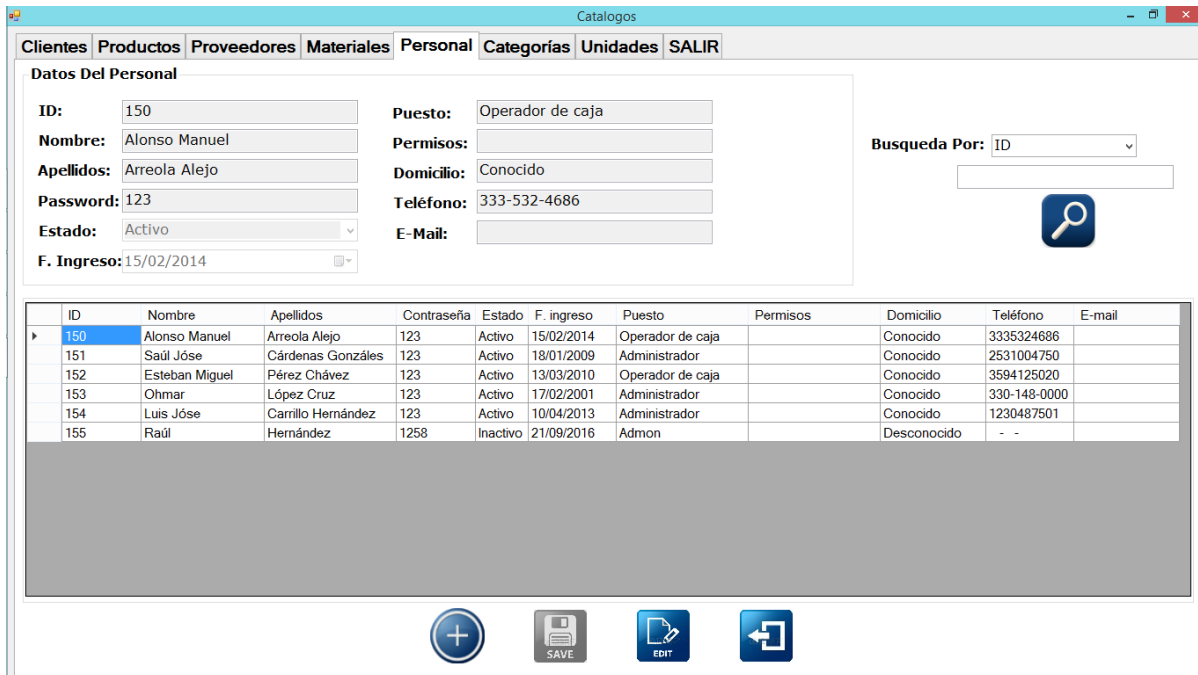


Figura 66. Catálogo de personal.

En la figura No. 67 se muestra el catálogo de categorías para la clasificación de productos.

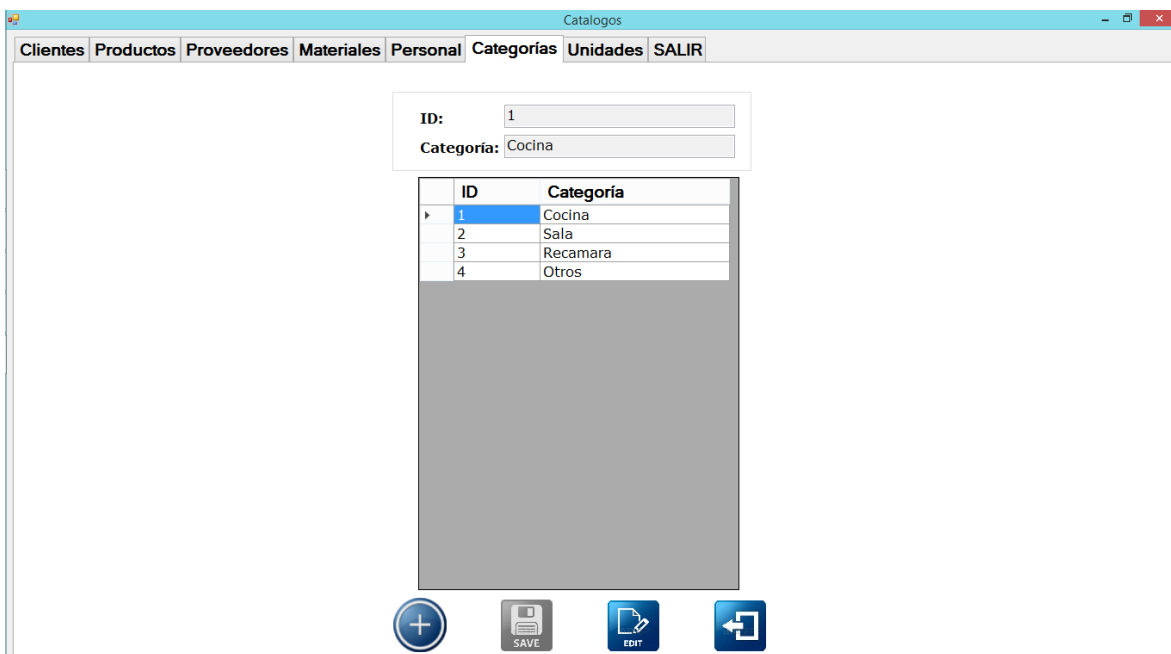


Figura 67. Catálogo de categorías.

En la figura No. 68 se muestra el catálogo de unidades para la clasificación de materiales. En la figura No. 69 se muestra el submenú de procesos.

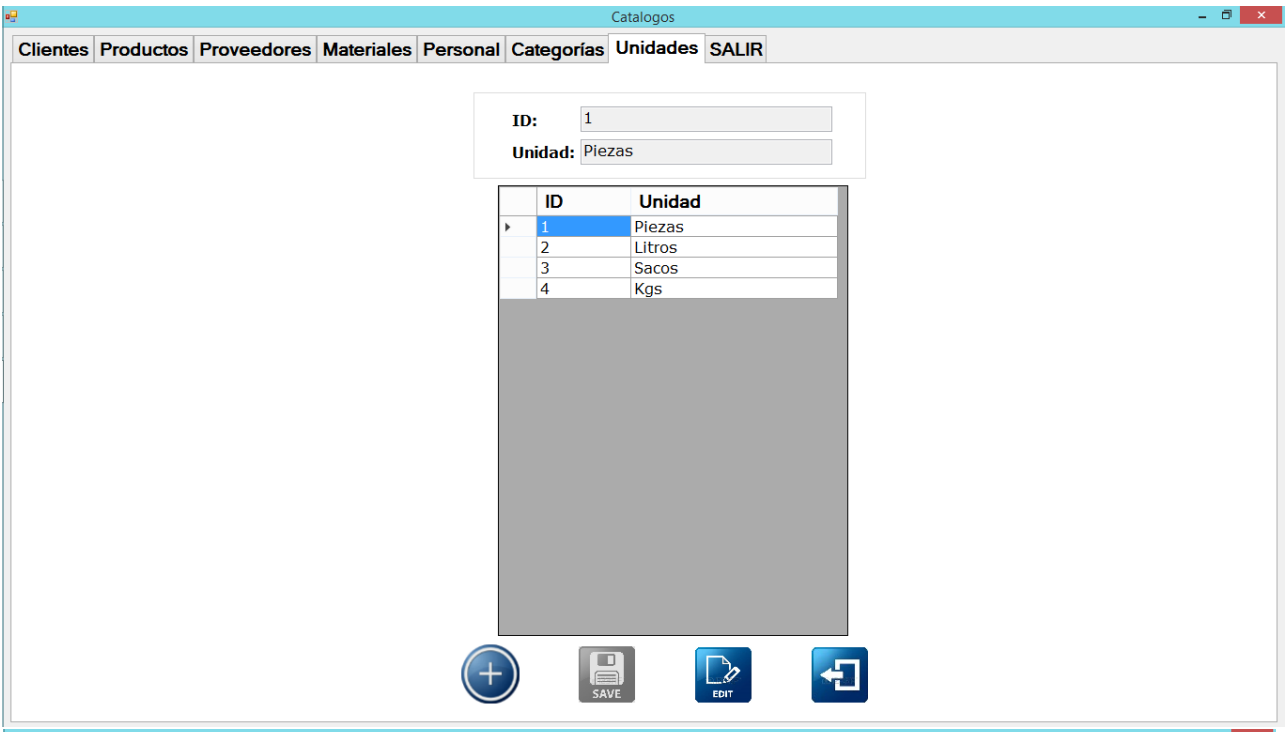


Figura 68. Catálogo de unidades.



Figura 69. Submenú de procesos.

En la figura No. 70 se muestra la pantalla principal de registrar ventas a crédito o contado.

Ventas

Registrar venta | Consulta general | Consulta individual | Consulta por fecha | Consulta por cliente y fecha

Datos Del Cliente

Apellidos: Ramírez Cruz Ciudad: Tecalitlan
 Nombre: Manuel Domicilio: Conocido
 Cuenta: 22 CP: 59070
 RFC: RCMS3475SDS Correo: ramirez@gmail.com
 SALDO: 0.00 Telefono: 342-378-4681

Fecha: 12/10/2016 Folio:

Condición: Contado
 Descuento: 0 %

Datos Del Personal

Apellidos: Pérez Chávez Puesto: Operador de caja
 Nombre: Esteban Miguel Teléfono: 3594125020
 Cuenta: 152 Domicilio: Conocido

Efectivo: Cambio:

Producto:
 Descripción: Sofa de Piel
 Clasificación: Sala
 Código: 103
 Precio/C: 300.00
 Stock: 7
 Existencias: 60

Agregar A Detalle:
 Precio/V: 400.00
 Importe:
 Cantidad:

Detalle de Venta

Código	Producto	Precio/V.	Cantidad	Importe	Quitar
103	Sofa de Piel	400.00	2	800	DELETE

Subtotal: 800 Iva: 128 Descuento: 0 Total: 928

Figura 70. Pantalla principal de registrar ventas.

En la figura No. 71 se muestra la pantalla principal para registrar abonos de las ventas realizadas a crédito.

Abonos

Registrar abono | Consulta general | Consulta individual | Consulta por cliente | Consulta por fecha | Consulta por venta

Datos Del Cliente

Nombre: Natalia Ciudad: Mazamitla
 Apellidos: Natalia Domicilio: Conocido
 Cuenta: 70 Código P: 49600
 RFC: 48D6S64FFG E-Mail: nata@hormail.com
 SALDO: 11000.00 Teléfono: 365-214-7800

Fecha: 12/10/2016 FOLIO:

Ventas:

Folio	Personal	Fecha	F. Vencimiento	Subtotal	Iva	Descuento	Total	Abonos	Saldo
1	Raúl	07/10/2016	16/02/2017	9580.00	1532.80	0.00	11112.80	112.80	11000.00

Realizar Abono

Folio Venta: 1
 Importe:

Figura 71. Pantalla principal de registrar abonos.

5.4 Diseño de Informes

En la figura No. 72 se muestra un reporte general de personal, más dos gráficos; uno donde se aprecian el total de personal para cada puesto, y dentro del otro la representación del personal activo e inactivo.

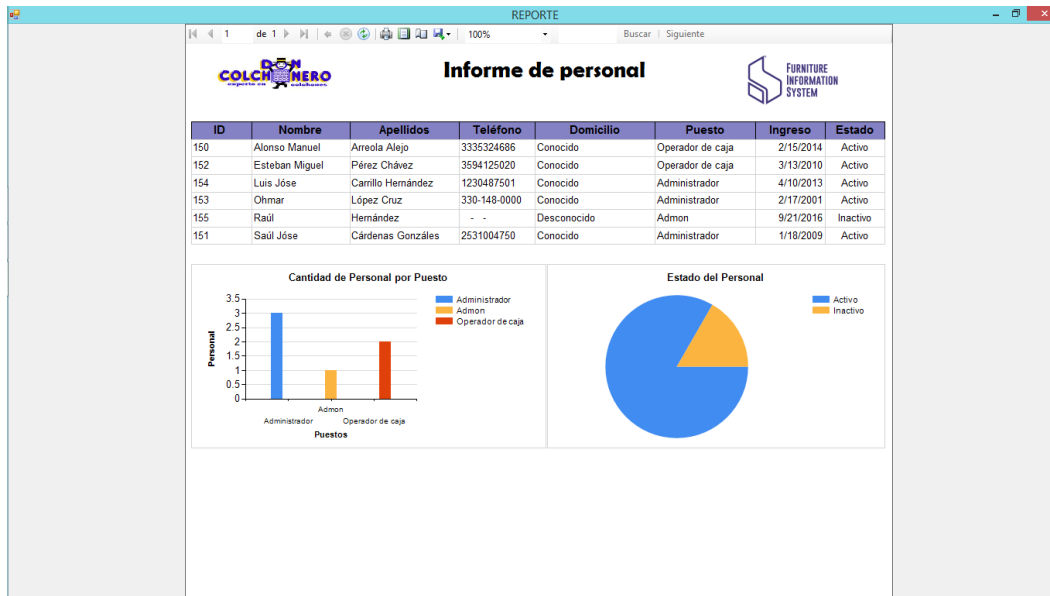


Figura 72. Reporte general de personal.

En la figura No. 73 se muestra un reporte general de clientes y un gráfico que representa las ciudades de donde pertenecen los clientes.



Figura 73. Reporte general de clientes.

En la figura No. 74 se muestra un reporte general de clientes con adeudos y un gráfico que representa las ciudades de donde pertenecen los clientes.



Figura 74. Reporte general de clientes con adeudos.

En la figura No. 75 se muestra un reporte general de proveedores y un gráfico que representa las ciudades de donde pertenecen los proveedores.

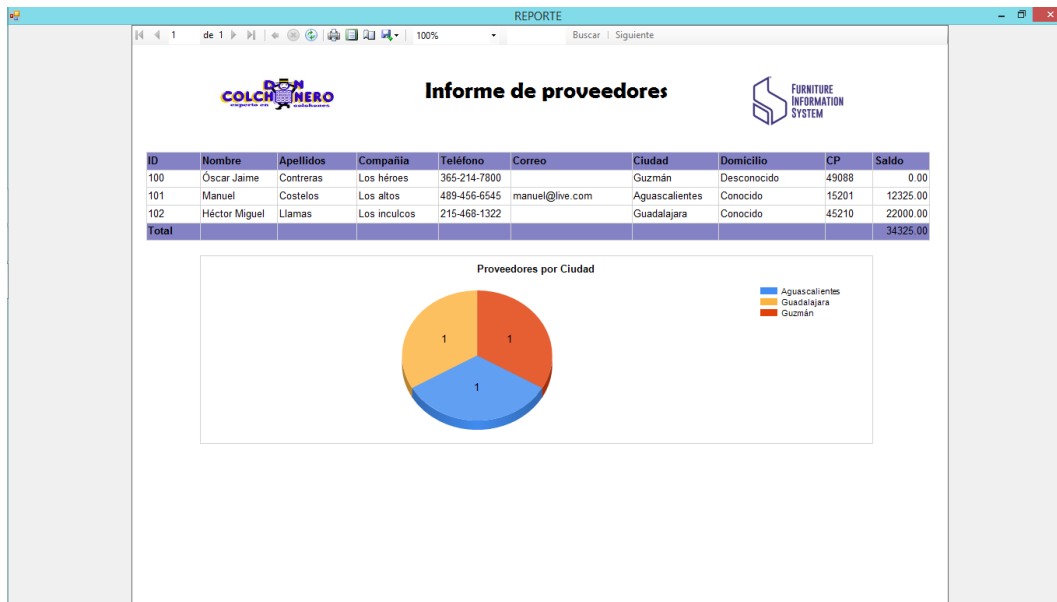


Figura 75. Reporte general de proveedores.

En la figura No. 76 se muestra un reporte general de materiales y un gráfico que representa las existencias y stock por material.

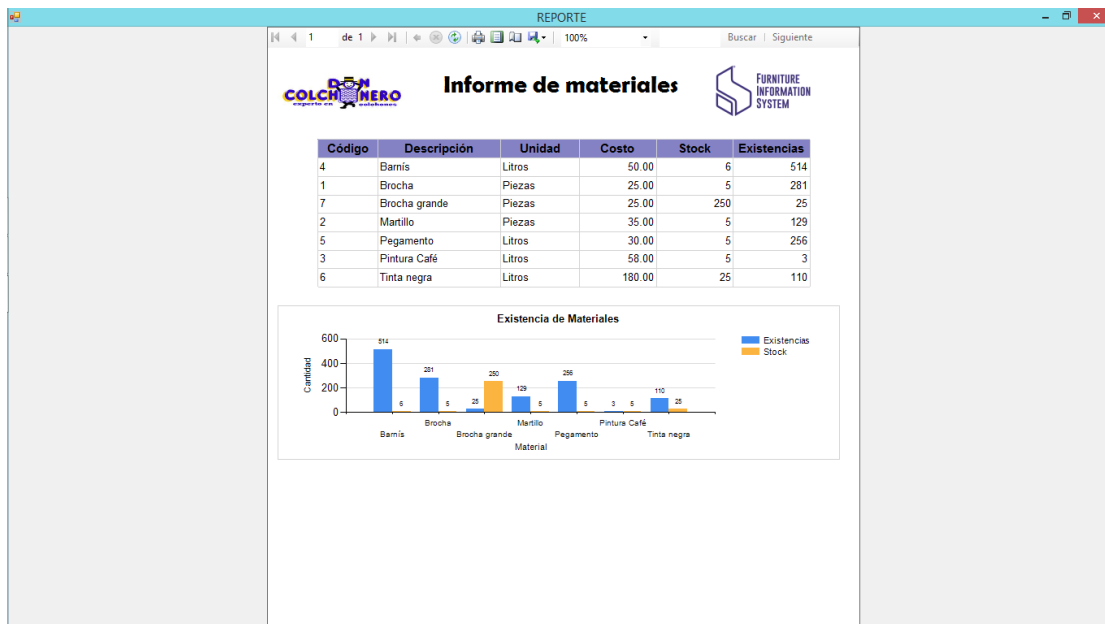


Figura 76. Reporte general de materiales.

En la figura No. 77 se muestra un reporte de productos por categoría y un gráfico que representa las existencias y stock por producto.



Figura 77. Reporte de productos por categoría.

En la figura No. 78 se muestra un reporte de compras de contado entre dos fechas y un gráfico que representa la cantidad de compras y monto total por proveedor.

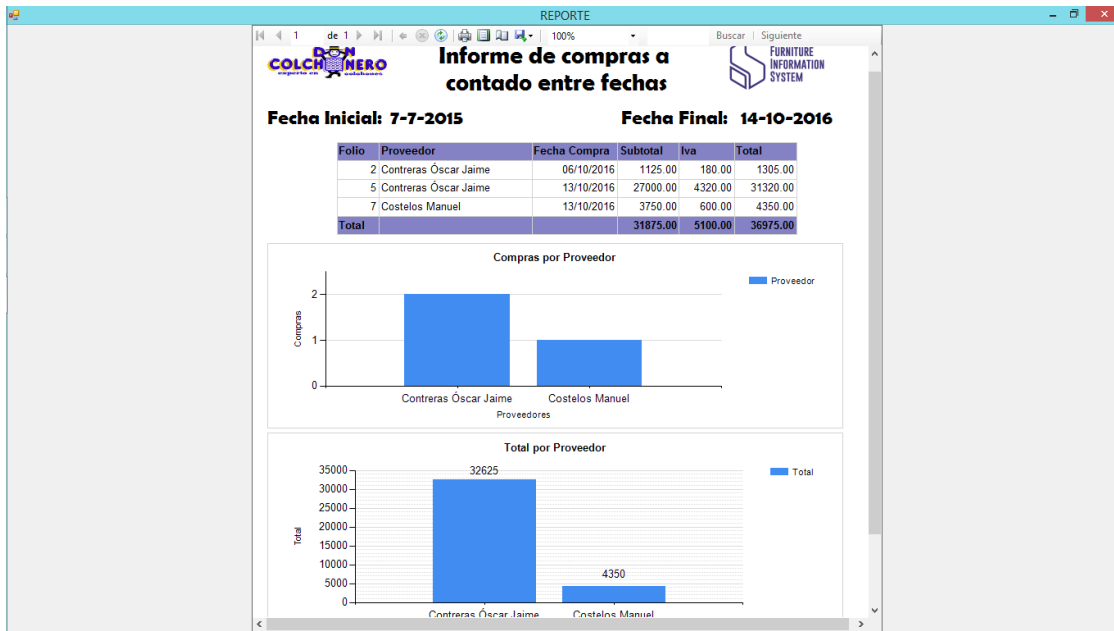


Figura 78. Reporte de compras de contado entre dos fechas.

En la figura No. 79 se muestra un reporte de ventas a crédito entre fechas y dos gráficos que representan las ventas y el monto por venta del cliente.

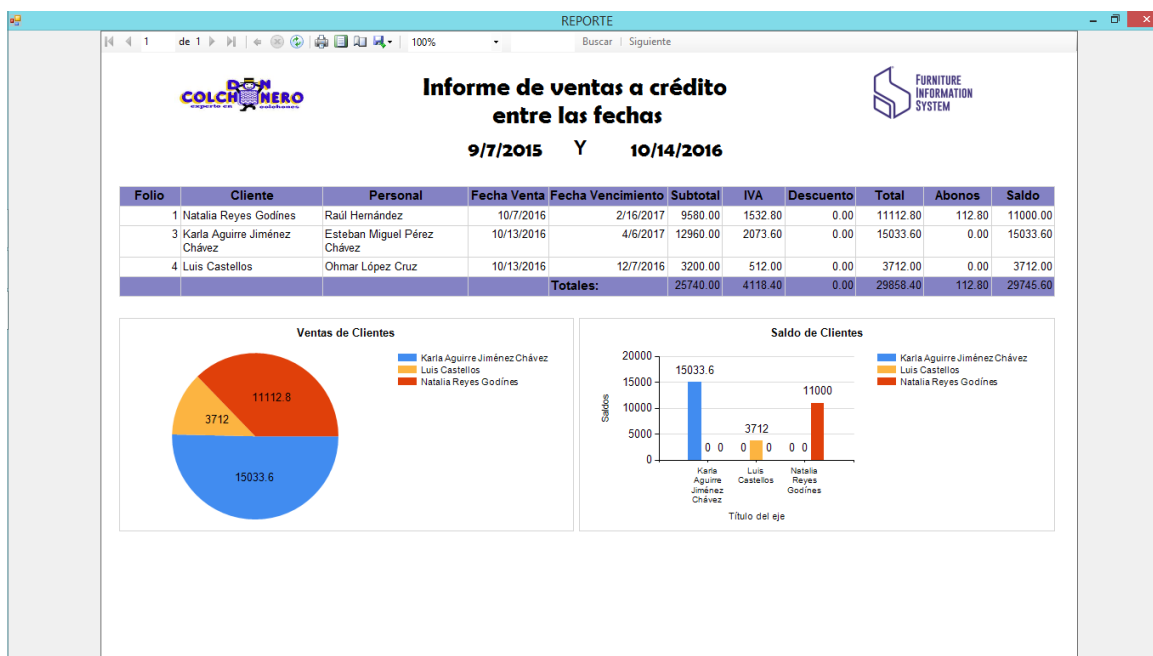


Figura 79. Reporte de ventas a crédito entre fechas.

CONCLUSIONES.

Se analizó y diseñó un sistema de información en cada uno de los procesos requeridos, que ayudó a la implementación del sistema en una red local y contribuyó a mejorar y agilizar las actividades administrativas cotidianas y concurrentes de la empresa “Don Colchonero Muebles”.

Este sistema aportará grandes ventajas a la empresa ya que al utilizarlo obtendrá una mayor administración de la información que se recopila cada día, que ayudará a tomar oportunamente las mejores decisiones para su negocio. La información que se genera se encuentra organizada y estructurada mediante consultas y reportes específicos que presentan una información íntegra y actualizada sobre las condiciones en que se encuentra la empresa.

El análisis y diseño de éste sistema aportó grandes y valiosos conocimientos, que ayudaron en adquirir mayor experiencia en el desarrollo de sistemas en la especialidad de Informática. La experiencia y las competencias que se lograron obtener son parte fundamental para las siguientes etapas que se presentarán tanto en el campo laboral como en la vida personal, ya que toda experiencia vivida y conocimientos adquiridos influye en todos los aspectos de la vida.

REFERENCIAS BIBLIOGRÁFICAS.

- BioDic, D. d. (16 de Junio de 2014). *BioDic Unificar*. Recuperado el 22 de Marzo de 2017, de https://www.biodic.net/palabra/unificar/#.WNK_tFXhDIU
- campusMVP. (22 de Septiembre de 2014). *Fundamentos de SQL: Inserción de datos - INSERT*. Recuperado el 26 de Abril de 2017, de <https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Insercion-de-datos-INSERT.aspx>
- campusMVP. (23 de Junio de 2014). *SELECT*. Recuperado el 26 de Abril de 2017, de *Fundamentos de SQL: Cómo realizar consultas simples con SELECT*: <https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Como-realizar-consultas-simples-con-SELECT.aspx>
- Castillo, E. (10 de Octubre de 2015). *DIAGRAMAS DE UML DINAMICOS VS ESTATICOS*. Recuperado el 25 de Abril de 2017, de <http://gruponextdiagramauml.blogspot.mx/2015/10/diagramas-de-uml-dinamicos-vs-estaticos.html>
- Clara, J. (11 de Octubre de 2015). *DIAGRAMA UML*. Recuperado el 25 de Abril de 2017, de <http://maribelteamuml.blogspot.mx/2015/10/uml-estaticos-y-dinamicos-julian-clara.html>
- Compitiendo, J. (21 de Octubre de 2016). *Definición de DENY y su sintaxis en sql server 2005*. Recuperado el 26 de Abril de 2017, de <https://juanmaritoniadi.wordpress.com/2010/11/23/definicion-de-deny-y-su-sintaxis-en-sql-server-2005/>
- Corporation, M. (11 de Marzo de 2016). *Diagramas de casos de uso de UML: Instrucciones*. Recuperado el 02 de Enero de 2017, de <https://msdn.microsoft.com/es-es/library/dd409432.aspx>
- Cruz, C. S. (22 de Agosto de 2014). *Prezi*. Recuperado el 02 de Enero de 2017, de <https://prezi.com/eov4nvponzzt/quinta-forma-normal-5fn/>
- Delgado, L. E. (21 de Mayo de 2016). *5.2 Lenguaje de Definición de Datos (DDL)*. Recuperado el 22 de Marzo de 2017, de http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/52_lenguaje_de_definicion_de_datos_ddl.html
- Díaz, S. G. (25 de Octubre de 2010). *UML*. Recuperado el 22 de Marzo de 2017, de *Casos de Uso y Diagramas de Clase*: <https://es.slideshare.net/guillermos/uml-casos-de-uso-y-diagramas-de-clase>
- Fernando, P. H. (29 de Julio de 2015). *Diagramas Estáticos*. Recuperado el 25 de Abril de 2017, de <https://www.scribd.com/doc/27996741/Presentacion-Diagramas-Estaticos>
- Fontela, C. (2011). *UML modelado de software para profesionales*. Buenos Aires, Argentina: Alfaomega. Recuperado el 4 de Enero de 2017
- Garay, J. (20 de Febrero de 2015). *DDL y DML en un dos por tres!!!* Recuperado el 26 de Abril de 2017, de <https://jfgaray.wordpress.com/2010/10/14/ddl-y-dml-en-un-dos-por-tres/>

- Hernández, O. E. (2 de Julio de 2015). *El Lenguaje Unificado de Modelado (UML)*. Recuperado el 22 de Marzo de 2017, de <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- IBM. (22 de Septiembre de 2016). *Sentencia GRANT de SQL*. Recuperado el 26 de Abril de 2017, de https://www.ibm.com/support/knowledgecenter/es/SS9UMF_12.1.0/igm/igm/tpc/dsq_priv_sql_grant.html
- Java, E. I. (3 de Julio de 2014). *SQL - Comando DROP- Sintaxis DROP*. Recuperado el 26 de Abril de 2017, de <http://java-white-box.blogspot.mx/2014/07/sql-comando-drop-sintaxis-drop.html>
- Knowledge, C. I. (12 de Septiembre de 2016). *DELETE*. Recuperado el 26 de Abril de 2017, de *Sentencia DELETE*: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0000939.html
- Knowledge, C. I. (4 de Agosto de 2016). *UPDATE*. Recuperado el 26 de Abril de 2017, de *Sentencia UPDATE*: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.udb.doc/admin/r0001022.htm
- LINUXTOTAL. (21 de Octubre de 2016). *MySQL, referencia de uso del comando 'ALTER TABLE'*. Recuperado el 26 de Abril de 2017, de https://www.linuxtotal.com.mx/index.php?cont=info_otros_005
- Pérez, P. J., & Merino, M. (16 de Febrero de 2017). *Definición de lenguaje de programación*. Recuperado el 22 de Marzo de 2017, de <http://definicion.de/lenguaje-de-programacion/>
- Rosa, K. E. (2005). *SQL SERVER*. Buenos Aires, Argentina: MP Ediciones. Recuperado el 5 de Enero de 2017
- Rouse, M. (24 de Agosto de 2016). *SQL*. Recuperado el 22 de Marzo de 2017, de *SQL o lenguaje de consultas estructuradas*: <http://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas>
- Rouse, M. (12 de Octubre de 2016). *SQL Server*. Recuperado el 26 de Abril de 2017, de <http://searchdatacenter.techtarget.com/es/definicion/SQL-Server>
- TechNet, M. (28 de Diciembre de 2014). *TRUNCATE*. Recuperado el 26 de Abril de 2017, de *Eliminar todas las filas mediante TRUNCATE TABLE*: [https://technet.microsoft.com/es-es/library/ms188249\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms188249(v=sql.105).aspx)
- TechNet, M. (14 de Noviembre de 2016). *DENY (Transact-SQL)*. Recuperado el 26 de Abril de 2017, de [https://technet.microsoft.com/es-es/library/ms188338\(v=sql.110\).aspx](https://technet.microsoft.com/es-es/library/ms188338(v=sql.110).aspx)