

SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE CLIENTES,
PROVEEDORES E INVENTARIOS
GESINV

JOHN FERNEY CHAVES CELIS
JONATHAN DIAZ VARGAS

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
CENTRO REGIONAL SOACHA
2010

SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE CLIENTES,
PROVEEDORES E INVENTARIOS
GESINV

JOHN FERNEY CHAVES CELIS
JONATHAN DIAZ VARGAS

CRISTIAN ACOSTA
Asesor metodológico

MAURICIO ORLANDO BERMUDEZ
Asesor tecnológico

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
CENTRO REGIONAL SOACHA
2010

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Soacha 07 de abril de 2010

DEDICATORIA

A nuestros Seres queridos, Familia y a la Universidad que nos brindó los conocimientos necesarios para que lucháramos y diéramos lo mejor y aplicar lo aprendido en la vida real, finalmente a todas aquellas personas que de una manera u otra nos apoyaron para terminar satisfactoriamente este proyecto.

AGRADECIMIENTOS

Principalmente a Dios que nos dio la fortaleza y el anhelo de luchar para construir parte de nuestro sueño que a partir de ahora empieza a formarnos como nuevas personas, para infundir los conocimientos en nuestro campo laboral.

A nuestros padres que se esforzaron por brindarnos el apoyo que necesitábamos para terminar nuestra carrera.

A nuestros asesores:

Mauricio Bermúdez (Asesor Tecnológico, Coordinador Tecnología en Informática)
Cristian Acosta (Asesor Metodológico)

Quienes nos brindaron parte de su tiempo y conocimientos para que nosotros lo aplicáramos en el proyecto.

TABLA CONTENIDO

| | Pág. |
|-----------------------------|------|
| 1. INTRODUCCIÓN | 1 |
| 2. TÍTULO DEL PROYECTO | 2 |
| 3. DESCRIPCIÓN DEL PROBLEMA | 3 |
| 4. OBJETIVOS | 4 |
| 4.1. OBJETIVO GENERAL | 4 |
| 4.2. OBJETIVOS ESPECIFICOS | 4 |
| 5. JUSTIFICACIÓN | 5 |
| 6. ALCANCE | 6 |
| 7. MARCO REFERENCIAL | 7 |
| 7.1. MARCO TEORICO | 7 |
| 7.1.1. MYSQL | 7 |
| 7.1.2. JAVA | 8 |
| 7.1.3. HTTP | 9 |
| 7.1.4. JAVA SCRIPT | 9 |
| 7.1.5. HTML | 10 |
| 7.1.6. CSS | 11 |
| 7.2. MARCO CONCEPTUAL | 12 |
| 7.3. MARCO INSTITUCIONAL | 13 |

| | | |
|-----|---|----|
| 8. | DICCIONARIO DE DATOS | 36 |
| 9. | ANALISIS Y DISEÑO DE LA INTERFAZ DEL PROCESAMIENTO DE DATOS | 48 |
| 10. | REFERENCIAS | 79 |
| 11. | CONCLUSIONES | 80 |
| 12. | GLOSARIO | 81 |

LISTAS DE FIGURAS

| | Pág. |
|--|------|
| Figura 1. Sistema actual de la Comercializadora 2012 (1) | 14 |
| Figura 2. Sistema actual de la Comercializadora 2012 (2) | 15 |
| Figura 3. Registro de mercancía | 16 |
| Figura 4. Diagrama de secuencia registro de mercancía | 17 |
| Figura 5. Salida de mercancía | 18 |
| Figura 6. Diagrama de secuencia salida de mercancía | 19 |
| Figura 7. Inventario | 20 |
| Figura 8. Diagrama de secuencia inventario | 21 |
| Figura 9. Sistema propuesto | 22 |
| Figura 10. Entrada de mercancía | 23 |
| Figura 11. Diagrama de secuencia entrada de mercancía | 24 |
| Figura 12. Devoluciones de entrada | 25 |
| Figura 13. Diagrama de secuencia devolución de entrada | 26 |
| Figura 14. Salida de mercancía | 27 |
| Figura 15. Diagrama de secuencia salida | 28 |
| Figura 16. Devoluciones de salida | 29 |
| Figura 17. Diagrama de secuencia devoluciones de salida | 30 |
| Figura 18. Diagrama de clases | 31 |

| | |
|---|----|
| Figura 19. Modelo entidad-relación | 32 |
| Figura 20. Ingreso a GESINV | 39 |
| Figura 21. Menú de GESINV | 40 |
| Figura 22. Ingreso de usuario | 41 |
| Figura 23. Buscar usuario | 42 |
| Figura 24. Listar usuario | 43 |
| Figura 25. Modificar usuario (1) | 43 |
| Figura 26. Modificar usuario (2) | 44 |
| Figura 27. Eliminar usuario | 45 |
| Figura 28. Nuevo Reporte de entrada | 45 |
| Figura 29. Listado Reporte de entrada | 46 |
| Figura 30. Modificar Reporte mercancía | 47 |
| Figura 31. Modificar Reporte de entrada | 47 |

LISTA DE TABLAS

| | Pág. |
|-----------------------------|------|
| Tabla 1. Proveedor | 33 |
| Tabla 2. Cabecera entrada | 33 |
| Tabla 3. Cabecera salida | 34 |
| Tabla 4. Ciudad | 34 |
| Tabla 5. Detalle entrada | 34 |
| Tabla 6. Detalle salida | 35 |
| Tabla 7. Producto | 35 |
| Tabla 8. Punto distribución | 36 |
| Tabla 9. Tipo detalle | 36 |
| Tabla 10. Tipo devolución | 36 |
| Tabla 11. Tipo documento | 37 |
| Tabla 12. Tipo usuario | 37 |
| Tabla 13. Usuario | 38 |

1. INTRODUCCIÓN

Una de las substanciales dificultades que se presentan en una comercializadora es el de tener un control adecuado sobre los procesos que se llevan a cabo dentro de la misma como son: Inventario de mercancía, para evitar la pérdida de tiempo e inconsistencias en la información, en el momento de realizar algún movimiento que comprometiera los recursos y materiales que se encontraban en la misma.

Lo que se ofrece es un recurso de software prototipo diseñado y elaborado para solventar el problema mencionado anteriormente, permite y concede la facilidad del ingreso de datos, fácil consulta y validación de los mismos. Esta solución se ejecuta por medio de un proceso que se basa en un control para la estabilidad de los datos; se encarga de almacenar los diferentes procesos que se realizan en el área de inventario y consulta de proveedores dentro de una base de datos controlando así las posibles fragilidades que se puedan presentar en la información de la comercializadora.

2. TITULO DEL PROYECTO

SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE CLIENTES, PROVEEDORES E INVENTARIOS GESINV

La meta que se pretende alcanzar al finalizar este proyecto es contar con una solución de software que nos permita ofrecer un manejo adecuado de la información de la COMERCIALIZADORA

3. DESCRIPCIÓN DEL PROBLEMA

En la actualidad la comercializadora cuenta con un sistema de inventario el cual se elabora de forma manual en un cuaderno y también se almacena algunas existencias de la mercancía en hojas de cálculo en Excel, de esta forma se ha generando dificultades en los siguientes aspectos: Control de inventario y proveedores.

Este aprieto se origina en la falta de orden y control que tienen tanto en la información de los productos como en los diferentes procesos de venta y en todo lo relacionado con el inventario de los productos; además no cuentan con él suficiente apoyo tecnológico, ni de personal capacitado que los orientaran en este tema, ocasionando así pérdida de tiempo al realizar cualquier proceso dentro de la comercializadora.

En resumen la solución propuesta a este hecho es la realización de un sistema de información que inicialmente consta de un módulo principal:
Inventario de productos: el cual tendrá registrados la cantidad de recursos que se encuentran almacenados dentro de la institución de forma ordenada, concreta y fehaciente, los registros se llevaran por cantidad, descripción, valor unitario y valor total.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Realizar un prototipo de un módulo de un sistema de información para la gestión, registro y control de inventario de productos de aseo que se encuentran en la COMERCIALIZADORA 2012.

4.2. OBJETIVOS ESPECIFICOS

- Analizar la información, documentación de la comercializadora 2012.
- Plantear el sistema propuesto.
- Diseñar el modelo entidad relación del prototipo.
- Crear el prototipo de las pantallas y formularios para el usuario.
- Analizar la viabilidad técnica en cuanto al sistema operativo más adecuado para este diseño.
- Realizar un presupuesto de la infraestructura a nivel de hardware y software a usar.
- Optimizar y probar la funcionalidad del prototipo.

5. JUSTIFICACION

Los innovadores estilos en el desarrollo de la tecnología, en lo concerniente a la creación y adelanto de información rápida y segura y a la optimización de recursos, hace necesario que se replanteen este tipo de programas, originados con anterioridad, ya que en general pueden presentar desventajas a nivel técnico en la implementación en cualquier tipo de empresa y por tanto en su beneficio.

Debido a lo anterior es necesario tener en cuenta la funcionalidad y las técnicas ya implementadas para orientar el sistema de información, lo que permitirá no solo ahorrar costos, sino también beneficiar de manera inmediata a la COMERCIALIZADORA.

Los beneficios frente a esto son muy claros ya que el usuario del sistema no debe remitirse a varios sitios en busca de una información específica. Entregando información clara, precisa y sin mayores complicaciones se establece en una premisa ineludible para el alcance de la efectividad y los objetivos en la transmisión de información por Internet. Los usuarios de este medio están en permanente búsqueda de rapidez, sencillez y puntualidad, tres conceptos claves ligados cada uno entre sí con un común denominador: receptores satisfechos (Sistema de información interactivo para la asociación de empresarios de Sibaté, Soacha y sur de Bogotá D.C. Asomuña, 2004, 3)

6. ALCANCE

El adelanto en los sistemas de información permite soluciones con respecto al procesamiento de datos de clientes e inventarios para la COMERCIALIZADORA, acelerando los sistemas de búsqueda y toda la información personal en la cual su historial sea actualizado periódicamente mejorando y optimizando los tiempos de procesamiento y de respuesta del sistema abierto.

7. MARCO REFERENCIAL

7.1. MARCO TEORICO

Las empresa hoy en día requieren de un sistema que maneje los datos de manera organizada y que no sea costoso, el lenguaje gestor de bases de datos relacionales con el cual se va a realizar el proyecto de la empresa es **MYSQL Y JAVA**.

7.1.1. MYSQL

SQL es una base de datos que esta generalmente definida como un conjunto integrado de datos que modelaran un universo dado. Este universo esta compuesto por objetos interrelacionados, los objetos de un mismo tipo constituyen una entidad y el lazo habido entre entidades se le denomina asociación.

El proceso de descripción de asociaciones y entidades se llama modelización y se hace con la ayuda de un modelo de datos, existen actualmente cuatro modelos de datos diferentes:

- **Modelo jerárquico.**
- **Modelo en red.**
- **Modelo objeto**
- **Modelo relacional:** el principio básico de este modelo consiste en representar tanto las entidades como las asociaciones con la ayuda de relaciones denominadas también tablas. Una tabla esta compuesta por líneas y columnas, cada línea representa un objeto (proveedor – articulo) las columnas representan los atributos de dicho objeto.

7.1.2. JAVA

A principios de los años '90 la firma Sun Microsystems desarrolló el lenguaje de programación Java. Este lenguaje orientado a objetos es compilado generalmente en un bytecode, pero también es posible crear código de máquina nativo.

Este bytecode se compila por lo general en el momento de ejecutarse, en forma de código nativo listo para la ejecución, y también existen dispositivos de hardware capaces de ejecutarlos directamente.

No hay que confundir JavaScript con Java, el primero es un lenguaje interpretado. La sintaxis de Java es similar a la utilizada por lenguajes C y C++, siendo más sencilla porque elimina herramientas de bajo nivel, tal es el caso de los punteros.

Durante varios años la biblioteca de clases de Java, requerida para ejecutar los programas realizados en tal lenguaje no estaba publicada como software libre, mientras que el compilador y la máquina virtual conforme a las especificaciones del Java Community Process habían sido licenciados bajo GNU General Public License.

La mayoría de las tecnologías Java fueron liberadas bajo esta licencia libre elaborada por Free Software Foundation entre noviembre de 2006 y mayo de 2007, ante la presión de la comunidad de desarrolladores de software libre.

También puede parecer útil el crear botones dinámicos, es decir, botones en los que utilizamos el mismo diseño y solo cambiamos el texto. Podremos por ejemplo crear un botón haciendo una única llamada a una función en la que introducimos el estilo del botón y el texto a introducir obteniendo automáticamente el botón deseado.

A partir de la librería de funciones gráficas podemos hacer esto y mucho más. Muchas otras funciones pensadas para Internet (tratamiento de cookies, accesos restringidos, comercio electrónico...) o para propósito general (funciones matemáticas, explotación de cadenas, de fechas, corrección ortográfica, comprensión de archivos...) son realizadas por este lenguaje. A esta inmensa librería cabe ahora añadir todas las funciones personales que uno va creando por necesidades propias y que luego son reutilizadas en otros sitios y todas aquellas intercambiadas u obtenidas en foros o sitios especializados.

7.1.3. HTTP

(Hypertext Transfer Protocol) se utiliza para transferencia de documentos HTML e imágenes entre clientes, esta orientado hacia la comunicación entre sistemas de información distribuidos, se basa en el envío de mensajes entre el cliente y el servidor. El cliente inicia una petición y el servidor le contesta.

7.1.4. JAVASCRIPT

Java script es un lenguaje de programación utilizado para crear paquetes encargados de realizar acciones dentro del sitio de una página Web. Con Java script podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Java script y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es del propio navegador.

Java script es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con solo un poco de práctica.

Entre las acciones técnicas que se pueden realizar en Java script tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, java script nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Java script es un lenguaje con muchas posibilidades, permite la programación de paquetes scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc.

7.1.5. HTML

HTML, HyperText Markup Language, es un lenguaje simple utilizado para crear documentos de hipertexto para WWW. No es un lenguaje de descripción de página como PostScript; HTML no permite definir de forma estricta la apariencia de una página, aunque una utilización algo desviada hace que se utilice en ocasiones como un lenguaje de presentación. Además, la presentación de la página es muy dependiente del browser (o programa navegador) utilizado: el mismo documento no produce el mismo resultado en la pantalla si se visualiza con un browser en modo línea, Mosaic o Netscape, o sea, HTML se limita a describir la estructura y el contenido de un documento, y no el formato de la página y su apariencia.

Una de las claves del éxito de WWW, aparte de lo atractivo de su presentación es sin duda, su organización y coherencia. Todos los documentos WWW comparten un mismo aspecto y una única interfaz, lo que facilita enormemente su manejo por parte de cualquier persona. Esto es posible porque el lenguaje HTML, en que están escritos los documentos, no solo permite establecer hiperenlaces entre diferentes documentos, sino que es un "lenguaje de descripción de página" independiente de la plataforma en que se utilice.

Es decir un documento HTML contiene toda la información necesaria sobre su aspecto y su interacción con el usuario, y es luego el browser que utilizemos el responsable de asegurar que el documento tenga un aspecto coherente, independientemente del tipo de estación de trabajo desde donde estemos efectuando la consulta.

Su simplicidad es tal que no es necesario utilizar un editor particular. Su gran permisividad exige rigor y atención en la estructura de documentos con el fin de que éstos se visualicen correctamente al margen del contexto y el browser utilizado.

Por tanto, HTML es un lenguaje muy sencillo que nos permite preparar documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos.

7.1.6. CSS

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe como se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando así la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

7.2. MARCO CONCEPTUAL

Las microempresas dedicadas a la comercialización de productos de aseo utilizaban tareas repetitivas que se multiplicaban y por estas razones se dio la necesidad de buscar una nueva forma de minimizar esfuerzos en la parte de organización y control de la información.

La tecnología se ha impuesto en los últimos tiempos como herramienta para llevar a cabo la sistematización de estas tareas repetitivas.

Para no ir muy lejos acudiremos a uno de las principales dificultades de las microempresas, el manejo de control de inventario, el cual tiene una gran relevancia, ya que anteriormente el manejo de la información se volvía más tedioso y se gastaba más tiempo, un insuficiente control del inventario genera pérdidas o extravíos de la información. Por todo esto se dio la necesidad de automatizar el control de inventarios.

7.3. MARCO INSTITUCIONAL

HISTORIA DE LA COMERCIALIZADORA 2012

La Comercializadora 2012, fue creada el 12 de Noviembre de 2005, con cinco empleados. Nuestras oficinas se encuentran localizadas en la Calle 5 No.69-29, en la Localidad de Kennedy de la ciudad de Bogotá.

QUIENES SOMOS

La Comercializadora 2012 es una microempresa que comenzó a fomentar los productos de aseo en la localidad de Kennedy, gracias a nuestro servicio nos fuimos posicionando en el mercado; ganando confiabilidad y satisfacción.

POLÍTICAS DE CALIDAD

Identificar y analizar las necesidades de nuestros CLIENTES para satisfacer sus necesidades con un excelente portafolio de productos de aseo.

MISION

Comercializadora 2012 es una microempresa que se dedica a la venta de productos de aseo; así como la gestión de los procesos y procedimientos correspondientes, en beneficio y satisfacción total de nuestros clientes con productos y servicios de excelente calidad. Con un personal altamente calificado para dar soluciones integrales que generen bienestar, compromiso y trabajo en equipo.

VISION

Ser la Comercializadora mayorista y operadora líder, representando los productos de aseo destacándose por el compromiso con el bienestar de nuestros clientes a través de artículos de alta calidad que satisfagan sus necesidades.

VALORES CORPORATIVOS

Para involucrarnos con una conciencia real y transformadora, entendemos que nuestros pensamientos, acciones y políticas deben proyectarse hacia la promoción de los siguientes valores, componentes de la brújula que guía nuestro compromiso individual y colectivo: Solidaridad

**COMERCIALIZADORA DE
PRODUCTOS DE ASEO**

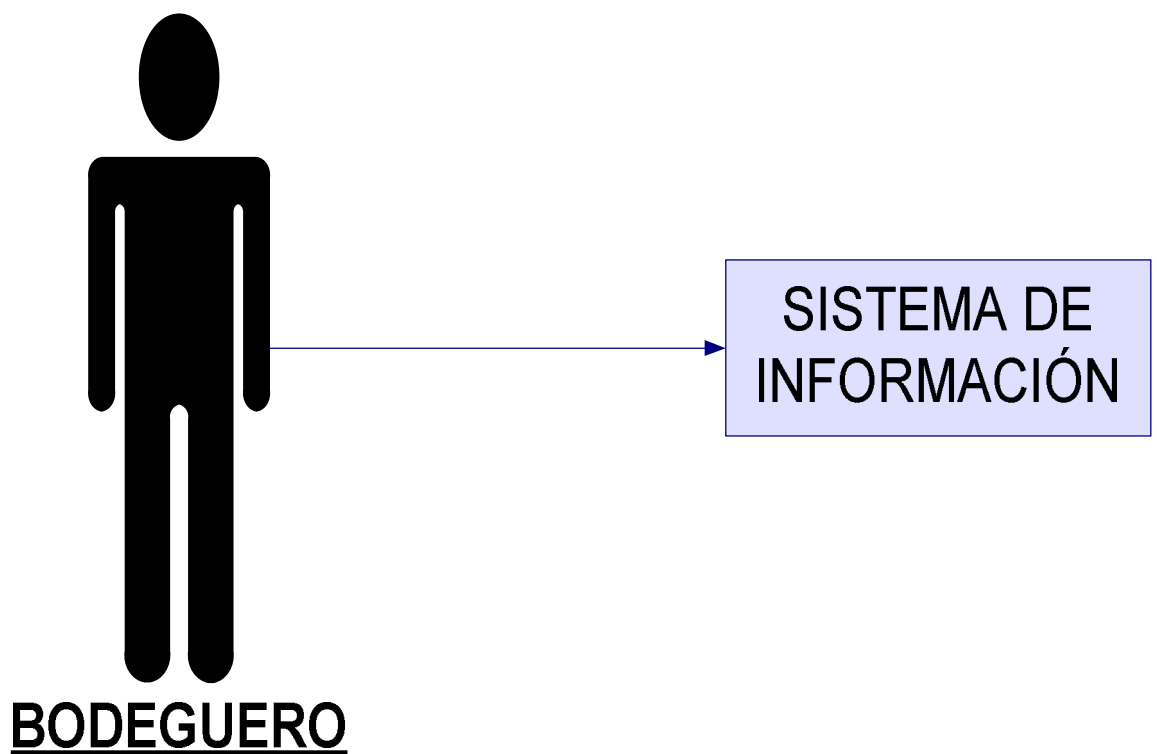


Figura 1. Sistema actual de la Comercializadora 2012 (1)

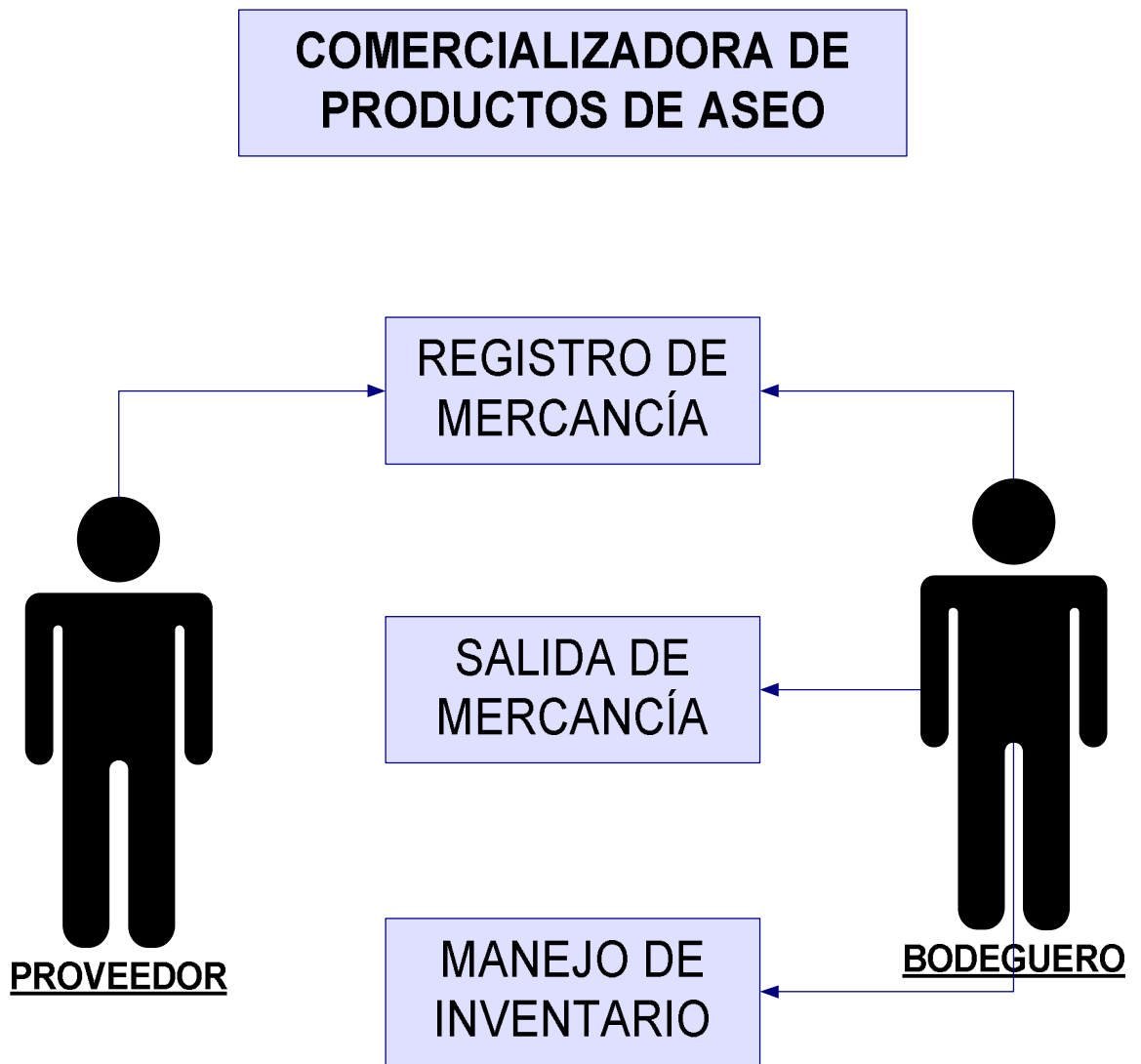


Figura 2. Sistema actual de la Comercializadora 2012 (2)

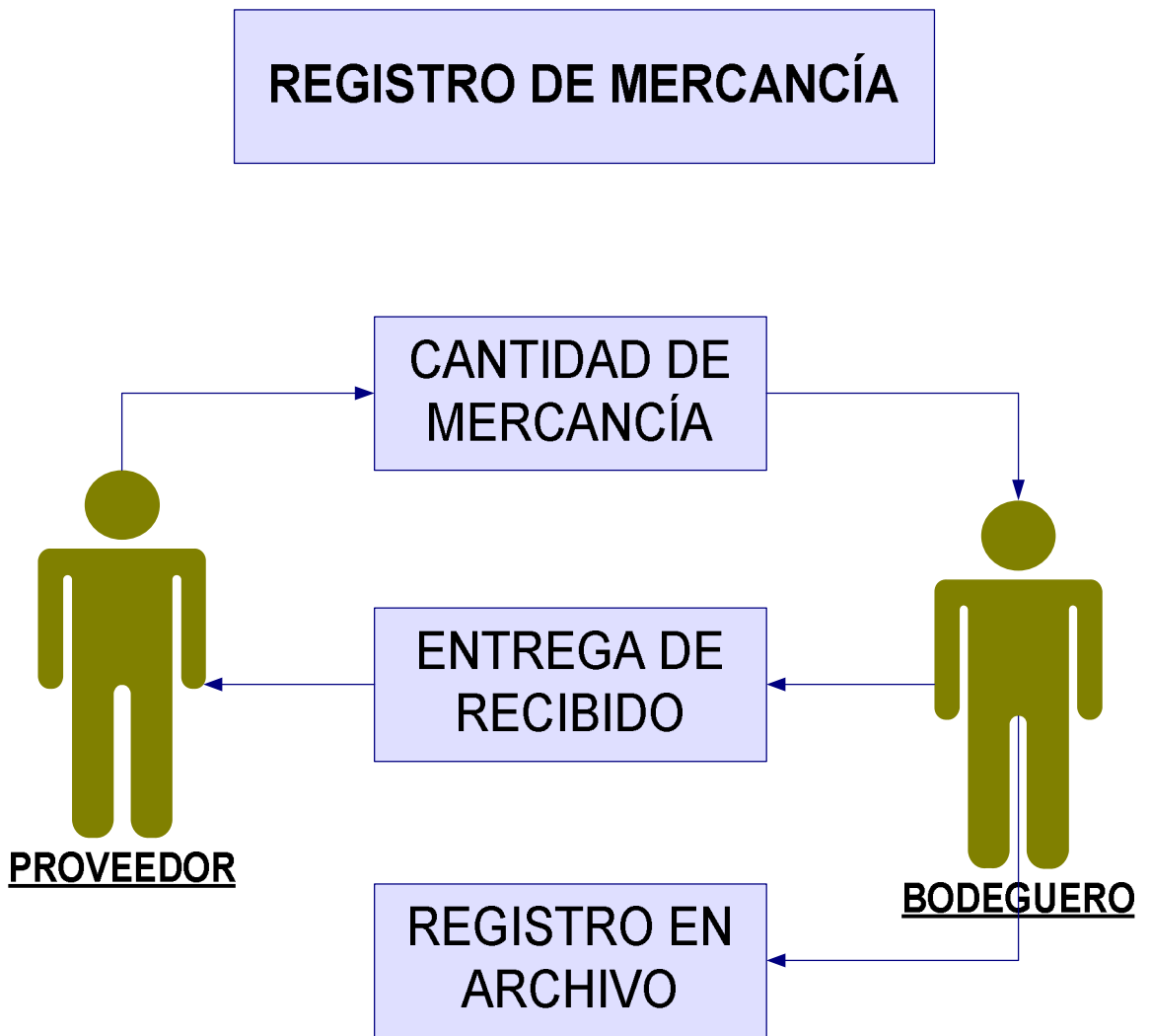


Figura 3. Registro de Mercancía

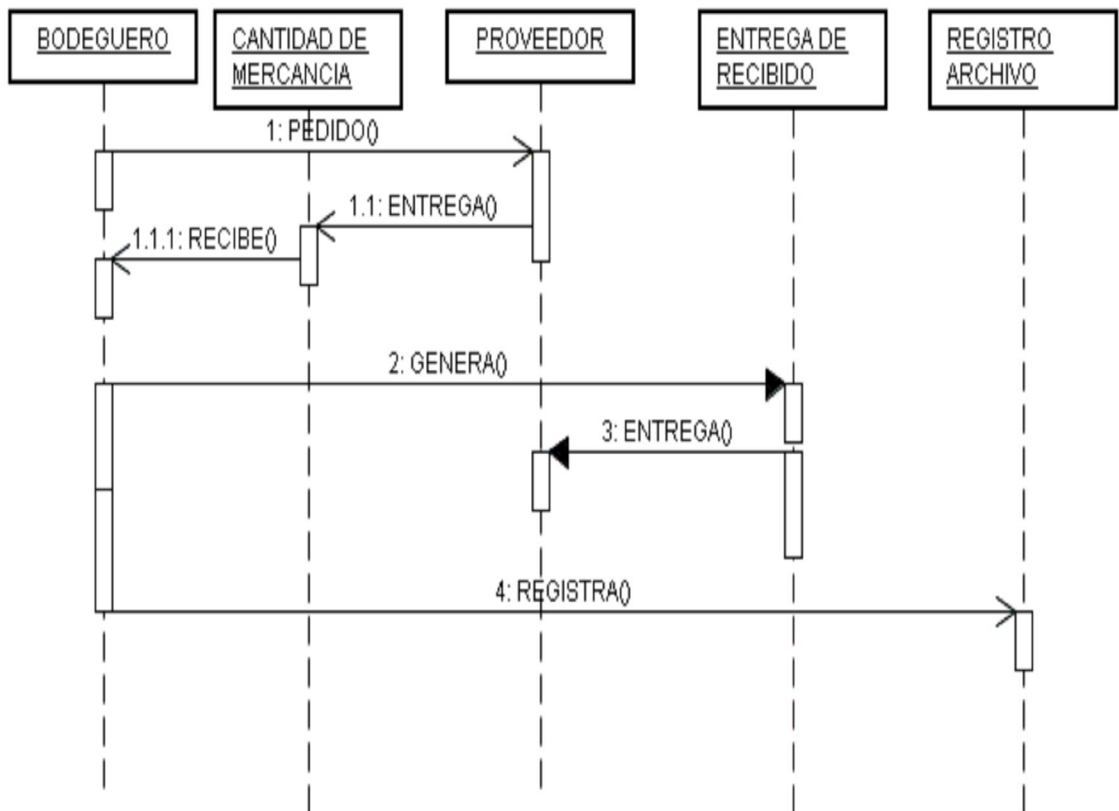


Figura 4. Diagrama de secuencia Registro de mercancía

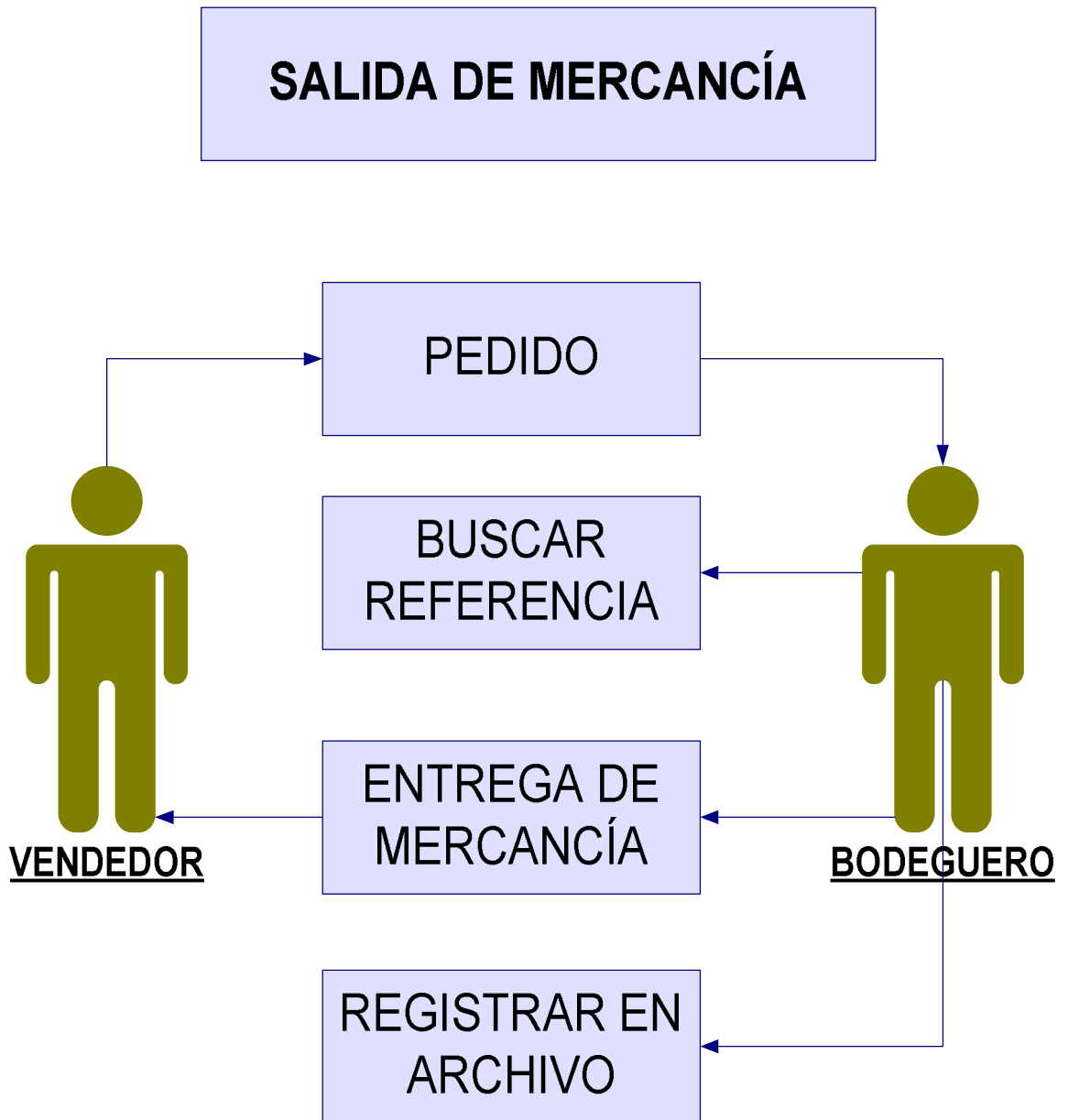


Figura 5. Salida de mercancía

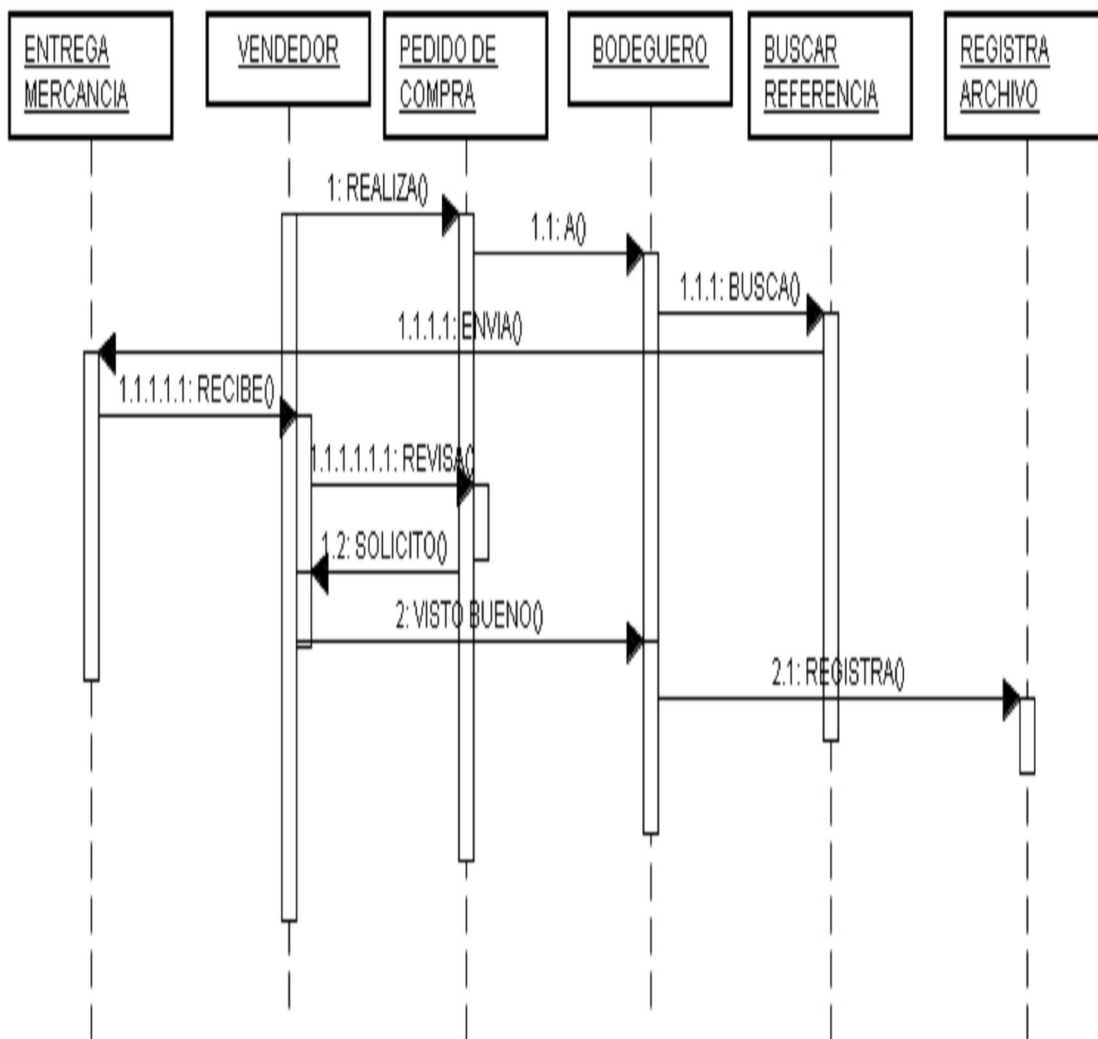


Figura 6. Diagrama de secuencia Salida de mercancía

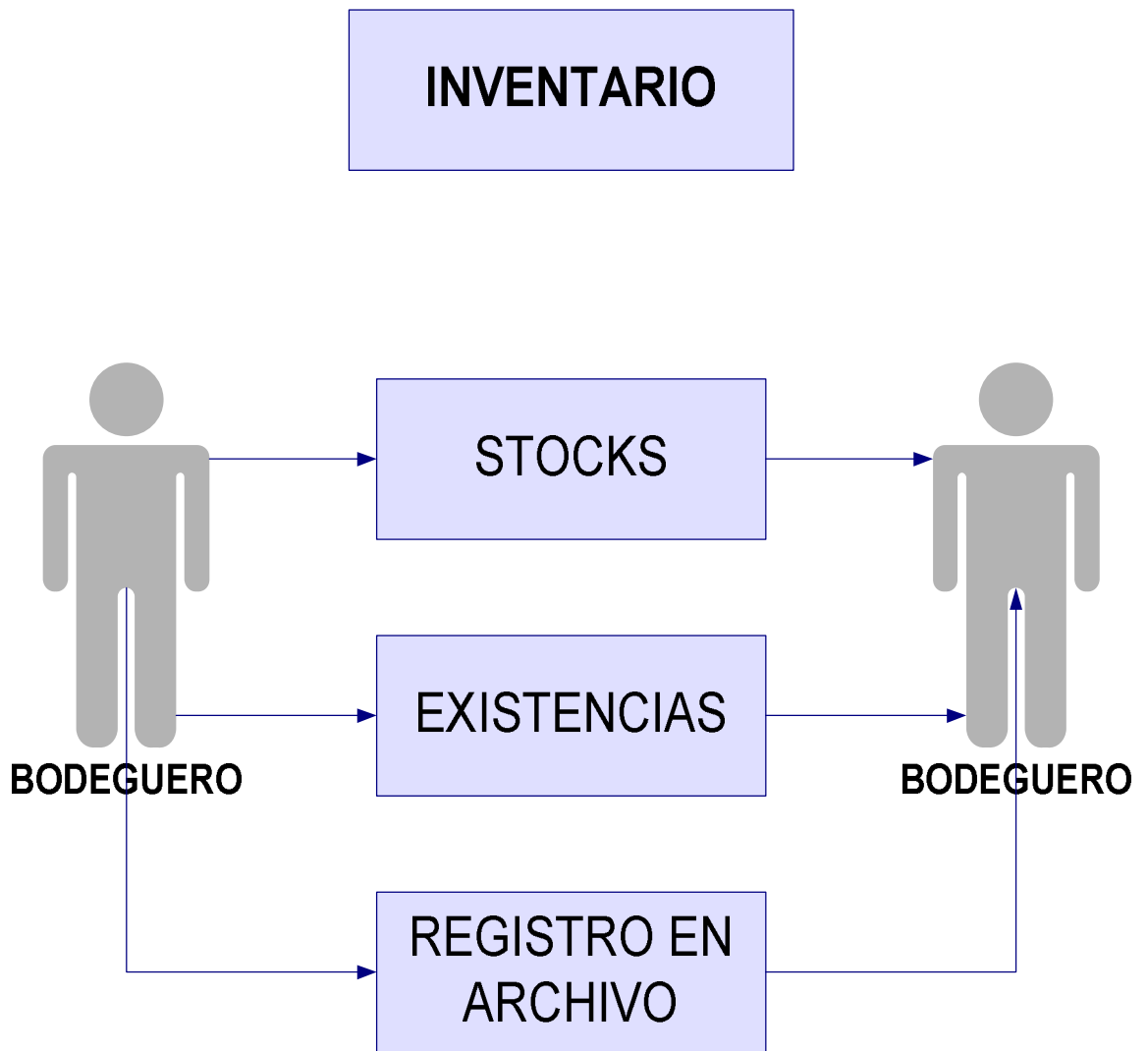


Figura 7. Inventario

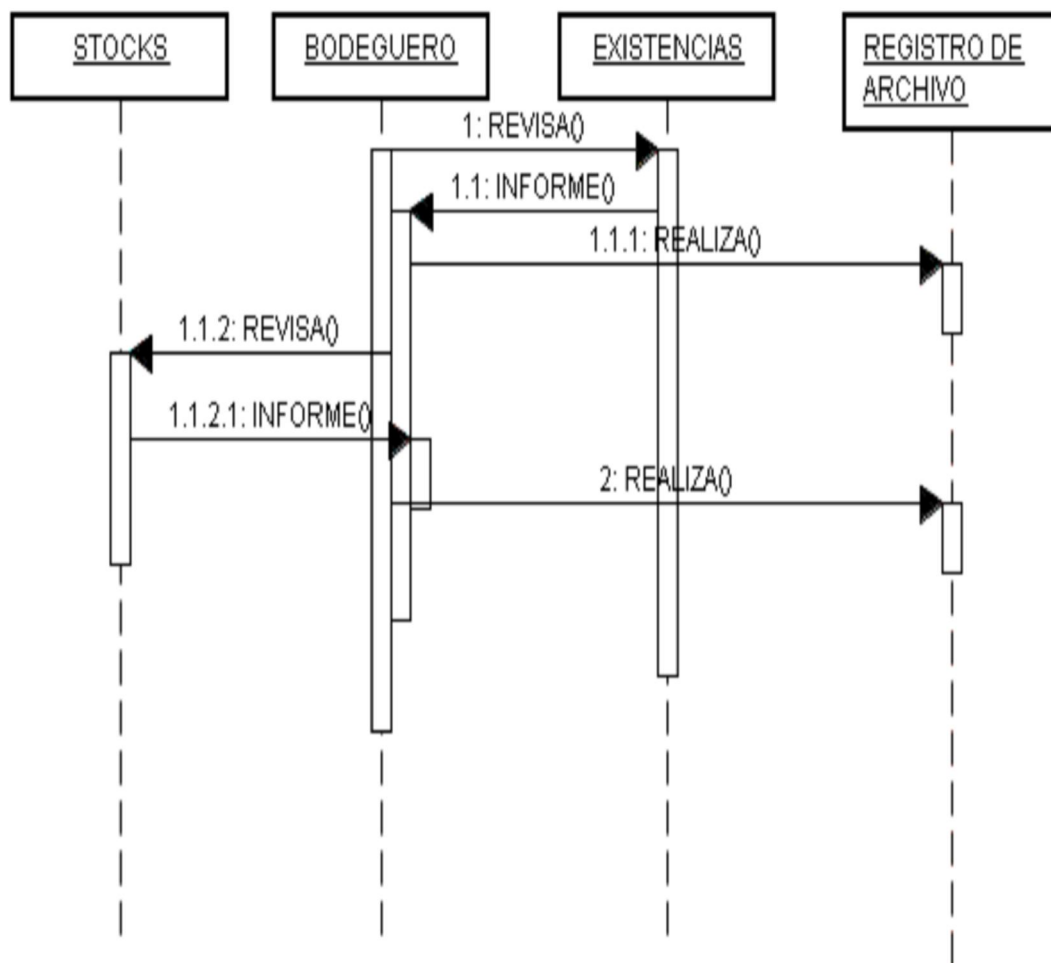


Figura 8. Diagrama de secuencia Inventario

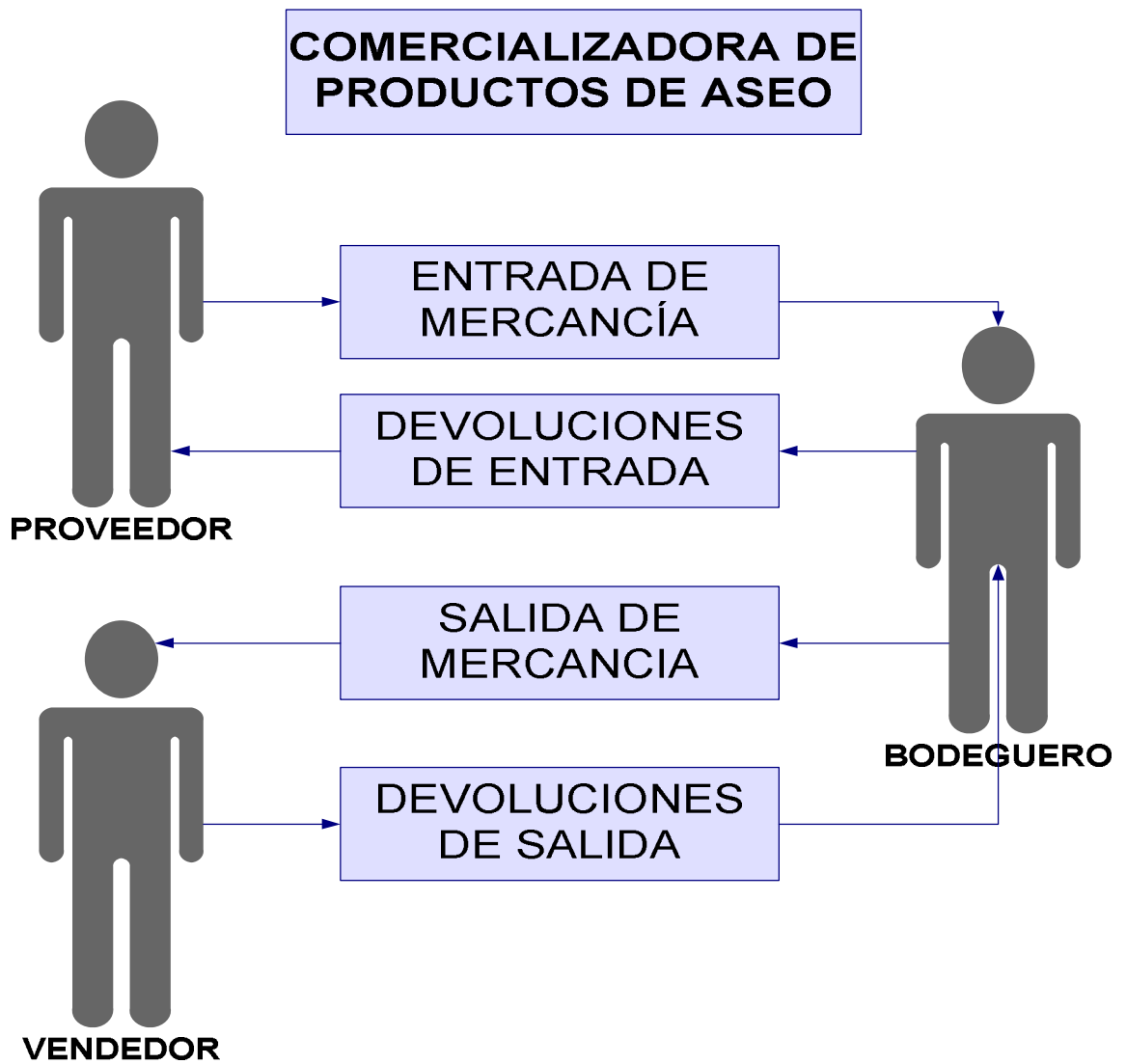


Figura 9. Sistema Propuesto

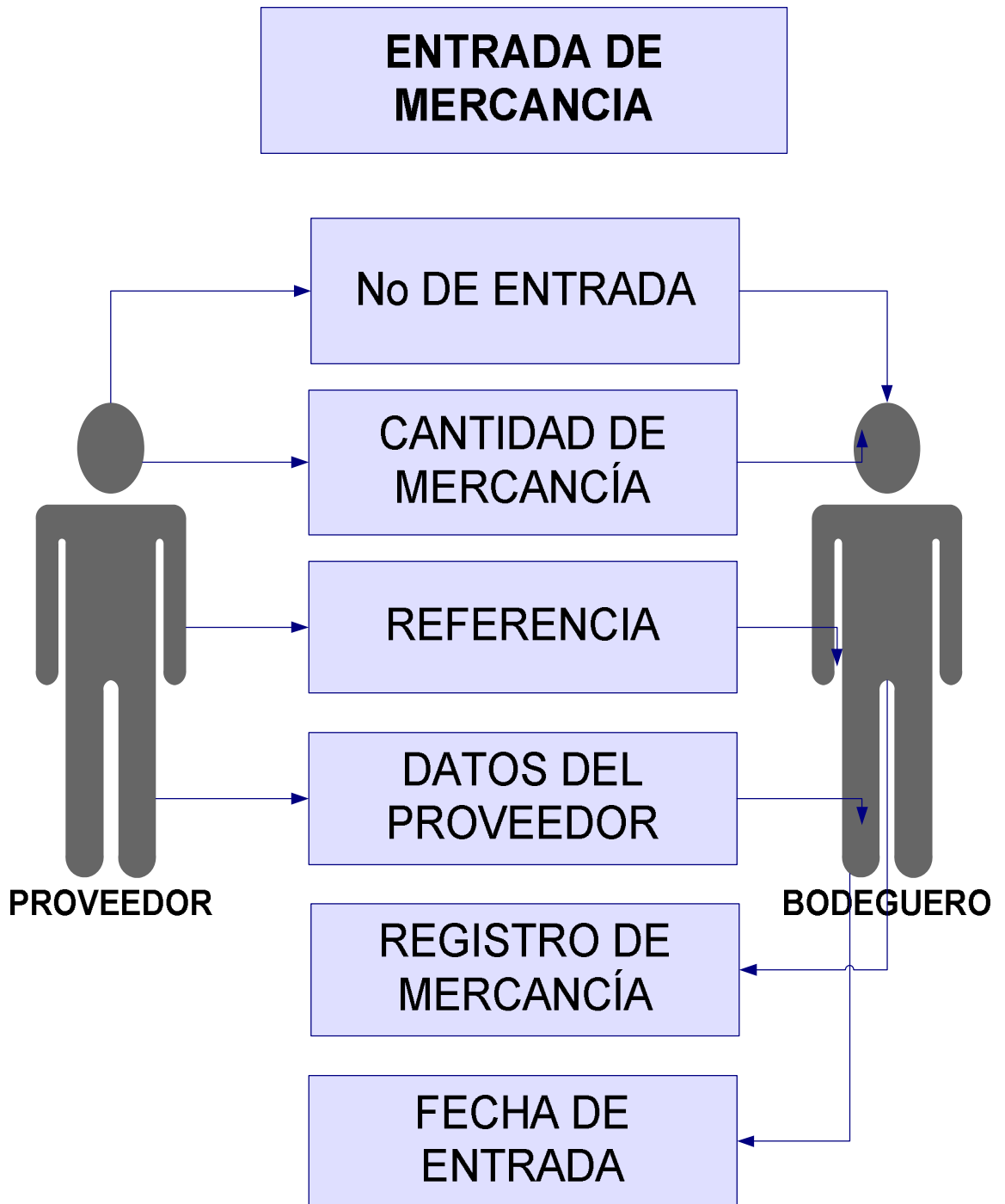


Figura 10. Entrada de mercancía

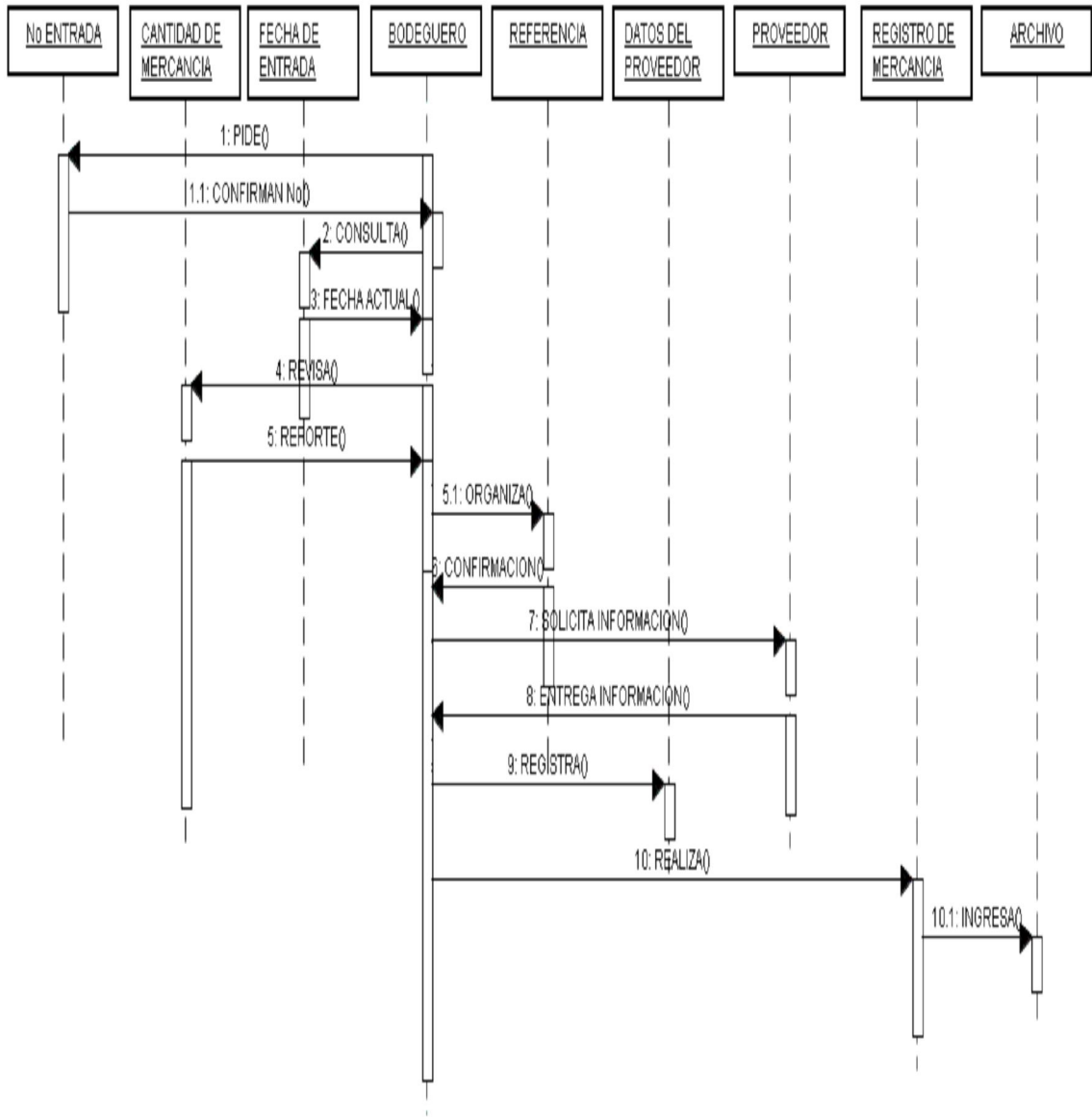


Figura 11. Diagrama de secuencia entrada de mercancía

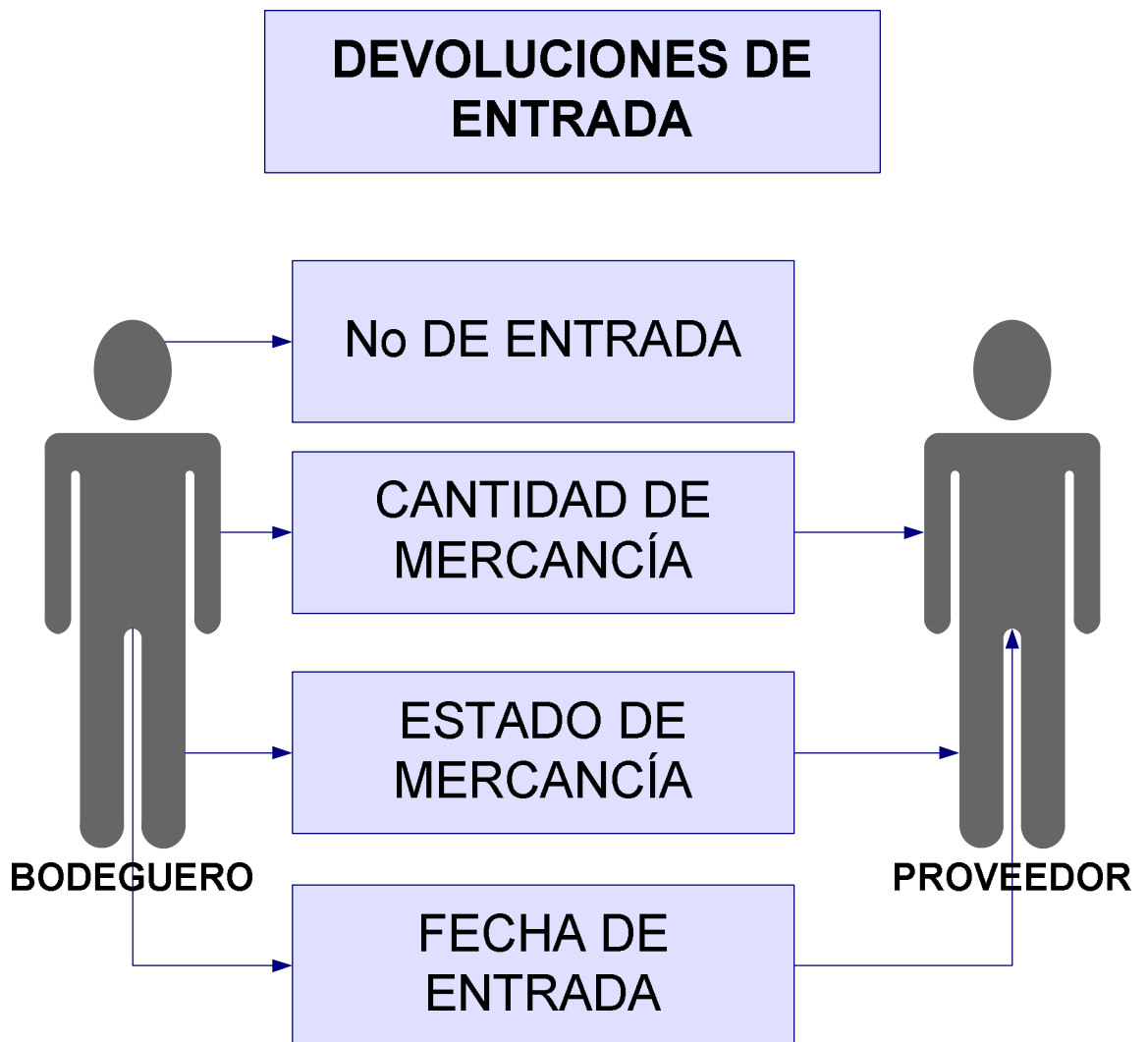


Figura 12. Devoluciones de entrada

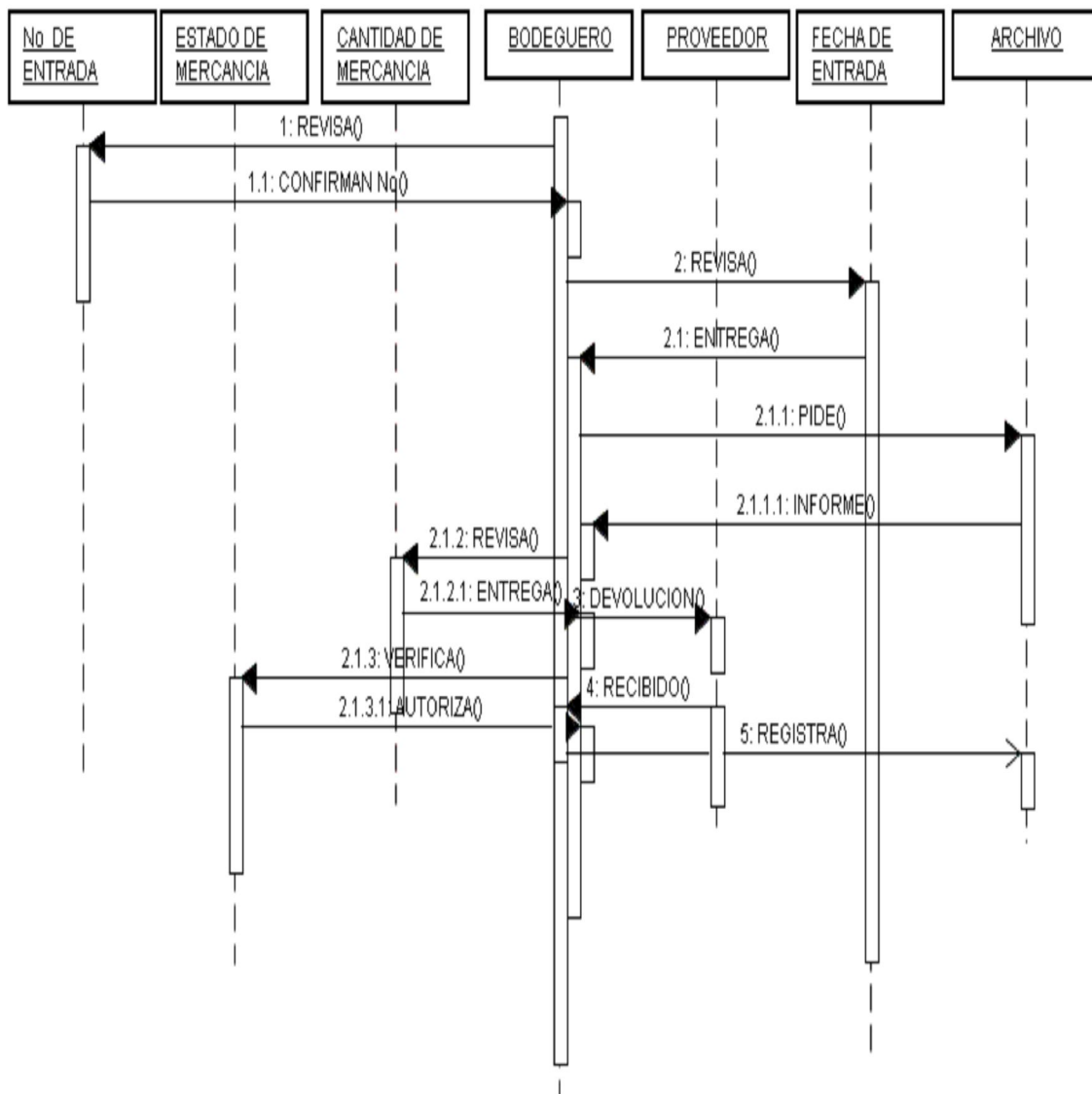


Figura 13. Diagrama de secuencia Devolución de entrada

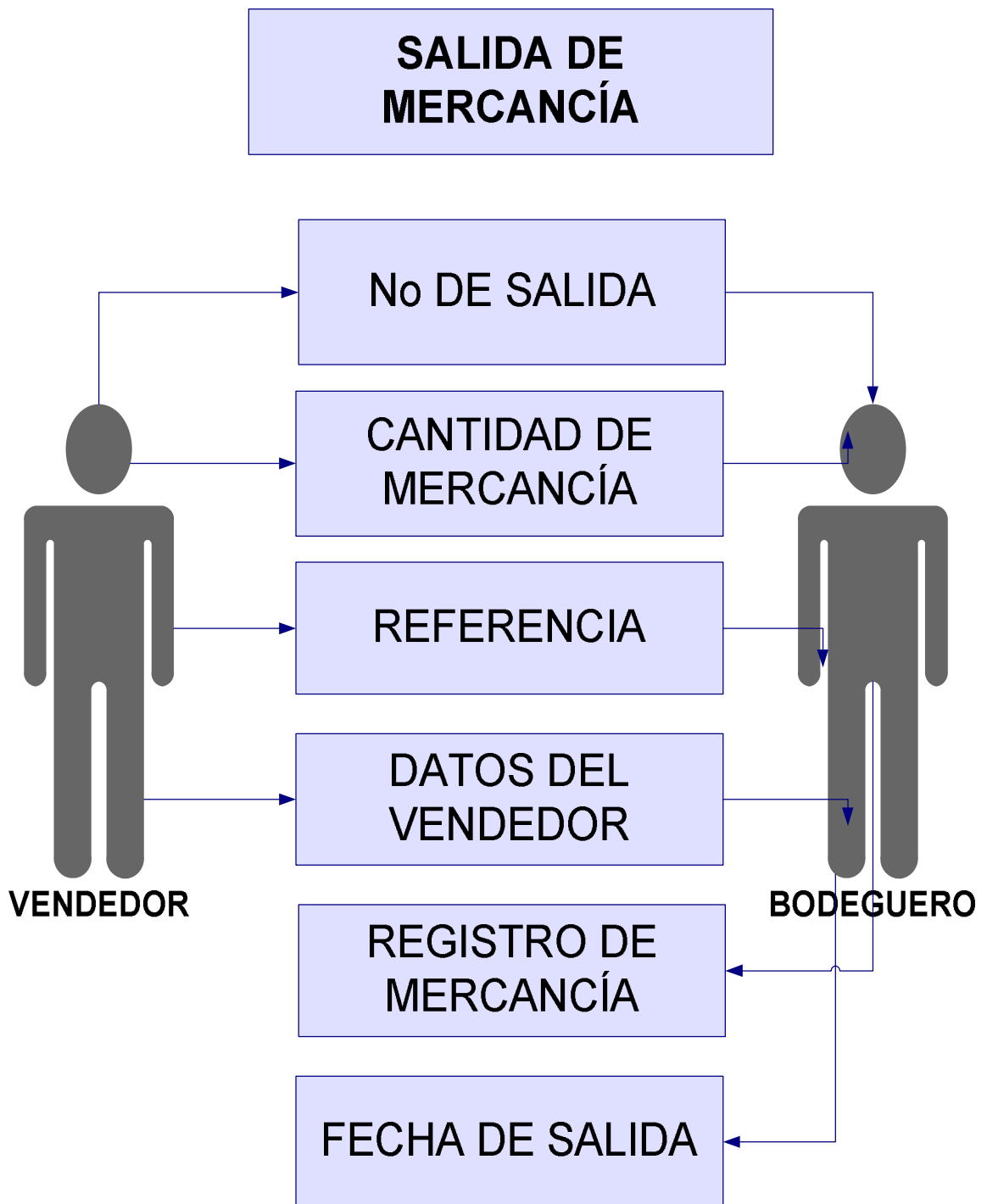


Figura 14. Salida de mercancía

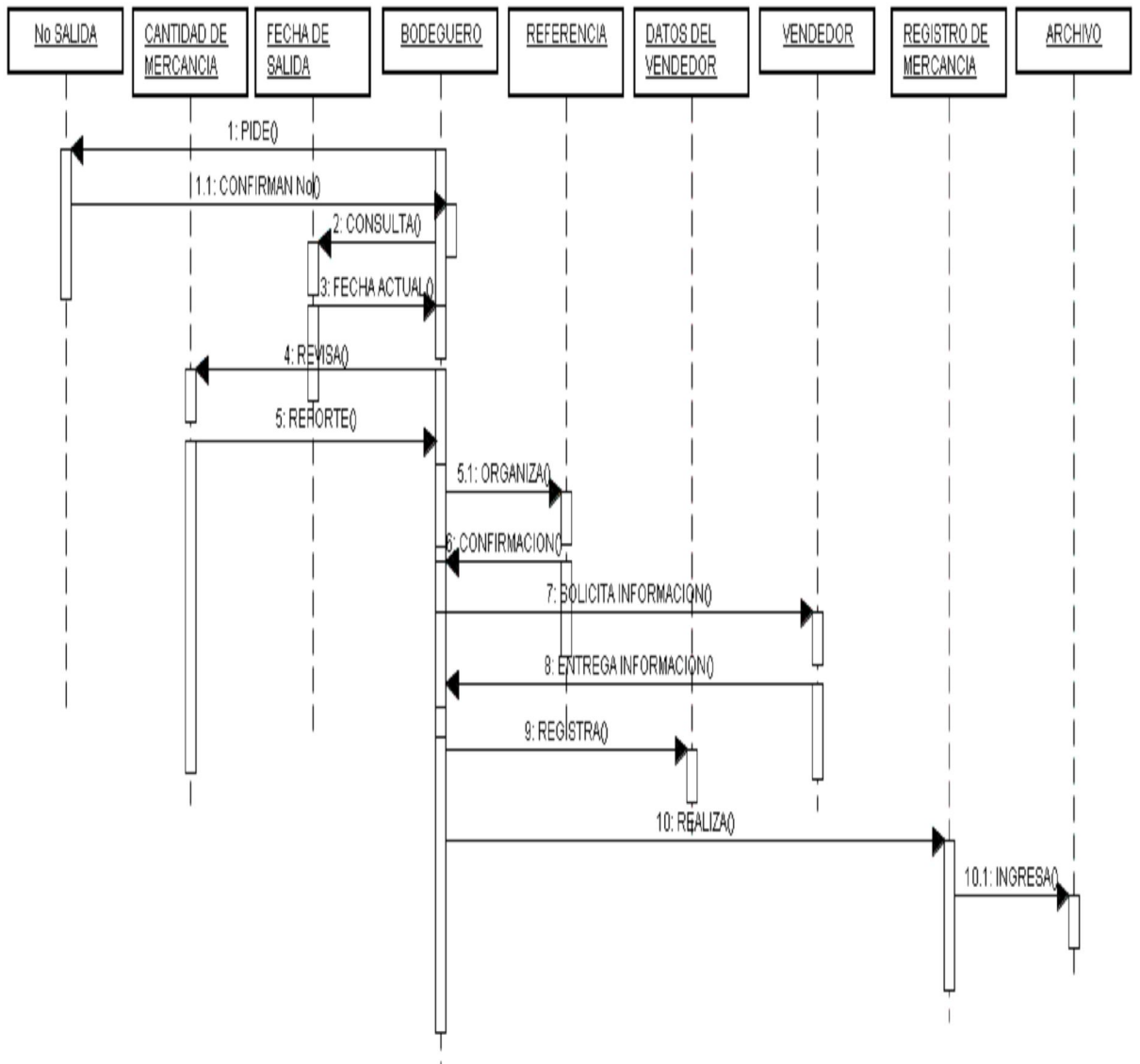


Figura 15. Diagrama de secuencia Salida

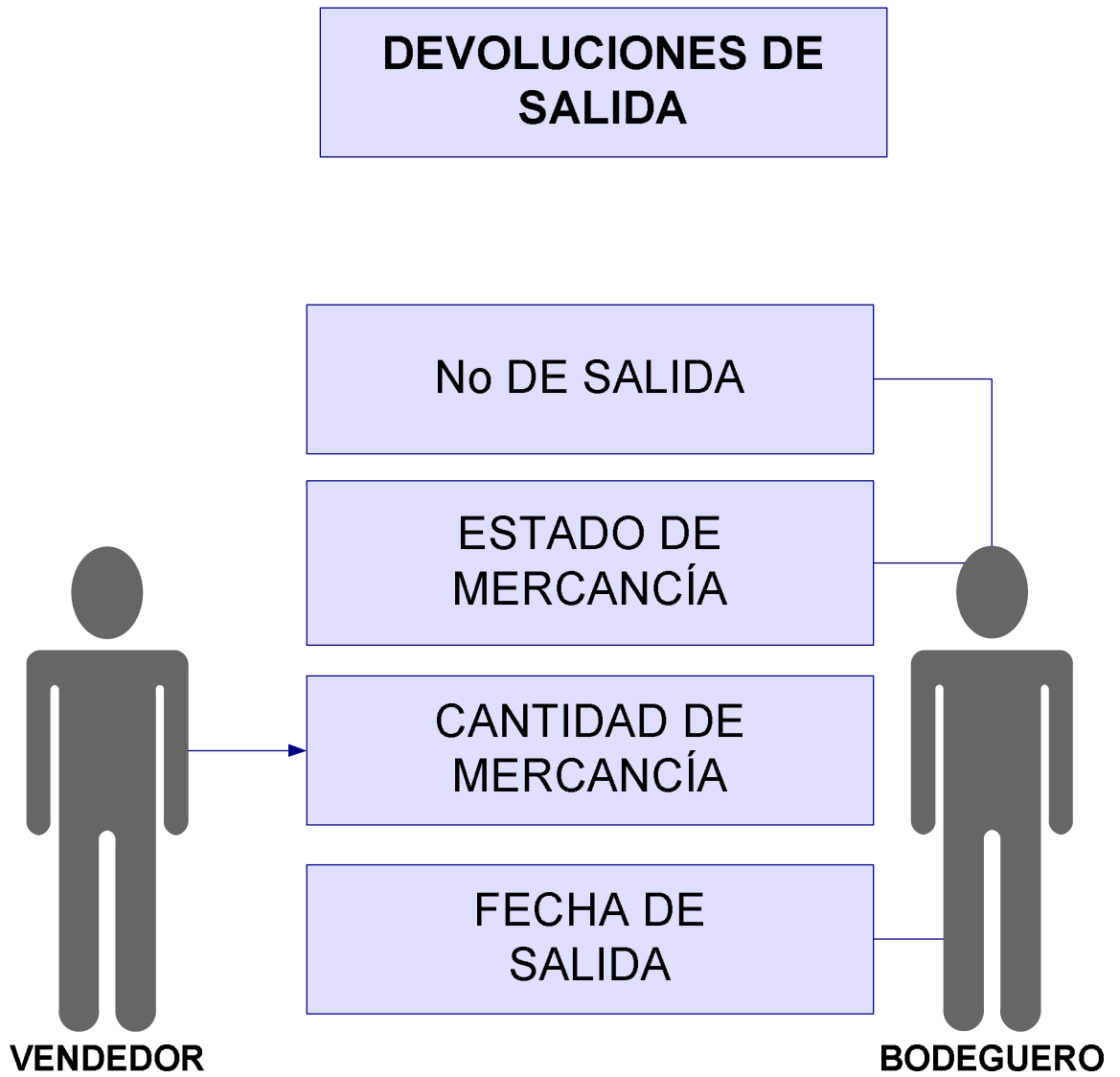


Figura 16. Devoluciones de salida

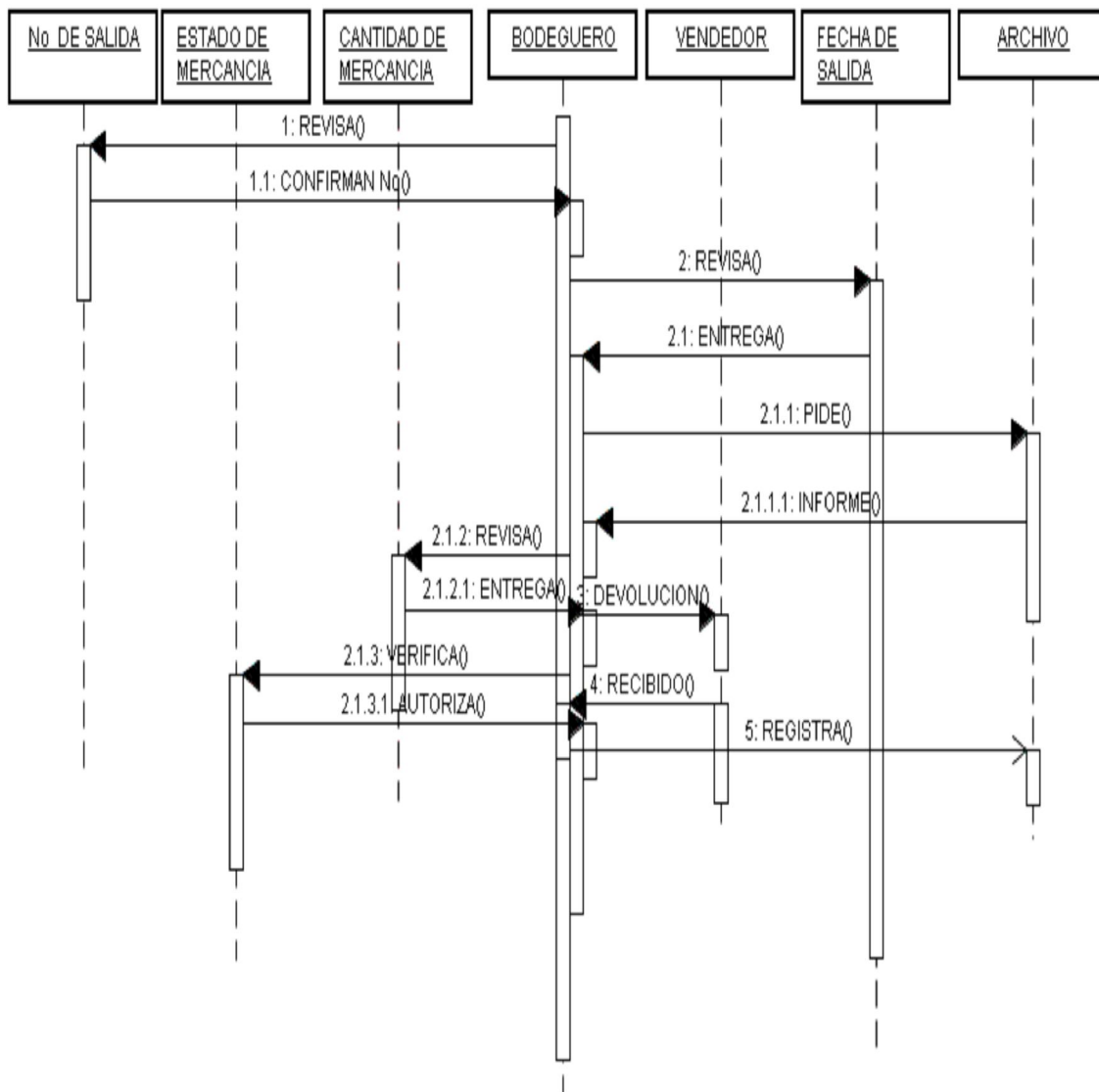


Figura 17. Diagrama de secuencia Devoluciones de salida

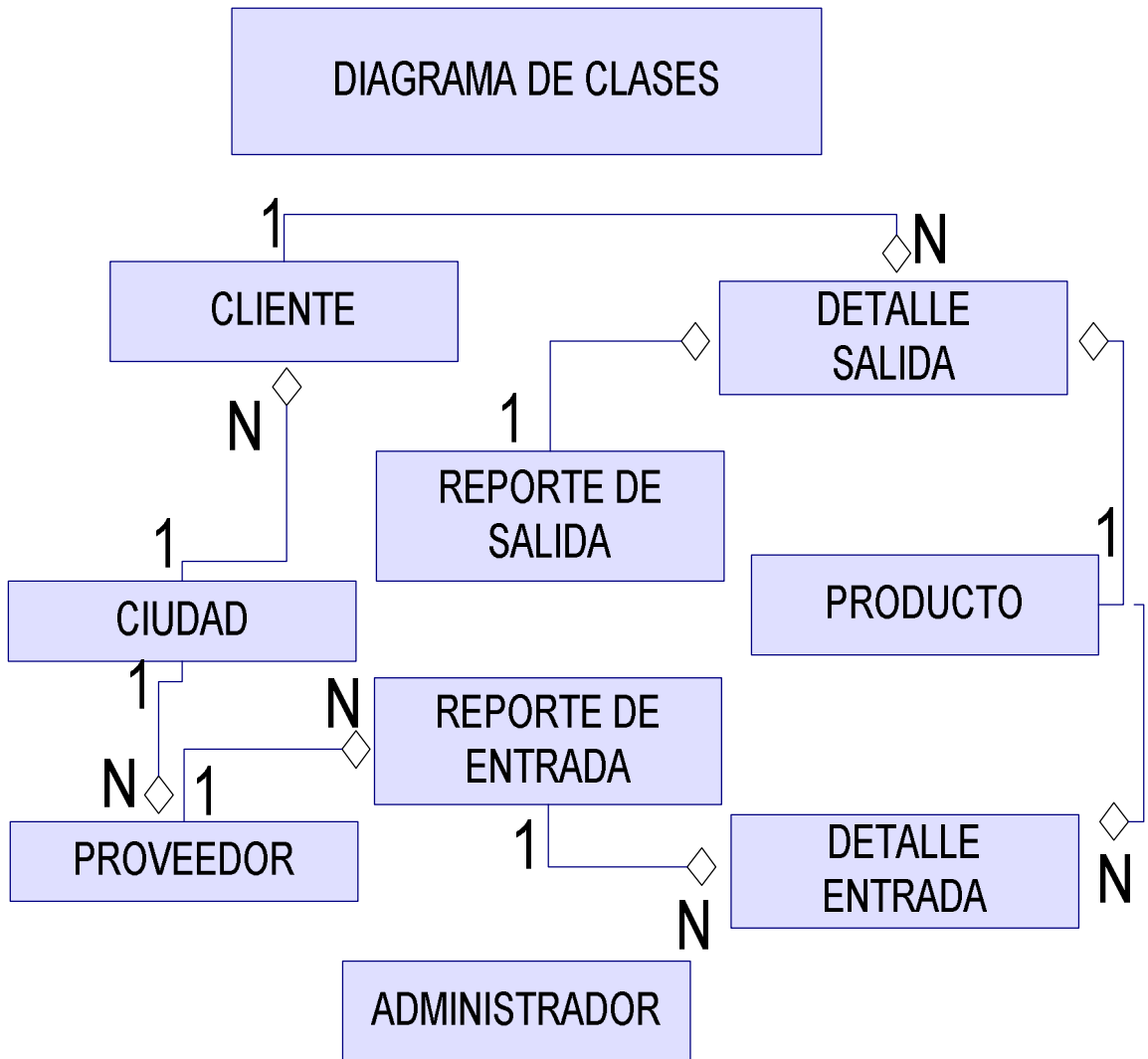


Figura 18. Diagrama de clases

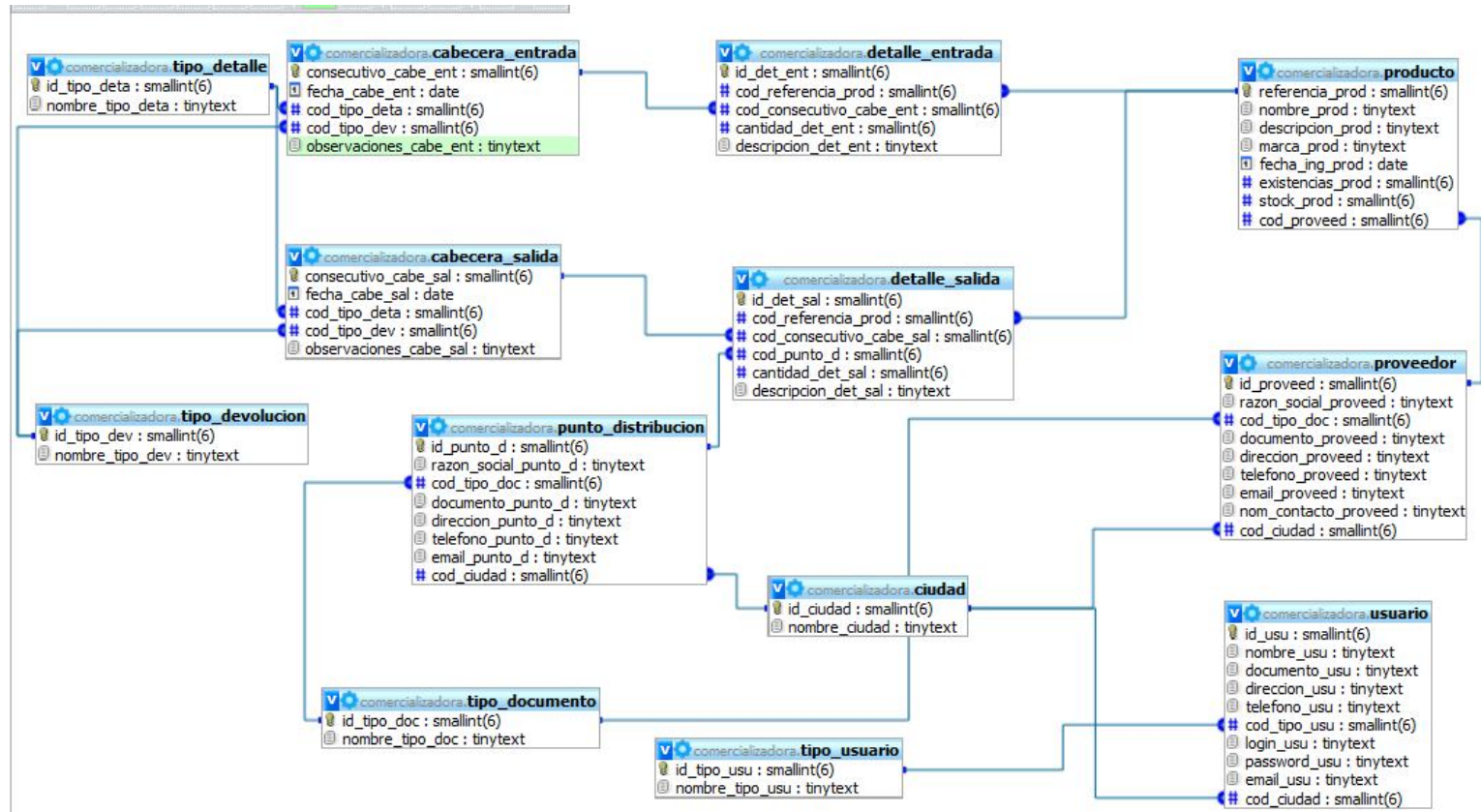


Figura 19. Modelo Entidad-Relación

8. DICCIONARIOS DE DATOS

Tabla 1. Proveedor

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|----------------------|-----------|----------|-------|----------|----------------|----------|--|
| ID_PROVEED | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA UN DATO NUMERICO EL CUAL IDENTIFICA EL PROVEEDOR |
| RAZON_SOCIAL_PROVEED | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE ALMACENA UN DATO TEXTO EL CUAL IDENTIFICA EL NOMBRE DEL PROVEEDOR |
| COD_TIPO_DOC | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL TIPO DE IDENTIFICACION DEL PROVEEDOR |
| DOCUMENTO_PROVEED | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE ALMACENA EL DOCUMENTO DEL PROVEEDOR |
| DIRECCION_PROVEED | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA LA DIRECCION DEL PROVEEDOR |
| TELEFONO_PROVEED | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA EL TELEFONO DEL PROVEEDOR |
| EMAIL_PROVEED | TEXTO | | NO | SI | NO | SI | CAMPO DE TEXTO DONDE SE ALMACENA EL CORREO ELECTRONICO DEL PROVEEDOR |
| NOM_CONTACTO_PROVEED | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA EL CONTACTO DEL PROVEEDOR |
| COD_CIUADAD | ENTERO | 6 BYTE | FK | NO | SI | SI | CAMPO NUMERICO QUE MUESTRA LA CIUDAD DONDE RESIDE EL PROVEEDOR |

Tabla 2. Cabecera entrada

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|------------------------|-----------|----------|-------|----------|----------------|----------|---|
| CONSECUTIVO_CABE_ENT | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO EL CUAL IDENTIFICA LA CABECERA DE ENTRADA |
| FECHA_CABE_ENT | FECHA | | NO | SI | SI | SI | CAMPO QUE ALMACENA UN DATO TIPO FECHA QUE CONTIENE LA FECHA DE LA CABECERA DE ENTRADA |
| COD_TIPO_DETA | ENTERO | 6 BYTES | FK | NO | SI | SI | CAMPO DE NUMERICO DONDE SE REGISTRA EL CODIGO DEL TIPO DE DETALLE |
| COD_TIPO_DEV | ENTERO | 6 BYTES | FK | NO | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DEL TIPO DE DEVOLUCION |
| OBSERVACIONES_CABE_ENT | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE ALMACENA LAS OBSERVACIONES |

Tabla 3. Cabecera de salida

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|------------------------|-----------|----------|-------|----------|----------------|----------|---|
| CONSECUTIVO_CABE_SAL | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL CONSECUTIVO DE CABECERA DE SALIDA |
| FECHA_CABE_SAL | FECHA | | NO | SI | SI | SI | CAMPO DATE QUE REGISTRA LA FECHA DE CABECERA DE SALIDA |
| COD_TIPO_DETA | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL CODIGO DEL TIPO DE DETALLE |
| COD_TIPO_DEV | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL CODIGO DEL TIPO DE DEVOLUCION |
| OBSERVACIONES_CABE_SAL | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA LAS OBSERVACIONES DE CABECERA DE SALIDA |

Tabla 4. Ciudad

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|----------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_CIUADAD | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL CODIGO DE LA CIUDAD |
| NOMBRE_CIUADAD | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA EL NOMBRE DE LA CIUDAD |

Tabla 5. Detalle entrada

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|--------------------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_DET_ENT | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL CODIGO DEL DETALLE DE ENTRADA |
| COD_REFERENCIA_PROD | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DE REFERENCIA DEL PRODUCTO |
| COD_CONSECUTIVO_CABE_ENT | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DEL CONSECUTIVO DE CABECERA DE ENTRADA |
| CANTIDAD_DET_ENT | ENTERO | 6 BYTES | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA LA CANTIDAD DE DETALLE DE ENTRADA |
| DESCRIPCION_DET_ENT | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE CAPTURA LA DESCRPCION DEL DETALLE DE ENTRADA |

Tabla 6. Detalle salida

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|--------------------------|-----------|----------|-------|----------|----------------|----------|--|
| ID_DET_SAL | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL CODIGO DEL DETALLE DE SALIDA |
| COD_REFERENCIA_PROD | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DE REFERENCIA DEL PRODUCTO |
| COD_CONSECUTIVO_CABE_SAL | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DEL CONSECUTIVO DE CABECERA DE SALIDA |
| COD_PUNTO_D | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO DONDE SE REGISTRA EL CODIGO DEL PUNTO DE DISTRIBUCION |
| CANTIDAD_DET_SAL | ENTERO | 6 BYTES | NO | SI | SI | SI | CAMPO NUMERICO QUE REGISTRA LA CANTIDAD DE DETALLE DE SALIDA |
| DESCRIPCION_DET_SAL | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE CAPTURA LA DESCRPCION DEL DETALLE DE SALIDA |

Tabla 7. Producto

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|------------------|-----------|----------|-------|----------|----------------|----------|--|
| REFERENCIA_PROD | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DE LA REFERENCIA DEL PRODUCTO |
| NOMBRE_PROD | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE ALMACENA EL NOMBRE DEL PRODUCTO |
| DESCRIPCION_PROD | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO DONDE SE REGISTRA LA DESCRIPCION DEL PRODUCTO |
| MARCA_PROD | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA LA MARCA DEL PRODUCTO |
| FECHA_ING_PROD | FECHA | | NO | SI | SI | SI | CAMPO DATE QUE CAPTURA LA FECHA DE INGRESO DEL PRODUCTO |
| EXISTENCIAS_PROD | ENTERO | 6 BYTES | NO | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA LAS EXISTENCIAS DEL PRODUCTO |
| STOCK_PROD | ENTERO | 6 BYTES | NO | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL NUMERO MINIMO DE MERCANCIA DE CADA PRODUCTO |
| COD_PROVEED | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE CAPTURA EL CODIGO DEL PROVEEDOR |

Tabla 8. Punto distribución

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|----------------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_PUNTO_D | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DE EL PUNTO DE DISTRIBUCION |
| RAZON_SOCIAL_PUNTO_D | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE CAPTURA LA RAZON SOCIAL |
| COD_TIPO_DOC | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL TIPO DE DOCUMENTO |
| DOCUMENTO_PUNTO_D | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA EL NUMERO DE DOCUMENTO |
| DIRECCION_PUNTO_D | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE CAPTURA LA DIRECCION DEL PUNTO DE DISTRIBUCION |
| TELEFONO_PUNTO_D | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA EL TELEFONO DEL PUNTO DE DISTRIBUCION |
| EMAIL_PUNTO_D | TEXTO | | NO | SI | NO | SI | CAMPO DE TEXTO QUE ALMACENA EL CORREO ELECTRONICO DEL PUNTO DE DISTRIBUCION |
| COD_CIUADAD | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE CAPTURA EL CODIGO DE LA CIUDAD |

Tabla 9. Tipo detalle

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|------------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_TIPO_DETA | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL TIPO DE DETALLE |
| NOMBRE_TIPO_DETA | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE CAPTURA EL TIPO DE DETALLE |

Tabla 10. Tipo devolución

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|-----------------|-----------|----------|-------|----------|----------------|----------|--|
| ID_TIPO_DEV | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL TIPO DE DEVOLUCION |
| NOMBRE_TIPO_DEV | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE CAPTURA EL TIPO DE DEVOLUCION |

Tabla 11. Tipo documento

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|-----------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_TIPO_DOC | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL TIPO DE DOCUMENTO |
| NOMBRE_TIPO_DOC | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE CAPTURA EL TIPO DE DOCUMENTO |

Tabla 12. Tipo usuario

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|-----------------|-----------|----------|-------|----------|----------------|----------|---|
| ID_TIPO_USU | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO QUE ALMACENA UN DATO NUMERICO DEL TIPO DE USUARIO |
| NOMBRE_TIPO_USU | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO EN EL CUAL SE CAPTURA EL TIPO DE USUARIO |

Tabla 13. Usuario

| CAMPO | TIPO_DATO | LONGITUD | CLAVE | UNICIDAD | OBLIGATORIEDAD | INDEXADO | DESCRIPCION |
|---------------|-----------|----------|-------|----------|----------------|----------|--|
| ID_USU | ENTERO | 6 BYTES | PK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA UN DATO NUMERICO EL CUAL IDENTIFICA AL USUARIO |
| NOMBRE_USU | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE ALMACENA UN DATO TEXTO EL CUAL IDENTIFICA EL NOMBRE DEL USUARIO |
| DOCUMENTO_USU | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE ALMACENA EL DOCUMENTO DEL USUARIO |
| DIRECCION_USU | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA LA DIRECCION DEL USUARIO |
| TELEFONO_USU | TEXTO | | NO | SI | SI | SI | CAMPO TEXTO QUE ALMACENA EL TELEFONO DEL USUARIO |
| COD_TIPO_USU | ENTERO | 6 BYTES | FK | SI | SI | SI | CAMPO NUMERICO QUE ALMACENA EL CODIGO DEL TIPO DE USUARIO |
| LOGIN_USU | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO DONDE SE ALMACENA EL LOGIN DEL USUARIO |
| PASSWORD_USU | TEXTO | | NO | SI | SI | SI | CAMPO DE TEXTO QUE REGISTRA EL PASSWORD DEL USUARIO |
| EMAIL_USU | TEXTO | | NO | SI | NO | SI | CAMPO DE TEXTO QUE CAPTURA EL EMAIL DEL USUARIO |
| COD_CIUADAD | ENTERO | 6 BYTE | FK | NO | SI | SI | CAMPO QUE MUESTRA LA CIUDAD DONDE RESIDE EL PROVEEDOR |

Mar 30 18:36:08 COT 2010

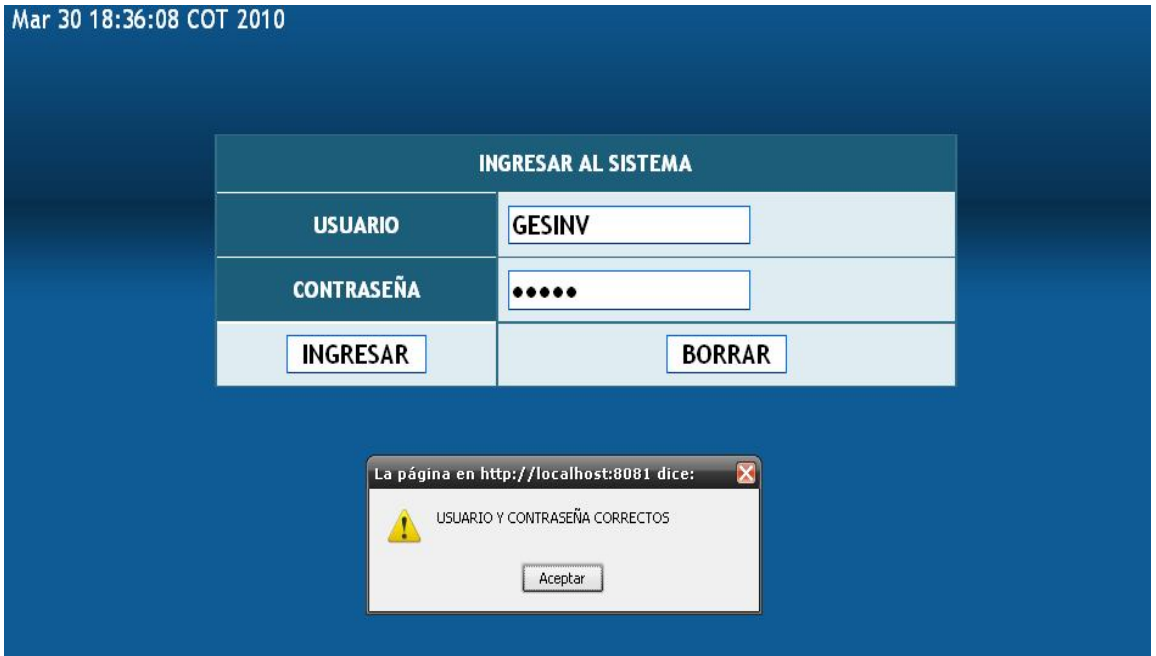


Figura 20. Ingreso a GESINV



Figura 21. Menú de GESINV

| NUEVO USUARIO | |
|--|-----------------------------------|
| NOMBRE | <input type="text"/> |
| DIRECCIÓN | <input type="text"/> |
| TELÉFONO | <input type="text"/> |
| TIPO DOCUMENTO | <input type="text" value="CC"/> |
| DOCUMENTO | <input type="text"/> |
| LOGIN | <input type="text"/> |
| PASSWORD | <input type="text"/> |
| E-MAIL | <input type="text"/> |
| CIUDAD | <input type="text" value="CALI"/> |
| <input type="button" value="GUARDAR"/> <input type="button" value="BORRAR"/> | |

Figura 22. Ingreso de Usuario

Tue Mar 30 18:23:19 COT 2010

| | |
|--|----------|
| SELECCIONA EL NOMBRE DEL USUARIO | Jonathan |
| <input type="button" value="MOSTRAR"/> | |

BUSCAR USUARIO

| | | |
|--|---|---------------------------------|
| BUSCAR USUARIO | | |
| <input type="radio"/> ID USUARIO | <input checked="" type="radio"/> NOMBRE | <input type="radio"/> DOCUMENTO |
| <input type="button" value="consultar"/> | | |

Figura 23. Buscar Usuario

Mar 30 18:30:45 COT 2010

LISTADO DE USUARIOS

| CÓDIGO | NOMBRE | DOCUMENTO | DIRECCIÓN | TELÉFONO | PERFIL | LOGIN | PASSWORD | E-MAIL | CIUDAD | |
|--------|----------|------------|------------------------------|----------|---------------|-------|----------|-------------------------|--------|--|
| 1 | Jonathan | 1032424877 | calle 35 a #45 37 este | 5793301 | Administrador | jonyy | 1234 | jon.erikfer@hotmail.com | CALI | <input type="button" value="CAMBIAR"/> |

BUSCAR USUARIO

BUSCAR USUARIO

ID USUARIO
 NOMBRE
 DOCUMENTO

Figura 24. Listar Usuario

Mar 30 18:38:03 COT 2010

MODIFICAR USUARIO

MODIFICAR USUARIO

SELECCIONE EL NOMBRE DEL USUARIO

Figura 25. Modificar Usuario (1)

| MODIFICAR USUARIO | |
|--|--|
| ID USUARIO | <input type="text" value="1"/> |
| NOMBRE | <input type="text" value="Jonathan"/> |
| DIRECCIÓN | <input type="text" value="calle 35 a # 45 37 este"/> |
| TELÉFONO | <input type="text" value="5793301"/> |
| DOCUMENTO | <input type="text" value="1032424877"/> |
| PERFIL | <input type="text" value="Administrador"/> |
| LOGIN | <input type="text" value="jonyy"/> |
| PASSWORD | <input type="password" value="••••"/> |
| E-MAIL | <input type="text" value="jon.erikfer@hotmail.com"/> |
| CIUDAD | <input type="text" value="CALI"/> |
| <input type="button" value="MODIFICAR"/> | |

Figura 26. *Modificar Usuario (2)*

Mar 30 18:41:32 COT 2010

LISTADO DE USUARIOS

| USUARIO | | | | | | | | | |
|------------|----------|------------|-------------------------|---------|---------------|-------|----------|-------------------------|--------|
| ID USUARIO | NOMBRE | DOCUMENTO | DIRECCIÓN | TÉLFONO | PERFIL | LOGIN | PASSWORD | E-MAIL | CIUDAD |
| 1 | Jonathan | 1032424877 | calle 35 a # 45 37 este | 5793301 | Administrador | jonyy | 1234 | jon.erikfer@hotmail.com | CALI |

ELIMINAR USUARIO

SELECCIONE EL NOMBRE DE USUARIO A ELIMINAR

Jonathan

La página en <http://localhost:8081> dice:

Seguro que desea eliminar el registro

Figura 27. Eliminar Usuario

Tue Mar 30 18:44:44 COT 2010

GESINV

PAGINA DE INICIO
CLIENTES
PRODUCTOS
ENTRADAS
SALIDAS

- USUARIO
- PROVEEDOR
- PUNTO DISTRIBUCIÓN
- PRODUCTO
- ENTRADA MERCANCIA
- SALIDA MERCANCIA
- REPORTE ENTRADA
- REPORTE SALIDA

Tue Mar 30 18:44:50 COT 2010

NUEVO REPORTE DE ENTRADA

| | |
|-----------------|---|
| NRO REPORTE | <input type="text" value="4"/> |
| FECHA REPORTE | <input type="text" value="2010/03/04"/> |
| TIPO DETALLE | <input type="text" value="Pedido"/> |
| TIPO DEVOLUCION | <input type="text" value="Defectuoso"/> |
| OBSERVACIONES | <div style="border: 1px solid gray; height: 30px;"></div> |

INSERTAR REPORTE ENTRADA

| Semana | Lun | Mar | Mie | Jue | Vie | Sab | Dom |
|--------|-----|-----|-----|-----|-----|-----|-----|
| 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 10 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 11 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 12 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 13 | 29 | 30 | 31 | | | | |

Hoy es Mar: 30, Mar 2010

Figura 28. Nuevo Reporte de entrada

LISTADO DE REPORTES DE ENTRADA

| NRO REPORTE | FECHA | TIPO DETALLE | TIPO DEVOLUCION | OBSERVACIONES |
|-------------|------------|--------------|-----------------|---------------------------------------|
| 1 | 2010-01-07 | Compra | Defectuoso | N/A |
| 2 | 2010-01-06 | Devoluciones | Incumplimiento | se devuelve pedido por incumplimiento |
| 3 | 2010-03-02 | Pedido | Defectuoso | |

Figura 29. Listado de Reportes de entrada



Figura 30. Modificar Reporte mercancía



Figura 31. Modificar Reporte de entrada

9. ANALISIS Y DISEÑO DE LA INTERFAZ DEL PROCESAMIENTO DE DATOS

En la clase declaramos los atributos, en el método constructor inicializamos los atributos dependiendo del tipo de dato que sea.

Por cada atributo que se declara se debe asignar dos métodos una de asignación (set) y otro de obtención (get) los cuales nos permite asignar a un método un valor que se captura por un formulario .jsp y el de obtención que nos permite traer el dato que se asigna en el método set.

En cada método de asignación capturamos el parámetro dependiendo del dato que sea, después lo asignamos al atributo de la clase y por último el atributo de tipo vector le asignamos el atributo capturado y la posición en la cual se va almacenar.

Por último retornar el dato en el método get y al final retornamos el método de obtención al cual se asigno el vector.

CLASE CONEXION

Se declara el nombre del paquete el cual va almacenar la clase clase conexión package Conectividad; se importa la librería java.sql la cual nos permite declarar las instrucciones de la clase

```
package Conectividad;

import java.sql.*;

//Declaracion de la clase conexion
public class Conexion {
    protected Connection canal; //Instanciar un objeto como canal de
    conexión
    protected Statement instruccion; //Instanciar un objeto como flujo
    de conexion

    protected String strcon; //Variable que almacena el logueo al web
    server

    public Conexion()
    {
        canal = null; //Instanciar un objeto como canal de conexión
        instruccion = null; //Instanciar un objeto como flujo de
    conexion

        strcon = "";

    } //Cierre metodo constructor

    public void Conectar()
    {
        strcon =
        "jdbc:mysql://localhost/comercializadora?user=root&password=";
        try
        {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            System.out.println("\nDriver de conexion instanciado...\n");

            canal = DriverManager.getConnection(strcon);
            System.out.println("\nCanal de conexión establecido...\n");

            instruccion =
            canal.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_U
            PDATEABLE);
            System.out.println("\nFlujo de conexión disponible...\n");
        }
        catch(java.lang.ClassNotFoundException e)
```

```

        {
            System.out.println("\nNo se encontro el driver de
conexion.\n");
        }
        catch(SQLException e)
        {
            System.out.println("\nNo fue posible establecer la
conexion.\n");
        }
        catch(java.lang.InstantiationException e)
        {
            System.out.println("\nNo fue posible instanciar la clase de
conexión.\n");
        }
        catch(java.lang.IllegalAccessException e)
        {
            System.out.println("\nSe produjo un acceso ilegal a la
clase.\n");
        }

    } //Cierre método Conectar

    public Statement getInstruccion()
    {
        return instruccion;
    } //Cierre método getInstruccion

    public Connection getCanal()
    {
        return canal;
    } //Cierre método getCanal

    public void liberarRecursos()
    {
        try
        {
            instruccion.close();
            canal.close();
            System.out.println("\nSe liberaron los recursos
utilizados....(Canal y Flujo de conexión).\n");
        } //Cierre del try
        catch(SQLException e)
        {
            System.out.println("\nNo fue posible cerrar el canal y el
flujo de conexion...\n");
        } //Cierre del catch

    } //Cierre método liberarCursor
} //Cierre de la clase conexion

```

CLASE CURSOR

Se declara el nombre del paquete el cual va almacenar la clase cursor

```
package Conectividad;
```

Se importa la librería java.sql la cual nos permite declarar el vector en la clase

```
import java.sql.*;
```

```
//Declaración de la clase cursor
```

```
public class Cursor {
    protected ResultSet tabla; //Instanciar un objeto como cursor

    public Cursor ()
    {
        tabla = null; //Instanciar un objeto como cursor
    } //Cierre método constructor

    public void Ejecutarconsulta (Statement instruccion1, String
    Consultal)
    {
        try
        {
            tabla = instruccion1.executeQuery(Consultal);
            System.out.println("\nSe ejecutó con éxito la sentencia
    SQL....\n");

        } //Cierre del try

        //Si la sentencia NO se ejecutó con éxito, capturar la
    excepción...
        catch(SQLException e)
        {
            System.out.println("\nNo se ejecutó la sentencia
    SQL....\n");

        } //Cierre del catch

    } //Cierre método Ejecutarconsulta

    public ResultSet getCursor()
    {
        return tabla;
    }
}
```

```

    }//Cierre método getCursor

    public void liberarCursor()
    {
        try
        {
            tabla.close();
            System.out.println("\nSe liberaron los recursos
utilizados....(Cursor).\n");
        }//Cierre del try
        catch(SQLException e)
        {
            System.out.println("\nNo fue posible cerrar el
cursor...\n");
        }//Cierre del catch

    }//Cierre método liberarCursor

} //Cierre clase Cursor

```

CLASE CIUDAD

```
package Persistencia;
```

```

/**
 *
 * @author JONATHAN DIAZ JHON CHAVES
 */

```

```

Se importa la librería la cual nos permite declarar el vector
import java.util.Vector;

```

```
Inicio de la clase Ciudad
```

```
public class Ciudad {
```

```
Se declaran los atributos de la clase
```

```

protected int id_ciudad;
protected String nombre;
protected Vector arrayCiudad;

```

```
Inicio del método constructor
```

```

public Ciudad()
{

```

Se inicializan los atributos de la clase Ciudad

```
id_ciudad=0;
nombre="";
arrayCiudad=new Vector();
asignarTamanoVector();
}//cierre del 53étodo constructor
protected void asignarTamanoVector()
{
    arrayCiudad.setSize(2);
}//cierre del método asignartamanovector

public void setid_Ciudad(int id_ciu)
{
    id_ciudad=id_ciu;
    arrayCiudad.setElementAt(id_ciudad, 0);
}//cierre del 53étodo id_ciudad

public int getid_Ciudad()
{
    return id_ciudad;
}//cierre del método getciudad

public void setNombre(String nom)
{
    nombre=nom;
    arrayCiudad.setElementAt(nombre,1);
}//cierre método setnombre

public String getNombre()
{
    return nombre;
}//cierre método getnombre

public Vector getCiudad()
{
    return arrayCiudad;
}//cierre método getciudad

}//cierre de la clase ciudad
```

CLASE USUARIO

```
package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JOHN CHAVES
 */
import java.util.Vector;
public class usuario {
    protected int id_usuario;
    protected String nombre;
    protected String documento;
    protected String direccion;
    protected String telefono;
    protected int cod_tipo_usuario;
    protected String login;
    protected String password;
    protected String email;
    protected int cod_ciudad;

    protected Vector arrayUsuario;
public usuario()
{
id_usuario=0;
nombre="";
documento="";
direccion="";
telefono="";
cod_tipo_usuario=0;
login="";
password="";
email="";
cod_ciudad=0;

arrayUsuario=new Vector();
asignarTamanoVector();

} //cierre del metodo constructor

protected void asignarTamanoVector()
{
arrayUsuario.setSize(10);
} //cierre del metodo asignar tamano vector

public void setid_Usuario(int id_usu)
{
    id_usuario=id_usu;
    arrayUsuario.setElementAt(id_usuario,0);
} //cierre del metodo setid_usuario
```



```

public int getid_Usuario()
{
    return id_usuario;
} // cierre del metodo getid_usuario

public void setNombre(String nom)
{
    nombre=nom;
    arrayUsuario.setElementAt(nombre,1);
} // cierre del metodo setnombre

public String getNombre()
{
    return nombre;
} // cierre del metodo getNombre

public void setDocumento(String doc)
{
    documento=doc;
    arrayUsuario.setElementAt(documento,2);
} // cierre del metodo setdocumento

public String getDocumento()
{
    return documento;
} // cierre del metodo getdocumento

public void setDireccion(String dir)
{
    direccion=dir;
    arrayUsuario.setElementAt(direccion,3);
} // cierre del metodo setdireccion

public String getDireccion()
{
    return direccion;
} // cierre del metodo getdireccion

public void setTelefono(String tel)
{
    telefono=tel;
    arrayUsuario.setElementAt(telefono,4);
} // cierre del metodo settelefono

public String getTelefono()
{
    return telefono;
} // cierre del metodo gettelefono

public void setcod_tipo_Usuario(int cod_usu)
{

```

```

        cod_tipo_usuario=cod_usu;
        arrayUsuario.setElementAt(cod_tipo_usuario,5);
    }//cierre del metodo setcod_tipo_usuario

    public int getcod_tipo_Usuario()
    {
        return cod_tipo_usuario;
    }//cierre del metodo getcod_tipo_usuario

    public void setLogin(String log)
    {
        login=log;
        arrayUsuario.setElementAt(login,6);
    }//cierre del metodo setlogin

    public String getLogin()
    {
        return login;
    }//cierre del metodo getlogin

    public void setPassword (String passw)
    {
        password=passw;
        arrayUsuario.setElementAt(password,7);
    }//cierre del metodo setpassword

    public String getPassword()
    {
        return password;
    }//cierre del metodo getpassword

    public void setEmail (String mail)
    {
        email=mail;
        arrayUsuario.setElementAt(email,8);
    }//cierre del metodo setemail

    public String getEmail()
    {
        return email;
    }//cierre del metodo getemail

    public void setcod_Ciudad(int cod_ciud)
    {
        cod_ciudad=cod_ciud;
        arrayUsuario.setElementAt(cod_ciudad,9);
    }//cierre del metodo setcod_ciudad

    public int getcod_Ciudad()
    {
        return cod_ciudad;
    }//cierre del metodo getcod_ciudad

```

```

public Vector getUsuario()
{
    return arrayUsuario;
} // cierre del metodo getusuario

} // cierre de la clase usuario

```

CLASE TIPO USUARIO

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JON CHAVES
 */
import java.util.Vector;
public class tipo_usuario {
    protected int id_tipo_usuario;
    protected String nombre;

    protected Vector arraytipo_Usuario;

    public tipo_usuario()
    {
        id_tipo_usuario=0;
        nombre="";
        arraytipo_Usuario=new Vector();
        asignarTamanoVector();
    } // cierre del metodo constructor
    protected void asignarTamanoVector()
    {
        arraytipo_Usuario.setSize(3);
    } // cierre del metodo asignar tamano vector
    public void setid_tipo_Usuario(int id_tipo_u)
    {
        id_tipo_usuario=id_tipo_u;
        arraytipo_Usuario.setElementAt(id_tipo_usuario,0);
    } // cierre del metodo setidtipoUsuario
    public int getid_tipo_Usuario(){
        return id_tipo_usuario;
    } // cierre del metodo getidtipousuario
    public void setNombre(String nom)
    {
        nombre=nom;
        arraytipo_Usuario.setElementAt(nombre,1);
    } // cierre metodo sernombre
    public String getNombre()
    {

```

```

        return nombre;
    }//cierre del metodo getnombre

    public Vector gettipo_Usuario()
    {
        return arraytipo_Usuario;
    }//cierre del metodo gettipoUsuario
}//cierre de la clase tipo_usuario

```

CLASE TIPO DOCUMENTO

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JON CHAVES
 */
import java.util.Vector;
public class Tipodocumento {
    protected int id_tipo_documento;
    protected String nombre;

    protected Vector arraytipo_Documento;

    public Tipodocumento()
    {
        id_tipo_documento=0;
        nombre="";
        arraytipo_Documento=new Vector();
        asignarTamanoVector();
    }//cierre del metodo constructor
    protected void asignarTamanoVector()
    {
        arraytipo_Documento.setSize(3);
    }//cierre del metodo asignar tamano vector
    public void setid_tipo_Documento(int id_tipo_d)
    {
        id_tipo_documento=id_tipo_d;
        arraytipo_Documento.setElementAt(id_tipo_documento,0);
    }//cierre del metodo setidtipodocumento
    public int getid_tipo_Documento(){
        return id_tipo_documento;
    }//cierre del metodo getidtipodocumento
    public void setNombre(String nom)

```

```

    {
        nombre=nom;
        arraytipo_Documento.setElementAt(nombre,1);
    }//cierre metodo setnombre
    public String getNombre()
    {
        return nombre;
    }//cierre del metodo getnombre

    public Vector gettipo_Documento()
    {
        return arraytipo_Documento;
    }//cierre del metodo gettipodocumento
} //cierre de la clase tipo_documento

```

CLASE TIPO DEVOLUCION

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JON CHAVES
 */
import java.util.Vector;
public class Tipodevolucion{
    protected int id_tipo_dev;
    protected String nombre;

    protected Vector arraytipo_Devolucion;

    public Tipodevolucion()
    {
        id_tipo_dev=0;
        nombre="";
        arraytipo_Devolucion=new Vector();
        asignarTamanioVector();
    } //cierre del metodo constructor
    protected void asignarTamanioVector()
    {
        arraytipo_Devolucion.setSize(2);
    } //cierre del metodo asignar tamanio vector
    public void setid_tipo_Devolucion(int id_tipo_d)
    {
        id_tipo_dev=id_tipo_d;
        arraytipo_Devolucion.setElementAt(id_tipo_dev,0);
    } //cierre del metodo setidtipodocumento
    public int getid_tipo_Devoluciion(){
        return id_tipo_dev;
    } //cierre del metodo getidtipodevolucion

```

```

public void setNombre(String nom)
{
    nombre=nom;
    arraytipo_Devolucion.setElementAt(nombre,1);
} //cierre metodo setnombre
public String getNombre()
{
    return nombre;
} //cierre del metodo getnombre

public Vector gettipo_Devolucion()
{
    return arraytipo_Devolucion;
} //cierre del metodo gettipodevolucion
} //cierre de la clase tipodevolucion

```

TIPO DETALLE

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JON CHAVES
 */
import java.util.Vector;
public class Tipodetalle{
    protected int id_tipo_deta;
    protected String nombre;

    protected Vector arraytipo_Detalle;

    public Tipodetalle()
    {
        id_tipo_deta=0;
        nombre="";
        arraytipo_Detalle=new Vector();
        asignarTamanioVector();
    } //cierre del metodo constructor
    protected void asignarTamanioVector()
    {
        arraytipo_Detalle.setSize(2);
    } //cierre del metodo asignar tamanio vector
    public void setid_tipo_Detalle(int id_tipo_d)
    {
        id_tipo_deta=id_tipo_d;
        arraytipo_Detalle.setElementAt(id_tipo_deta,0);
    } //cierre del metodo setidtipodocumento
    public int getid_tipo_Detalle(){

```

```

        return id_tipo_deta;
    } // cierre del metodo getidtipodocumento
    public void setNombre(String nom)
    {
        nombre=nom;
        arraytipo_Detalle.setElementAt(nombre,1);
    } // cierre metodo setnombre
    public String getNombre()
    {
        return nombre;
    } // cierre del metodo getnombre

    public Vector gettipo_Detalle()
    {
        return arraytipo_Detalle;
    } // cierre del metodo gettipodocumento
} // cierre de la clase tipo_documento

```

CLASE PROVEEDOR

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ AND JOHN CHAVES
 */

import java.util.Vector;
public class Proveedor {
    protected int id_proveedor;
    protected String razon_social;
    protected int cod_tipo_documento;
    protected String documento;
    protected String direccion;
    protected String telefono;
    protected String email;
    protected String contacto;
    protected int cod_ciudad;
}

```

```

        protected Vector arrayProveedor;
public Proveedor()
{
    id_proveedor=0;
    razon_social="";
    cod_tipo_documento=0;
    documento="";
    direccion="";
    telefono="";
    email="";
    contacto="";
    cod_ciudad=0;

    arrayProveedor=new Vector();
    asignarTamanoVector();
} //cierre metodo constructor

protected void asignarTamanoVector()
{
    arrayProveedor.setSize(10);
} //cierre metodo asignar tamano vector

public void setid_Proveedor(int id_prove)
{
    id_proveedor=id_prove;
    arrayProveedor.setElementAt(id_proveedor,0);
} //cierre del metodo setid_proveedor
public int getid_Proveedor()
{
    return id_proveedor;
} //cierre metodo getid_proveedor

public void setrazon_Social(String razon_s)
{
    razon_social=razon_s;
    arrayProveedor.setElementAt(razon_social,1);
} //cierre metodo setrazon_social
public String getrazon_Social()
{
    return razon_social;
} //cierren metodo getrazon_social
public void setcod_tipo_Documento(int cod_tipo_d)
{
    cod_tipo_documento=cod_tipo_d;
    arrayProveedor.setElementAt(cod_tipo_documento,2);
} //cierre metodo setcod_tipo_documento

public int getcod_tipo_Documento()
{
    return cod_tipo_documento;
} //cierre metodo getcod_tipo_documento

```



```

public void setDocumento(String doc)
{
    documento=doc;
    arrayProveedor.setElementAt(documento,3);

} //Cierre metodo setdocumento
public String getDocumento()
{
    return documento;
} //cierre metodo getdocumento

public void setDireccion(String dir)
{
    direccion=dir;
    arrayProveedor.setElementAt( direccion,4);
} //cierre metodo setdireccion

public String getDireccion()
{
    return direccion;
} //cierre metodo getdireccion
public void setTelefono(String tel)
{
    telefono=tel;
    arrayProveedor.setElementAt(telefono,5);
} //cierre metodo settelefono
public String getTelefono()
{
    return telefono;
} //cierre metodo gettelefono
public void setEmail(String ema)
{
    email=ema;
    arrayProveedor.setElementAt(email,6);
} //cierre metodo setemail
public String getEmail()
{
    return email;
} //cierre metodo getemail
public void setContacto(String cont)
{
    contacto=cont;
    arrayProveedor.setElementAt(contacto,7);
} //cierre metodo setcontacto
public String getContacto()
{
    return contacto;
} //cierre metodo getcontacto
public void setcod_Ciudad(int cod_ciu)
{
    cod_ciudad=cod_ciu;
    arrayProveedor.setElementAt(cod_ciudad, 8);
}

```

```

} // cierre metodo setcod_ciudad
public int getcod_Ciudad()
{
    return cod_ciudad;
} // cierre metodo getcod_ciudad

public Vector getProveedor()
{
    return arrayProveedor;
} // cierre metodo getproveedor

} // cierre de la clase proveedor

```

CLASE PRODUCTO

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ JOHN CHAVES
 */
import java.sql.Date;
import java.util.Vector;
public class Producto {
protected int referencia_producto;
protected String nombre;
protected String descripcion;
protected String marca;
protected String fecha;
protected int existencias;
protected int stocks;
protected int cod_proveedor;

protected Vector arrayProducto;
public Producto(){
    referencia_producto=0;
    nombre="";
    descripcion="";
    marca="";
    fecha="";

```

```

    existencias=0;
    stocks=0;
    cod_proveedor=0;

    arrayProducto=new Vector();
    asignarTamanoVector();

} // cierre del metodo constructor
protected void asignarTamanoVector()
{
    arrayProducto.setSize(9);
} // cierre metodo asignartamanovector
public void setreferencia_Producto(int referencia_pro)
{
    referencia_producto=referencia_pro;
    arrayProducto.setElementAt(referencia_producto, 0);
} // cierre metodo setreferencia_producto
public int getreferencia_Producto()
{
    return referencia_producto;
} // cierre metodo getreferencia_producto
public void setNombre(String nom)
{
    nombre=nom;
    arrayProducto.setElementAt(nombre, 1);
} // cierre del metodo setnombre
public String getNombre()
{
    return nombre;
} // cierre del metodo getnombre
public void setDescription(String des)
{
    descripcion=des;
    arrayProducto.setElementAt(descripcion, 2);
} // cierre del metodo setdescripcion
public String getDescription(){
    return descripcion;
} // cierre del metodo getdescripcion
public void setMarca(String mar)
{
    marca=mar;
    arrayProducto.setElementAt(marca, 3);
} // cierre metodo setmarca
public String getMarca()
{
    return marca;
} // cierre metodo getmarca
public void setFecha(String fec)
{
    fecha=fec;
    arrayProducto.setElementAt(fecha, 4);
} // cierre del metodo setfecha

```

```

public String getFecha()
{
    return fecha;
} //cierre del metodo getfecha
public void setExistencias(int exis)
{
    existencias=exis;
    arrayProducto.setElementAt(existencias, 5);
} //cierre metodo existencias
public int getExistencias()
{
    return existencias;
} //cierre del metodo getexistencias
public void setStocks(int sto)
{
    stocks=sto;
    arrayProducto.setElementAt(stocks,6);
} //cierre del metodo serstocks
public int getStocks()
{
    return stocks;
} //cierre del metodo getstocks

public void setcod_Proveedor(int cod_pro)
{
    cod_proveedor=cod_pro;
    arrayProducto.setElementAt(cod_proveedor, 7);
} //cierre del metodo sercod_proveedor

public int getcod_Proveedor()
{
    return cod_proveedor;
} //cierre del metodo getproveedor
public Vector getProducto()
{
    return arrayProducto;
}

} //cierre de la clase producto

```

CLASE PUNTO DISTRIBUCION

```
package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JOHN CHAVES
 */

import java.util.Vector;
public class Puntodistribucion {
    protected int id_punto;
    protected String razon_social;
    protected int cod_tipo_documento;
    protected String documento;
    protected String direccion;
    protected String telefono;
    protected String email;
    protected int cod_ciudad;

    protected Vector arrayPunto_distribucion;
public Puntodistribucion()
{
    id_punto=0;
    razon_social="";
    cod_tipo_documento=0;
    documento="";
    direccion="";
    telefono="";
    email="";
    cod_ciudad=0;

    arrayPunto_distribucion=new Vector();
    asignarTamanoVector();
} //cierre del metodo constructor punto distribucion

protected void asignarTamanoVector()
{
    arrayPunto_distribucion.setSize(9);
} //cierre del metodo asignar tamano vector

public void setid_Punto (int id_punt)
{
    id_punto=id_punt;
    arrayPunto_distribucion.setElementAt(id_punto,0);
} //cierre del metodo setid_punto
public int getid_Punto()
{
```

```

        return id_punto;
    } // cierre del metodo getid _punto

    public void setrazon_social (String razon_soc)
    {
        razon_social=razon_soc;
        arrayPunto_distribucion.setElementAt(razon_social,1);
    } // cierre del metodo setrazon_social
    public String getrazon_social ()
    {
        return razon_social;
    } // cierre del metodo getrazon_social

    public void setcod_tipo_documento (int cod_doc)
    {
        cod_tipo_documento=cod_doc;
        arrayPunto_distribucion.setElementAt(cod_tipo_documento,2);
    } // cierre del metodo setcod_tipo_documento
    public int getcod_tipo_documento()
    {
        return cod_tipo_documento;
    } // cierre del metodo getcod_tipo_documento

    public void setDocumento(String doc)
    {
        documento=doc;
        arrayPunto_distribucion.setElementAt(documento,3);
    } // cierre del metodo documento
    public String getDocumento()
    {
        return documento;
    } // cierre del metodo getdocumento

    public void setDireccion (String dir)
    {
        direccion=dir;
        arrayPunto_distribucion.setElementAt(direccion,4);
    } // cierre del metodo setdireccion
    public String getDireccion()
    {
        return direccion;
    } // cierre del metodo getdireccion

    public void setTelefono(String tel)
    {
        telefono=tel;
        arrayPunto_distribucion.setElementAt(telefono,5);
    } // cierre del metodo settelefono
    public String getTelefono()
    {
        return telefono;
    } // cierre del metodo getTelefono

```

```

public void setEmail (String mail)
{
    email=mail;
    arrayPunto_distribucion.setElementAt(email,6);
} //cierre del metodo setemail
public String getEmail()
{
    return email;
} //cierre del metodo getemail

public void setcod_Ciudad (int cod_ciud)
{
    cod_ciudad=cod_ciud;
    arrayPunto_distribucion.setElementAt(cod_ciudad,7);
} //cierre del metodo setcod_ciudad
public int getcod_Ciudad()
{
    return cod_ciudad;
} //cierre del metodo getcod_ciudad

public Vector getPunto_distribucion()
{
    return arrayPunto_distribucion;
} //cierre del metodo getpunto_distribucion

} //cierre de la clase punto distribuidor

```

CLASE FECHA

```

package Persistencia;

import java.util.Date;
import java.util.Vector;

/**
 *
 * @author JONATHAN
 */
public class Fecha {
    protected Date fecha;
    protected Vector arrayFecha;
    public Fecha()
    {
        fecha=new Date();
        arrayFecha=new Vector();
        asignarTamanoVector();
    }

```

```

} // cierre metodo constructor
public void asignarTamanoVector()
{
    arrayFecha.setSize(1);
} // cierre metodo
public void setFecha1(Date fec)
{
    fecha=fec;
    arrayFecha.setElementAt(fecha,0);
} // cierre del metodo fecha
public Date getFecha1()
{
    return fecha;
} // cierre del metodo getfecha
public Vector getFecha()
{
    return arrayFecha;
} // cierre del meodo getfecha

} // cierre de la clase fecha

```

CABECERA DE ENTRADA

```

package Persistencia;

/**
 *
 * @author JONATHAN
 */
import java.util.Date;
import java.util.Vector;
public class cabecera_entrada {
    protected int consecutivo;
    protected String fecha;
    protected int cod_tipo_deta;
    protected int cod_tipo_dev;
    protected String observaciones;

    protected Vector arraycabecera_Entrada;

public cabecera_entrada(){
    consecutivo=0;
    fecha="";
    cod_tipo_deta=0;

```



```

        cod_tipo_dev=0;
        observaciones="";
        arraycabecera_Entrada=new Vector();
        asignarTamanoVector();
    }//cierre del metodo constructor
    protected void asignarTamanoVector()
    {
        arraycabecera_Entrada.setSize(6);
    }//cierre metodo asignar tamano vector
    public void setConsecutivo(int conse)
    {
        consecutivo=conse;
        arraycabecera_Entrada.setElementAt(consecutivo,0);
    }//cierre del metodo setconsecutivo
    public int getConsecutivo()
    {
        return consecutivo;
    }//cierre del metodo getconsecutivo
    public void setFecha(String fech)
    {
        fecha=fech;
        arraycabecera_Entrada.setElementAt(fecha, 1);
    }//cierre del metodo setfecha
    public String getFecha()
    {
        return fecha;
    }//cierre del metodo getfecha
    public void setcod_tipo_Detalle(int cod_tipo_de)
    {
        cod_tipo_deta=cod_tipo_de;
        arraycabecera_Entrada.setElementAt(cod_tipo_deta,2);
    }//cierre del metodo cod_tipo_Detalla
    public int getcod_tipo_Detalle()
    {
        return cod_tipo_deta;
    }//cierre del metodo cod_tipo_detalle
    public void setcod_tipo_Devolucion(int cod_tipo_de)
    {
        cod_tipo_dev=cod_tipo_de;
        arraycabecera_Entrada.setElementAt(cod_tipo_dev, 3);
    }//cierre del metodo cod_tipo_devolucion
    public int getcod_tipo_Devolucion()
    {
        return cod_tipo_dev;
    }//cierre del metodo getcod_tipo_devolucion
    public void setObservaciones(String observa)
    {
        observaciones=observa;
        arraycabecera_Entrada.setElementAt(observaciones, 4);
    }//cierre del metodoserobservaciones
    public String getObservaciones(){
        return observaciones;
    }

```

```

} // cierre del metodo getobservaciones
public Vector getcabecera_Entrada(){
    return arraycabecera_Entrada;
} // cierre del metodo getcabecera_entrada
} // cierre de la clase cabecera_entrada

```

CABECERA DE SALIDA

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Persistencia;

import java.util.Date;
import java.util.Vector;

/**
 *
 * @author JONATHAN DIAZ & JOHN CHAVES
 */
public class cabecera_salida{
    protected int consecutivo;
    protected String fecha;
    protected int cod_tipo_deta;
    protected int cod_tipo_dev;
    protected String observaciones;

    protected Vector arraycabecera_Salida;

    public cabecera_salida(){
        consecutivo=0;
        fecha="";
        cod_tipo_deta=0;
        cod_tipo_dev=0;
        observaciones="";
    }

```

```

        arraycabecera_Salida=new Vector();
        asignarTamanoVector();
    }//cierre del metodo constructor
    protected void asignarTamanoVector()
    {
        arraycabecera_Salida.setSize(5);
    }//cierre metodo asignar tamano vector
    public void setConsecutivo(int conse)
    {
        consecutivo=conse;
        arraycabecera_Salida.setElementAt(consecutivo,0);
    }//cierre del metodo setconsecutivo
    public int getConsecutivo()
    {
        return consecutivo;
    }//cierre del metodo getconsecutivo
    public void setFecha(String fech)
    {
        fecha=fech;
        arraycabecera_Salida.setElementAt(fecha, 1);
    }//cierre del metodo setfecha
    public String getFecha()
    {
        return fecha;
    }//cierre del metodo getfecha
    public void setcod_tipo_Detalle(int cod_tipo_de)
    {
        cod_tipo_deta=cod_tipo_de;
        arraycabecera_Salida.setElementAt(cod_tipo_deta,2);
    }//cierre del metodo cod_tipo_Detalla
    public int getcod_tipo_Detalle()
    {
        return cod_tipo_deta;
    }//cierre del metodo cod_tipo_detalle
    public void setcod_tipo_Devolucion(int cod_tipo_de)
    {
        cod_tipo_dev=cod_tipo_de;
        arraycabecera_Salida.setElementAt(cod_tipo_dev, 3);
    }//cierre del metodo cod_tipo_devolucion
    public int getcod_tipo_Devolucion()
    {
        return cod_tipo_dev;
    }//cierre del metodo getcod_tipo_devolucion
    public void setObservaciones(String observa)
    {
        observaciones=observa;
        arraycabecera_Salida.setElementAt(observaciones, 4);
    }//cierre del metodoserobservaciones
    public String getObservaciones(){
        return observaciones;
    }//cierre del metodo getobservaciones
    public Vector getcabecera_Salida(){

```

```

        return arraycabecera_Salida;
    }//cierre del metodo getcabecera_entrada
} //cierre de la clase cabecera_entrada

```

CLASE DETALLE ENTRADA

```

package Persistencia;

/**
 *
 * @author JONATHAN DIAZ & JOHN CHAVES
 */
import java.util.Vector;
public class Detalle_entrada {
    protected int id_det;
    protected int cod_referencia_prod;
    protected int cod_consecutivo_cabe_ent;
    protected int cantidad;
    protected String descripcion;

    protected Vector arrayDetalle_entrada;

    public Detalle_entrada()
    {
        id_det=0;
        cod_referencia_prod=0;
        cod_consecutivo_cabe_ent=0;
        cantidad=0;
        descripcion="";

        arrayDetalle_entrada=new Vector();
        asignarTamanoVector();
    } //cierre del metodo constructor del detalle entrada

    protected void asignarTamanoVector()
    {
        arrayDetalle_entrada.setSize(6);
    } //cierre del metodo asignar tamano vector

    public void setid_Det(int id_d)

```

```

{
    id_det=id_d;
    arrayDetalle_entrada.setElementAt(id_det,0);
} // cierre del metodo setid_set
public int getid_Det()
{
    return id_det;
} // cierre del metodo getid_det

public void setcod_referencia_Prod(int cod_prod)
{
    cod_referencia_prod=cod_prod;
    arrayDetalle_entrada.setElementAt(cod_referencia_prod,1);
} // cierre del metodo set_cod_referencia_prod
public int getcod_referencia_Prod()
{
    return cod_referencia_prod;
} // cierre del metodo getcod_referencia_prod

public void setcod_consecutivo_cabe_Ent(int cod_cabe_ent)
{
    cod_consecutivo_cabe_ent=cod_cabe_ent;
    arrayDetalle_entrada.setElementAt(cod_consecutivo_cabe_ent,2);
} // cierre del metodo setcod_consecutivo_cabe_ent
public int getcod_consecutivo_cabe_Ent()
{
    return cod_consecutivo_cabe_ent;
} // cierre del metodo getcod_consecutivo_cabe_ent

public void setCantidad(int cant)
{
    cantidad=cant;
    arrayDetalle_entrada.setElementAt(cantidad,3);
} // cierre del metodo setcantidad_det_ent
public int getCantidad()
{
    return cantidad;
} // cierre del metodo getcantidad_det_ent

public void setDescription(String desc)
{
    descripcion=desc;
    arrayDetalle_entrada.setElementAt(descripcion,4);
} // cierre del metodo setdescripcion
public String getDescription()
{
    return descripcion;
} // cierre del metodo detdescripcion

public Vector getDetalle_entrada()
{
    return arrayDetalle_entrada;
}

```

```
}//cierre del metodo getdetalle entrada  
}  
}//cierre de la clase detalle entrada
```

CLASE DETALLE DE SALIDA

```
package Persistencia;  
  
/**  
 *  
 * @author JONATHAN DIAZ & JOHN CHAVES  
 */  
import java.util.Vector;  
public class Detalle_salida {  
    protected int id_det;  
    protected int cod_referencia_prod;  
    protected int cod_consecutivo_cabe_sal;  
    protected int cod_punto_d;  
    protected int cantidad;  
    protected String descripcion;  
  
    protected Vector arrayDetalle_salida;  
  
    public Detalle_salida()  
    {  
        id_det=0;  
        cod_referencia_prod=0;  
        cod_consecutivo_cabe_sal=0;  
        cod_punto_d=0;  
        cantidad=0;  
        descripcion="";  
  
        arrayDetalle_salida=new Vector();  
        asignarTamanoVector();  
    }  
}//cierre del metodo constructor detalle salida
```

```

protected void asignarTamanoVector()
{
    arrayDetalle_salida.setSize(7);
} // cierre del metodo asignar tamano vector

public void setid_Det(int id_d)
{
    id_det=id_d;
    arrayDetalle_salida.setElementAt(id_det,0);
} // cierre del metodo setid_det
public int getid_Det()
{
    return id_det;
} // cierre del metodo getid_det

public void setcod_referencia_Prod(int cod_prod)
{
    cod_referencia_prod=cod_prod;
    arrayDetalle_salida.setElementAt(cod_referencia_prod,1);
} // cierre del metodo setcod_referencia_prod
public int getcod_referencia_Prod()
{
    return cod_referencia_prod;
} // cierre del metodo getcod_referencia_prod

public void setcod_consecutivo_cabe_Sal(int cod_cabe_sal)
{
    cod_consecutivo_cabe_sal=cod_cabe_sal;
    arrayDetalle_salida.setElementAt(cod_consecutivo_cabe_sal,2);
} // cierre del metodo setcod_consecutivo_cabe_sal
public int getcod_consecutivo_cabe_Sal()
{
    return cod_consecutivo_cabe_sal;
} // cierre del metodo getcod_consecutivo_cabe_sal

public void setcod_punto_D(int cod_d)
{
    cod_punto_d=cod_d;
    arrayDetalle_salida.setElementAt(cod_punto_d,3);
} // cierre del metodo setcod_punto_d
public int getcod_punto_D()
{
    return cod_punto_d;
} // cierre del metodo getcod_punto_d

public void setCantidad(int cant)
{
    cantidad=cant;
    arrayDetalle_salida.setElementAt(cantidad,4);
} // cierre del metodo setcantidad
public int getCantidad()

```

```
{
    return cantidad;
} // cierre del metodo getcantidad

public void setDescripcion(String desc)
{
    descripcion=desc;
    arrayDetalle_salida.setElementAt(descripcion,5);
} // cierre del metodo setdescripcion
public String getDescripcion()
{
    return descripcion;
} // cierre del metodo getdescripcion

public Vector getDetalle_salida()
{
    return arrayDetalle_salida;
} // cierre del metodo getdetalle_salida

} // cierre de la clase detalle salida
```


10. REFERENCIAS

Asociación Psicológica Americana. Quinta edición. (2010). Bogotá D.C. APA, pág. 2 - 10.

INSTITUTO COLOMBIANO DE NORMAS TECNICAS. Normas colombianas para la presentación de trabajos de investigación. Quinta actualización. Bogotá D.C. ICONTEC, 2002. pág. 130. NTC 1486.

PRESSMAN, ROGER. Ingeniería del Software un enfoque practico. Cuarta edición. Bogotá D.C. Mc Graw Hill Interamericana de España, 2002. pág. 602.

MARTÍNEZ ANGEL, Jonathan. SISTEMA DE INFORMACIÓN INTERACTIVO PARA LA ASOCIACIÓN DE EMPRESARIOS DE SIBATE, SOACHA Y SUR DE BOGOTA D.C. ASOMUÑA. Santafé de Bogotá, 2004, 3 p. Trabajo de grado (Tecnología en Informática). Corporación Universitaria Minuto de Dios. Facultad de Ingeniería Sistemas. Programa de Tecnología en Informática.

Consultado el 30 de Agosto de 2009 en el sitio web <http://www.apastyle.org/eleceref.html>

Consultado el 10 de Octubre de 2009 en el sitio web <http://www.desarrolloweb.com/manuales/>

11. CONCLUSIONES

A través de este proyecto nos hemos podido dar cuenta la gran importancia que tiene el aplicar un control de inventarios por medio de un sistema de información el cual facilitara la organización de dicha Comercializadora y así poder ver la productividad.

Fue interesante conocer el manejo de una Comercializadora de productos de aseo ya que de estas poco se escucha hablar y nos pudimos dar cuenta el gran movimiento de productos que esta maneja.

De acuerdo al seguimiento que realizamos durante la elaboración del sistema de información encontramos que la situación financiera de la empresa es óptima ya que encontramos que el costo de la utilidad es superior al costo de comercialización.

12. GLOSARIO

- **Aplicación:** Programa escrito para resolver un problema, dar un informe o actualizar un(os) archivo(s) determinado(s) o una base de datos especificada.
- **Base de datos:** Serie de datos afines acerca de un tema organizado de una forma practica tal, que proporciona una base o fundamento para procedimientos, como la recuperación de información, la elaboración de conclusiones y toma de decisiones. A cualquier agrupamiento de información que sirva a estos propósitos se le califica como una base de datos. La información se divide por lo general en registros de datos, cada uno de los cuales tiene uno o más campos de datos.
- **Bases de datos relacionales:** Utiliza los valores coincidentes de campos comunes de dos tablas para relacionar información de ambas.
- **Cliente:** Computador cuya función es la de solicitar información o aplicaciones al servidor de la red.
- **Dato:** Información dispuesta de manera adecuada para su tratamiento por un computador.
- **Diagrama de flujo:** Diagrama secuencial empleado en muchos campos para mostrar los procedimientos detallados que se deben seguir al realizar una tarea, como un proceso de fabricación. También se utilizan en la resolución de problemas, como por ejemplo en algoritmos. Los diagramas de flujo se usan normalmente para seguir la secuencia lógica de las acciones en el diseño de programas de computadoras.
- **Diccionario de datos:** Base de datos acerca de la terminología que se utilizará en un sistema de información. El significado de un diccionario de datos, puede considerarse su contenido como "datos acerca de los datos"; es decir, descripciones de todos los demás objetos (archivos, programas, informes, sinónimos...) existentes en el sistema.

Un diccionario de datos almacena la totalidad de los diversos esquemas y especificaciones de archivos, así como sus ubicaciones. Si es completo incluye también información acerca de qué programas utilizan qué datos, y qué usuarios están interesados en unos u otros informes.

- Información: Conjunto de datos que le sirven a una persona para concebir una idea.
- Manual: Folleto en que se compendia lo mas sustancial de una materia.
- Proceso: Secuencia de acciones o sentencias que colectivamente realizan una tarea deseada.
- Sistema: Conjunto de objetos y de relaciones entre objetos y sus atributos, ambos ligados los unos a los otros y a su entorno para formar un todo. Un sistema es cualquier conjunto de dispositivos que colaboran en la realización de una tarea; se refiere también a cualquier colección o combinación de programas, procedimientos, datos y equipamiento utilizado en el procesamiento de información.
- Sistema de información: Conjunto de elementos que interactúan entre si, con el fin de apoyar las actividades de una empresa.
- Usuario: Es la persona que utiliza o aprovecha los servicios prestados por una empresa.