

ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE LATACUNGA

FACULTAD DE INGENIERÍA EN SISTEMAS E
INFORMÁTICA

DESARROLLO DE UNA APLICACION WEB PARA LA
GESTION DE INVENTARIOS EMPLEANDO LA TECNOLOGIA
JAVA BEANS UTILIZANDO SOFTWARE DE CODIGO
ABIERTO
(CASO PRACTICO CULTIVOS PREMIUN S.A)

PROYECTO PREVIO A LA OBTENCION DEL TITULO DE
INGENIERO EN SISTEMAS E INFORMÁTICA

CHINGO CHILUISA LILIA PILAR
MASAPANTA AIMARA GLADYS YOLANDA



**FACULTAD DE INGENIERÍA
EN SISTEMAS E INFORMÁTICA**

Chingo Lilia
Masapanta Gladys

DESARROLLO DE UNA APLICACION WEB PARA LA GESTION DE
INVENTARIOS EMPLEANDO LA TECNOLOGIA JAVA BEANS
UTILIZANDO SOFTWARE DE CODIGO ABIERTO
(CASO PRACTICO CULTIVOS PREMIUN S.A)

Latacunga, septiembre 2005

CAPITULO I

1. Introducción

Las flores ecuatorianas son consideradas como las mejores del mundo por su calidad y belleza inigualables.

La situación geográfica del país permite contar con micro climas y una luminosidad que proporciona características únicas a las flores como son: tallos gruesos, largos y totalmente verticales, botones grandes y colores sumamente vivos y el mayor número de días de vida en el florero.

El sector florícola ha portado al desarrollo del país, como el de algunas provincias y ha demostrado un mayor crecimiento durante los últimos años, se trata de un sector dinámico de crecimiento; Ecuador con el fin de introducirse en un mercado internacional y poder competir como uno de los mejores países exportadores de flores prueba con nuevas variedades que se caracterizan por sus formas variadas, colores, tamaños, por su larga vida después del corte y por ser muy fuertes, resisten la manipulación, convirtiéndose en el país con el mayor número de hectáreas cultivadas y produciendo la gama más variada de colores, permitiendo de esta manera darse a conocer mundialmente e incrementar fuentes de trabajo para un país tercer mundista brindando la estabilidad en los hogares.

Las florícolas se orienta a la optimización del uso de los recursos con proyección a largo plazo y es dinámica, ya que está sujeta a un mejoramiento continuo para alcanzar más altos niveles de rentabilidad de las empresas dentro de un concepto de desarrollo sostenible.

1.2 Situación Actual de Cultivos Premiun S.A.

Cultivos Premiun S.A se encuentra ubicada en la parroquia de Eloy Alfaro provincia del Cotopaxi, uno de los lugares más hermosos de la Sierra Ecuatoriana. La finca tiene una extensión de 12 hectáreas siendo su principal razón social la producción de excelentes rosas, la misma que se encuentra constituida por 12 bloques, lo restante están ocupadas por oficinas, post-cosecha, cuartos fríos, bodegas, parqueaderos y espacios verdes.

La misión de Cultivos Premiun S. A. es preocuparse de todos los detalles, desde su cultivo hasta que la rosa sea enviada en perfectas condiciones, apoyada de personal especializado que está capacitado para clasificar y empacar correctamente, con el fin de promover y fortalecer el prestigio del sector florícola ecuatoriano a nivel nacional e internacional, a través del desarrollo humano, ecológico, comercial, logístico y tecnológico, siendo reconocido por su seriedad, prestigio comercial y su consistente calidad de producto.

Cultivos Premiun S.A se encarga de cultivar 57 variedades de rosas las mismas que se encuentran dentro de los invernaderos y están contruidos para proteger el producto final, teniendo en cuenta sus características fisiológicas y de mercado; estos invernaderos cumplen unas condiciones mínimas como: tener grandes dimensiones, la transmisión de luz debe ser adecuada, la altura tiene que ser considerable y la ventilación en los meses calurosos debe ser buena, la calefacción durante el invierno, junto con la instalación de mantas térmicas para la conservación del calor durante la noche y agua de manantial.

La finca cuenta con un depósito que suministra el agua a los invernaderos garantizando el suministro constante. En un período de seis meses la flor se

encuentra en su etapa final dando paso a la recolección, el proceso empieza a las 7 : 00 de la mañana, con la ayuda de técnicos y trabajadores inicia todos los días del año y en todos los invernaderos. El corte es realizado por personal calificado para esta tarea, cada uno conoce el punto de corte específico de la variedad y el manejo de la flor después del corte. Durante el proceso de corte de la rosa se la agrupa por variedad y se la pone en mallas plásticas con protección para evitar el maltrato del botón y follaje. De manera casi inmediata pasa un coche por el sistema de cable vía, recogiendo las mallas y trasladándolas a la post-cosecha esto toma un tiempo máximo de 10 minutos, con este proceso rápido se garantiza una mayor duración de vida de las rosas.

Antes de ingresar a la sala de post-cosecha, las mallas que vienen con las rosas del corte del día son revisadas una a una para examinar que los puntos de corte sean los establecidos y estén uniformes. Luego de este control ingresan a la sala de post cosecha; previamente pasan a ser clasificadas, todas las rosas tienen la primera hidratación en post-cosecha que está entre 3 y 5 grados centígrados, en el cual permanecen mínimo por 2 horas, esto ayuda a que la flor tenga una buena hidratación inicial en post-cosecha.

La flor es clasificada de acuerdo a su variedad, tamaño, punto de corte, grosor del botón, coloración y condición fitosanitaria en cada una de las mesas de clasificación de la sala de acuerdo a los parámetros y es colocada en los árboles de categorización (En este punto se decide si la flor es apta para exportación, orientándose a los parámetros del control de calidad lo que no sucede con la flor nacional) o es desechada para ser picada y convertida en compost (abono orgánico). Luego de pasar por un estricto control de calidad, la flor apta para exportación es embonchada en ramos de 6,10, 12, 20, 24 o 25 tallos dependiendo de las necesidades de los clientes. La finca produce rosas de 40 a 90 cm. con intervalos de grado de 10cm. La disponibilidad de flor en los diferentes grados viene determinada por las características mismas de cada variedad y por el manejo que le den los técnicos en la finca.

Todos los ramos producidos son realizados con láminas de cartón corrugado e impreso el logotipo de la empresa y son protegidos por láminas plásticas

transparentes; en algunos se puede incluir gys, verdes, y comida, dependiendo de las especificaciones del cliente (gys es una flor decorativa que son colocados en los bouquets, verdes son helechos su finalidad es adornar el ramo, la comida es un sobre de 10 gramos de cristal que se deposita en el agua para conservar la flor). Este empaque es estándar, a menos de que el cliente requiera de otras especificaciones, en este caso la finca se adapta a las necesidades requeridas. La información de cada ramo se llena en formularios de manera manual para luego ser digitados en el paquete Excel (identificando la variedad, grado y número de tallos), permitiendo que la información este disponible en el momento adecuado para la comercialización. Los procesos cumple con todos los estándares internacionales de clasificación y embonche, de esta manera se garantiza que los clientes reciben siempre un producto homogéneo y que llene sus expectativas. Todo este proceso puede ser visualizado desde las oficinas que se encuentran ubicadas en la parte superior de la post-cosecha, al nivel de los cuartos fríos. Esto brinda una gran ventaja a los clientes ya que los ejecutivos de comercialización pueden revisar prácticamente cada despacho, asegurando aún más la calidad del producto. La flor como producto a comercializar tiene varias características que son de especial relevancia para posteriormente entender como se maneja el mercado. Teniendo en cuenta las siguientes características de las flores:

- Son altamente perecibles(vida corta)
- No tienen uso alternativo(un solo fin)
- Su consumo es dinámico y esta relacionado a variables como la estética, modas, tradiciones, culturas de los pueblos, etc.
- No es un producto de primera necesidad, bien lujoso.

Presentación

Cuando se comercializa una flor se debe tener en cuenta que espera del producto el consumidor:

- Tallo recta.
- Tallo firme.
- Largo de tallo.

- Abertura de botón floral homogénea.
- Durabilidad del tallo.
- Buenas condiciones: color, presencia de todas las hojas superiores.
- Ausencia de plagas y/o enfermedades.
- Ausencia de tallos deformes.
- Ausencia de polen que manche mantelería y ropa.
- Precio razonable

Cultivos Premiun S A se ajusta a estas especificaciones, con la finalidad que el producto llene las expectativas anheladas por los clientes, tomando este punto como referencia para lograr una comercialización exitosa y transparente apoyados por un personal altamente calificados y realizando la transacción directamente desde la finca.

El objetivo primordial de la clasificación de los tallos es conocer cuántos tallos posee la finca para ser exportada y lo sobrante de la flor se la considera nacional cuyo fin es satisfacer las necesidades internas de nuestro sector; una vez ingresados los tallos el vendedor obtiene información exacta del disponible para la venta a este disponible se resta los tallos que van a ser utilizados en los standing (ordenes fijas que se tiene mediante un convenio con los clientes fijos) lo restante contacta otros clientes para ofrecer la flor.

Cultivos Premiun S. A controla la producción almacenada a través del inventario disponible, con la finalidad de realizar ventas óptimas, al mismo tiempo verificar los datos físicos (cuánta flor disponible) con los datos impresos en papel. De manera similar la finca realiza un control de tallos en cultivo, tallos procesados en post-cosecha, cuántos tallos ingresan para la exportación y cuántos tallos tienen en flor nacional, flor vendida y la flor que salió para obsequio.

1.3 Código de Barras

Un código de barras son símbolos con imágenes formadas por combinaciones de barras y espacios paralelos, de anchos variables. Representan números que a su vez pueden ser leídos y descifrados por los lectores ópticos o scanner.

Los códigos sirven para identificar los productos de forma única, cuenta con información detallada del artículo o del documento que los contiene, asociándose con una base de datos.

En la actualidad, el código de barras permite que cualquier producto sea identificado en cualquier parte del mundo, de manera ágil y sin posibilidad de error. Además logra una captura, almacenamiento y manejo de datos con gran eficiencia, reduciendo costos de operación, su utilización es muy común en la producción y distribución de trascendencia artículo, aunque se ha alcanzado a cubrir cualquier tipo de actividad humana, tanto en la industria e instituciones educativas como en el gobierno, la banca, instituciones médicas, aduanas y aseguradoras, entre otras utilizándolos para manejar más fácil todos los datos que pueden incluir y que las empresas e instituciones necesitan para la oportuna toma de decisiones.

1.3.1 Simbología en código de barras

La simbología es considerada un lenguaje en la tecnología de código de barras. Una simbología permite a un escáner y al código de barras "hablarse de tú a tu". Cuando un código de barras es digitalizado, es la simbología la que permite que la información se lea de manera precisa. Y es entonces, cuando un código de barras se imprime, la simbología permite a la impresora comprender la información que necesita ser turnada dentro de una etiqueta.

1.3.2 Tipos de Simbologías para Código de Barras

Al igual de la existencia en muchos idiomas y alfabetos, también existen varias simbologías de código de barras desarrollados con propósitos distintos; a primera vista se suelen parecer, sin embargo, tienen diferencias, dependiendo de la aplicación para la que fueron creados.

Se clasifican en dos grupos

1.3.2.1 Lineales (1-D)

Los lineales (1-D) se usan en productos y permiten incluir mensajes cortos.



1.3.2.1.1 UPC (Universal product code)

Este es el símbolo usado en objetos destinados al chequeo lineal. Los símbolos UPC son de longitud mixta, se usan en la venta al detalle y la industria alimenticia, y no se usa de alguna otra manera en otra. Se desarrollaron para cubrir las necesidades de almacenaje ya que 12 dígitos caben dentro de un espacio compacto razonable.



1.3.2.1.2 CÓDIGO 39

Es un estándar no utilizado para la industria alimenticia. Generalmente se utiliza para identificar inventarios y para propósitos de seguimiento en las industrias. Sin embargo el código 39 produce una barra relativamente larga y puede no ser adecuada si la longitud es un factor de consideración.



1.3.2.1.3 CÓDIGO 128

Se utiliza cuando es necesaria una amplia selección de caracteres más de lo que puede proporcionar el Código39 pero no tan amplio como proporciona el código en dos dimensiones. Cuando la dimensión de la etiqueta es importante, el código 128 es una buena alternativa porque es muy compacta lo que resulta en un símbolo denso.



1.3.2.1.4 POSNET

Es sólo para el Servicio Postal de Estados Unidos, esta simbología codifica los códigos postales para un procesamiento más rápido de entrega del correo.



1.3.2.2 Dimensiones 2-D

Son usados en documentos para controlar envíos o en seguros médicos, en general, en documentos que requieren la inserción de mensajes más grandes (de hasta 2 725 dígitos) como un expediente clínico completo.



Código de barras de dos dimensiones.

1.3.2.2.1 PDF 417

Conocido como un código de dos dimensiones, es una simbología de alta densidad no lineal que recuerda un rompecabezas. Pero la diferencia entre éste y los otros tipos de código de barras, es que el PDF417 es en realidad un Portable Data File (Archivo de Información Portátil, PDF) lo opuesto a simplemente ser sólo una referencia numérica.

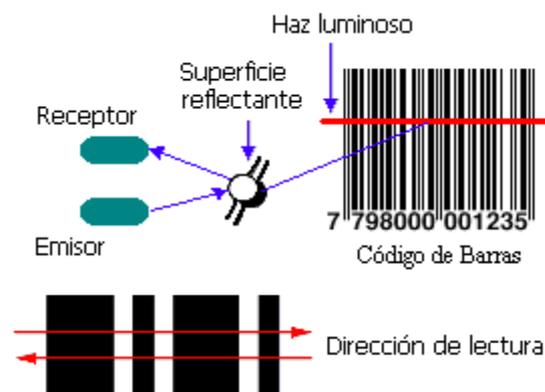


1.3.3 Lector Óptico

Los lectores ópticos o scanner son dispositivos que captan la información contenida en el código de barras. Son fáciles de instalar, existiendo en el mercado una gran variedad en marcas, tamaños, formas, pudiendo tener la forma de pluma o rastrillo (requieren hacer contacto con el código) o tipo pistola láser, que realizan la lectura a distancia.

1.3.4 Funcionamiento del lector

El lector de código de barras decodifica la información a través de la digitalización proveniente de una fuente de luz reflejada en el código y luego se envía la información a una computadora como si la información hubiese sido ingresada por teclado.



El procedimiento: el símbolo de código de barras es iluminado por una fuente de luz visible o infrarrojo, las barras oscuras absorben la luz y los espacios las reflejan nuevamente hacia un escáner.

El escáner transforma las fluctuaciones de luz en impulsos eléctricos los cuales copian las barras y el modelo de espacio en el código de barras. Un decodificador usa algoritmos matemáticos para traducir los impulsos eléctricos en un código binario y transmite el mensaje decodificado a un terminal manual, PC, o sistema centralizado de computación.

1.3.4 Ventajas del uso de Códigos de Barras

- Mayor precisión en la información
- Mayor productividad de los empleados
- Mayor precisión y control de inventarios
- Rastreo preciso en actividades
- Rastreo preciso de bienes transportados
- Rastreo de documentos durante un proceso
- Mejor control de entradas y salidas
- Rapidez en la captura de datos
- Reducción de errores
- Reducción de las mermas
- Los equipos de lectura e impresión de códigos de barras son flexibles y fáciles de conectar e instalar.

1.4 Gestión de Inventario

El Inventario es el conjunto de productos y/o recursos utilizados en una organización (materias primas, productos terminados, repuestos, producto en proceso) empleados para satisfacer una demanda futura, el inventario es la base primordial para toda empresa, cualquiera que sea su tipo o naturaleza de la empresa. El saber la cantidad para invertir en él es necesario porque ayudará a determinar cual es monto a cubrir en materia prima para producción y a los agentes de venta el saber que cuentan con unidades suficientes para cubrir su demanda y cualquier eventualidad que pueda aumentar las utilidades de la empresa, y para conocer de que manera puede disminuir sus costos por tener inventarios que cubran todas estas características.

El inventario representa una inversión considerable para todas las empresas cualquiera que sea su naturaleza, es por ello que se hace indispensable prestarle atención especial a su manejo.

Para realizar una eficiente administración los responsables de esta área, deben controlar todos los niveles del inventario y considerar que este es una inversión significativa que si no se maneja de una forma adecuada puede convertirse en un problema que afectaría la gestión financiera de la empresa.

Una empresa que tenga un gran número de artículos de inventario debe analizar cada uno de ellos para determinar la inversión aproximada por unidad.

Una gran cantidad de organizaciones tienen en sus bodegas una gran cantidad de artículos que no tienen una misma característica, muchos de estos artículos son relativamente de bajo costo, en tanto que otros son bastante costosos y representan gran parte de la inversión de la empresa. Algunos de los artículos del inventario, aunque no son especialmente costosos tienen una rotación baja y en consecuencia exigen una inversión considerable; otros artículos, aunque tienen un costo alto por unidad, rotan con suficiente rapidez para que la inversión necesaria sea relativamente baja.

1.4.1 Control de Inventario

El control de un Inventario se lo realiza dependiendo de la naturaleza de la empresa, pero por lo general los aspectos más relevantes para el control de inventario son: producción y ventas.

Para llevar un control sobre producción y ventas se realiza un informe dependiendo de la necesidad de que la empresa requiera este, pudiendo ser diario, semanales o mensuales semestrales.

En este informe se podrá visualizar la producción y ventas rigiéndose a lo almacenado por el área de producción y registrado por el área de ventas, teniendo cifras exactas en números y gráficamente (en diagramas de pastel y barra) logrando un visible el éxito o fracaso de la de la empresa, ayudando así para la toma de decisiones en forma oportuna y feraz, con el único objetivo de acrecentar la empresas.

1.4.2 Comercialización

El concepto nace hace varios años con una sola preocupación por vender, y obtener utilidades, este proceso abarca desde que el producto está listo para la venta hasta que llegue al cliente, con un solo objetivo de obtener utilidades necesarias para el desarrollo de la empresa que lo ofrezca, se debe tener en cuenta la definición del negocio es decir definir cuál es la razón de ser y cuál es la necesidad de los consumidores que el producto satisface. Para ello, se necesita saber cuáles son las necesidades más frecuentes que satisfacen a los clientes cuando compran el producto, por ejemplo: las industrias de bebidas alcohólicas "venden" alegría, diversión, además de la bebida, los supermercados "venden" marca, exclusividad, estatus o diseño, la industria de cosméticos "vende" belleza, esperanza, las florícolas venden decoración, lujo incluso existen productos que "venden" seguridad, confianza y felicidad, y los consumidores están dispuestos a pagar por ello.

Para definir un negocio se determinan las necesidades ocultas que a los consumidores se satisfacen cuando compran el producto. La frase clave es conocer el mercado. Las necesidades del mercado, es decir de los consumidores son las que dan la pauta para poder definir mejor que es lo que se va a vender y a quienes así como dónde y como lo hace, en conclusión la comercialización no es otra cosa que "el conjunto de todas las actividades empresariales relacionadas con el flujo de bienes y servicios desde el punto de la producción inicial (agraria) hasta que ellos (los bienes) están en manos del consumidor último". Teniendo en clara que los productos se diferencian fundamentalmente en la presentación, calidad, precio, diseño y esto hace que no sea sencillo escoger el producto más adecuado a nuestras necesidades

CAPITULO II

SOFTWARE DE CÓDIGO ABIERTO O LIBRE

2. INTRODUCCIÓN

Código abierto es una tendencia que nació hace varios años con sus principios en UNIX, pero durante la segunda mitad de la presente década tomo un enorme impulso, al grado de que los desarrollos en esta modalidad rivalizan con productos de empresas como Microsoft, además los mercados se han dado cuenta que son una excelente forma de mejorar los productos, crear lealtad del consumidor e incluso, incrementar las ventas.

Los códigos abiertos son de dominio público, permitiendo a cualquier persona obtenerlos, modificarlos y mejorarlos; con el fin de enriquecer de manera automática el producto. Esto es lo que se conoce como "Copyleft", en oposición al "Copyright", con los que se registran todos los productos comerciales.

El procedimiento para crear un programa libre se inicia con el registro (o Copyright), para después integrarle una serie de condiciones de distribución, en las que se especifica que el programa puede ser copiado, modificado y utilizado sin límite, siempre y cuando, se distribuya este y cualquier otro programa derivado del mismo con el código fuente y los términos de distribución. La distribución de copias de estos productos sin las características establecidas o el intentarlo volverlo propietario, es ilegal.

Open Source o "Software Libre" es la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, es decir que tiene cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades, el acceso al código fuente.
- La libertad de distribuir copias, con lo que se puede ayudar a los demás.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

"Software libre" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial

Cuadro de Software de código abierto

Sistemas Operativos	Linux	Red Hat Linux
		Gento Linux
		Solaris
	Debian	Augustux: live CD, KDE
		Gnoppix: live CD, GNOME
		Guadalinex: live CD, GNOME
		Knoppix: live CD, KDE
Navegadores Web	Amaya	
	Epiphany	
	Mozilla	
	Lynx	
Herramientas de Desarrollo	Base se Datos	Mysql
		Post Mysql
	Lenguajes de	Php

	Programación	JSP
		Perl
Entornos de Escritorio	KDE (Kit Desktop Environment) Para linux	
	GNOME funcionamiento bajo tecnología X Window	
Servidor de Ficheros	Samba	
Suite de ofimática	OpenOffice	
Editor de gráficos	GIMP	
Servidores Web	Apache	
	Tomcat	

TABLA 1

A continuación se detalla los Software mas relevantes y consumidos por los usuarios seleccionados por sus características.

2.1 Linux

Linux es un Sistema Operativo, como lo son UNIX, o Windows, o DOS o MacOS. Un Sistema Operativo es un conjunto de programas que trabajan coordinadamente para facilitarle al usuario la "operación" de su computadora: lectura y escritura de discos, manejo de archivos y directorios, manejo de periféricos, etc.

Los sistemas operativos pueden presentar una interfaz de texto (como las "clásicas" o "viejas") o una interfaz gráfica con íconos y ventanas que facilitan y hacen más estético el uso del mismo.

Linux es un sistema basado en los principios de UNIX, por lo que se trata de un sistema operativo muy seguro y poderoso, con ventajas adicionales: es gratuito y abierto.

Linux es un sistema jerárquico, es decir, utiliza una estructura en árbol al igual que MS-DOS. Tanto los archivos como los directorios se identifican utilizando casi todos los caracteres diferenciándose entre mayúsculas y minúsculas.

Linux es distribuido bajo la Licencia General Pública de GNU, lo cual significa que puede ser distribuido, copiado y modificado gratuitamente, a condición de no imponer ninguna restricción en sucesivas distribuciones.

2.1. 2 Arquitectura de Linux

Los componentes de Linux son: el núcleo (kernel), el shell, el sistema de archivos y las utilidades.

2.1.2.1 El núcleo

Es el programa modular que ejecuta programas y administra dispositivos de hardware tales como los discos y las impresoras.

2.1.2.2 El shell

Proporciona una interfaz para el usuario. Recibiendo órdenes del usuario y enviándolas al núcleo para que se ejecuten.

2.1.2.3 El sistema de archivo

Los archivos están organizados en directorios. Cada directorio puede contener un número cualquiera de subdirectorios, cada uno de los cuales puede a su vez, contener otros archivos.

2.1.2.4 Las utilidades

Son programas especializados, tales como editores, compiladores y programas de comunicaciones, que realizan operaciones de computación estándar.

El núcleo, el shell y el sistema de archivos forman la estructura básica del sistema operativo. Con estos tres elementos pueden ejecutar programas, gestionar archivos e interactuar con el sistema. Además, las utilidades han pasado a ser consideradas como características estándar del sistema.

Estas utilidades efectúan operaciones sencillas otras son programas complejos con sus propios juegos de órdenes. Pueden clasificarse en tres amplias categorías: editores, filtros y programas de comunicaciones.

También existen utilidades que efectúan operaciones con archivos y administración de programas.

Linux dispone de varios tipos de sistema de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X Windows (Interfase gráfico estándar para máquinas UNIX), comparado con los entornos comerciales.

2. 1.3 Características de Linux

2.1.3.1 Linux Multitarea

La multitarea no consiste en hacer que el procesador realice más de un trabajo al mismo tiempo (un solo procesador no tiene esa capacidad), lo único que realiza es presentar las tareas de forma intercalada para que se ejecuten varias simultáneamente, sin necesidad de tener que parar la ejecución de cada aplicación.

2.1.3.2 Linux Multiusuario

Permite que múltiples usuarios accedan al mismo tiempo a través de terminales, y que distribuya los recursos disponibles entre todos. Así mismo, el sistema proporciona la posibilidad de que más de un usuario pueda trabajar con la misma versión de un mismo programa al mismo tiempo, y actualiza inmediatamente cualquier cambio que se produzca en la base de datos, quedando reflejado para todos.

2.1.3.3 Linux multiplataforma

Es decir que puede correr en distintos tipos de Hardware o Sistemas Computacionales tales como Intel, AMD, motorola, sun, sparc, etc

2.1.4 Ventajas de Linux

- Linux envuelve el concepto de Software Libre, esto brinda al usuario del sistema muchas ventajas entre las cuales esta la de disponer del código fuente de los programas, mediante esto el usuario puede adaptar los programas a sus necesidades personales, puede hacer mejoras, puede ver cómo está programado y hacer su propio programa sin necesidad de comprarlo. El concepto de Software Libre busca librar al usuario de los intereses del sector comercial y así desarrollar software funcional, más complejo y que realmente atienda las necesidades del mismo.

- No se limita a ejecutar sus propias aplicaciones y programas, sino que se extiende y adapta a otras plataformas, esto lo hace por medio de programas de emulación para que dentro de LINUX se puedan correr aplicaciones y programas de otros sistemas operativos. Se distribuye su código fuente, lo cual permite a cualquier persona que así lo desee hacer todos los cambios necesarios para resolver problemas que se puedan presentar, así como también agregar funcionalidad.

- Es desarrollado en forma abierta por cientos de usuarios distribuidos por todo el mundo a través de la red de redes Permitiendo un rápido y eficiente ciclo de desarrollo.

- Cuenta con un amplio y robusto soporte para comunicaciones y redes, lo cual hace que sea una opción atractiva tanto para empresas como para usuarios individuales.
- Da soporte a una amplia variedad de hardware y se puede correr en una multitud de plataformas: PC's convencionales, computadoras Macintosh y Amiga, así como costosas estaciones de trabajo

2.1.5 Desventajas de Linux

- Linux un sistema operativo diferente a lo ofrecido por el mercado, lo que le hace ser un sistema operativo difícil de mantener y tener una excesiva dificultad en el soporte de nuevas aplicaciones
- Linux no ofrece entrenamiento o certificación, como lo hace Windows, debiendo ser adquirido estas características de conocimiento del sistema operativo a través de la experiencia del usuario o por manuales ofrecidos en la red de redes.
- Linux no brinda mantenimiento ni soporte técnico, razón suficiente para que los usuarios prefieran Windows y no Linux, aunque windows tenga un costo por su licencia.

2.2 Tomcat

El Tomcat es un servidor Web que se utiliza para ejecutar aplicaciones realizadas con Java en sus divisiones

JSP y Servlets (paginas con HTML, javaScript), es el servidor Java que más se ajusta a los estándares de implementación para las Java Server Pages (JSP) y las especificaciones Java Servlet. Está integrado al proyecto

Apache-Jakarta, fue concebido y fundado con licencia **GNU FDL** para instalar, configurar y distribuir el servidor de Servlet y JSP Tomcat.

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web (Apache, y Netscape), existiendo numerosas particularidades que hacen que sea una plataforma de predilecto para el desarrollo de aplicaciones y servicios web

2.2.1 Arquitectura de Tomcat

Tomcat tiene la siguiente estructura de directorios con los cuales trabaja:

Bin._ Contiene los scripts ejecutables de arrancar/parar de Tomcat

Conf._ Contiene varios archivos de configuración incluyendo server.xml (el fichero de configuración principal de Tomcat) y web.xml que configura los valores por defecto para las distintas aplicaciones desplegadas en Tomcat.

Doc._ Contiene toda la documentación sobre Tomcat.

Lib._ Contiene varios archivos jar que son utilizados por Tomcat.

Webapps._ es el directorio que contiene las aplicaciones web realizados por el usuario, y a demás asilan todos los ejemplos que soporta Tomcat.

Adicionalmente se puede crear, los siguientes directorios:

Work._ Generado automáticamente por Tomcat, este es el sitio donde

Tomcat sitúa los archivos intermedios (como las páginas JSP compiladas) durante su trabajo. Si se borra este directorio mientras se está ejecutando Tomcat no se podrá ejecutar páginas JSP.

2.2.2 Características de Tomcat

2.2.2.1 Interfaz Amigable

Posee una interfaz amigable e intuitiva para cualquier tipo de usuario sea este experto o principiante, esta es una característica primordial que pocos software poseen.

Compilador Jasper se encarga de compila JSPs convirtiéndolas en servlets, funcionando con cualquier servidor web que soporte servlets y JSPs.

2.2.2.2 Software Libre

Es libre para todos los usuarios disponen de un acceso libre al código fuente y al igual que su forma binaria, según los términos establecidos en la Apache Software Licence.

2.2.2.3 Servidor de Aplicaciones Web (java)

Tiene la capacidad transformar páginas estáticas a dinámicas lo que le hace ser un servidor de aplicaciones web en java.

Creación de nuevos Contexto._ tiene la facilidad de crear nuevos contexto para trabajar conjuntamente con apache

2.2. 3 Fundamentos para el funcionamiento de Tomcat

Acontinuación se detalla los fundamentos para trabajar con Tomcat

- Instalación del JDK(("Standard Development Kit)
- Configuración de las Variables de Entorno

2.2.3.1 JDK (Java Development Kit)

Java Development Kit es un paquete para desarrollar cualquier aplicación Java, incluye el API de Java (especificaciones de la manera en que un componente Java puede ofrecer una funcionalidad determinada;), el compilador de Java, y el JVM "Java Virtual Machine" (interpreta el Byte Code, generado por cualquier programa en Java, el Byte Code,) para la plataforma correspondiente.

JDK agrupa diversas funcionalidades necesarias para desarrollar programas Java, en la actualidad existen diversos ambientes y plataformas, los cuales ofrecen:

- Un compilador Java, capaz de generar Byte-Code.
- Un JVM ("Java Virtual Machine"), ejecutar Byte- Code.
- Un conjunto de Clases base utilizadas para generar programas Java.
- Otras utilerías para administrar código escrito en Java.

2.2.3.1.1 Estructura de JDK

Se encuentra dividida en tres directorios :

bin: Incluye los ejecutables para la generación de programas Java

lib: Contiene las librerías (Clases) base empleadas en la generación de la gran mayoría de programas Java.

jre: Incluye el ambiente necesario para ejecutar programas Java, el cual a su vez se encuentra sub-dividido en distintos directorios.

2.2.3.2 Variables de Entorno

Las variables de entorno, como su nombre indica son variables, que almacenan valores que describen las propiedades del entorno de trabajo pueden ser diferentes para los usuarios, Las variables de entorno se definen en los archivos de inicialización, al arrancar los programas, o desde la shell de los usuarios.

2.2.3.2.1 PATH

Define la ruta de acceso al sistema, en la que se encuentran almacenados los paths de aquellos directorios a los que el usuario tiene acceso directo , pudiendo ejecutar los comandos de java proporcionadas por el JDK, sin necesidad de acceder a dicho directorio.

2.2.3.2.2 CLASSPATH

Define las rutas de acceso para las diversas librerías empleadas en ambientes Java.

2.2.4 Archivos de configuración de Tomcat

La configuración de Tomcat se basa en dos ficheros:

2.2.4.1 server.xml

Es el fichero de configuración global de Tomcat. Este es un archivo en XML, tiene una estructura conforme a XML, si la estructura del archivo es alterada Tomcat indicara el problema cuando arranque, el archivo se encuentra bajo el directorio

/usr/local/tomcat/conf

2.2.4.2 web.xml

Es desplega la aplicación que está conteniendo el directorio WEB-INF. Define varios parámetros que son usados cuando se despliega la aplicación en el contenedor de Servlets/JSP Tomcat. Este tiene una estructura XML

2.2.5 Ventajas de Tomcat

- Es un servidor de aplicaciones web para Java, gratuito fácil de adquirir, con documentación necesaria repartida por toda la red de redes tanto para su uso e instalación.
- Es un servidor de paginas dinámicas que se ejecuta en varios sistemas operativos como Windows XP ,98, milleniun, Solares, Linux .

2.2.6 Desventajas de Tomcat

- Tomcat tiene insuficiente rendimiento cuando sirve de servidor de paginas estáticas, siendo necesario que Tomcat y apache combinen sus mejores características para que las aplicaciones web, que posean tanto paginas estáticas como dinámicas tengan un excelente rendimiento y eficiencia

2. 3 JSP (Servidor de Páginas Java)

JSP (Java Server Pages) es una tecnología para generar páginas web de forma dinámica en el servidor, con especificaciones de [Sun Microsystems](#) (compañía líder en soluciones con alta Tecnología Informática para sistemas abiertos, Sun significa "Standfor University Network".), basado en [scripts](#) utilizando lenguaje [java](#).

JSP desarrolla aplicaciones web independientes de la plataforma, permite separar la interfaz del usuario de la generación del contenido dinámico, logrando procesos de desarrollo mas rápidos y eficientes, pueden también acceder directamente a componentes Java Beans ó Enterprise Java Beans (EJB), instan ciándolos y establecer propiedades e invocar a métodos desde la página JSP.

JSP permiten desarrollar aplicaciones n-capas donde se separan la capa de datos, la lógica del negocio y la lógica de presentación, encapsulando, generalmente, en Beans el acceso a los datos.

La especificación de JSP es el producto de varias empresas desarrolladoras de software, encaminadas por **Sun Microsystems**. Sun permito hacer tales especificación logrando como objetivo, la libre disponibilidad , con una característica de la tecnología Java "Write Once, Run Anywhere" (Escríbelo una vez, córrelo donde quieras).

JSP se basa en los siguientes conceptos:

- Plantillas o Templates. son una parte importante del contenido esta compuesto por una plantilla.(fragmentos HTML o de texto).
- Contenido dinámico. provee una forma simple de agregar datos dinámicos a cada plantilla al permitir incrustar instrucciones de programación de lenguaje java en este, aunque se puede utilizar otro lenguaje que sea soportado por el contenedor de JSP
- Encapsulación de funcionalidad. JSP provee dos formas distintas para encapsular funcionalidad: componentes JavaBean y bibliotecas de etiquetas.

2.3.1 Arquitectura de JSP

Además del HTML normal, existen tres tipos de construcciones JSP elementos de script, directivas y acciones

2.3.1.1 Plantilla de Texto: HTML estático

Las páginas JSP pueden estar constituidas por HTML estático, conocido como plantilla de texto, creada en cualquier herramienta para generar páginas Web. La única excepción para que la plantilla pase tal como es, es que debemos tener al inicio "<%" el al final "<\%" con esto el contenedor de paginas Web (Tomcat) asumirá que es una pagina JSP, pero en realidad será una plantilla de texto o HTML estático

2.3.1.2 Elementos de script

Permiten especificar o insertar código Java dentro del servlet que se generará desde la página JSP. Hay tres formas:

Expresiones.

Scriptlets

Declaraciones

2.3.1.2.1 Expresiones JSP

Una expresión es un pequeño fragmento de código que devuelve un resultado (salida) por pantalla. Tiene la siguiente forma:

`%= expresión Java(es cualquier expresión de Java válida) %>`

La expresión Java se evalúa, se convierte en un string, y luego es e insertada en la página . La evaluación es realizada durante la ejecución (es decir cuando se solicita la página) teniendo un acceso total a la información sobre la solicitud.

2.3.1.2.2 Scriptlets JSP

Es un conjunto de instrucciones o sentencias de Java incluidas en una página HTML. Estas instrucciones se distinguen del HTML porque están colocadas entre los marcadores <% instrucciones o sentencias de Java %> para que el intérprete de JSP sepa que debe procesar todo el código que se encuentre dentro de estas etiquetas. Su sintaxis es:

```
<% sentencias; [sentencias; ...] %>
```

2.3.1.2.3 Declaraciones JSP

Permite definir métodos o campos que serán insertado en el cuerpo de la clase del servlet, fuera del método service. La declaración no genera ninguna salida por esta razón se usan conjunto con las expresiones

Tienen la siguiente forma:

```
<%! Código Java%>
```

utilizando conjuntamente con las expresiones;

```
<%! private int x = 0; %>
```

```
<%= ++x %>
```

2.3.1.3 Directivas.

Son etiquetas que se utilizan en una página JSP cuya principal característica es incluir el símbolo @ en su sintaxis, utilizadas para definir y manipular una serie de atributos dependientes de la página que afectan a todo el JSP.

Su forma es

```
<%@ nombre_directiva [ atributo_i = "valor_i" ] %>
```

Las más relevantes son:

2.3.1.3.1 Page

Permite definir uno o más atributos sensibles que afecta a la página asociada

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
```

2.3.1.3.2 Incluye

Permite incluir ficheros en el momento en que la página JSP se traducida a un servlet.

```
<%@ include file="url relativa" %>
```

2.3.1.4 Acciones

Controlan el comportamiento en tiempo de ejecución del JSP utilizan constructores de sintaxis XML (sensibles a mayúsculas) . Las acciones pueden insertar ficheros dinámicamente, reutilizar componentes JavaBeans, etc.

Un ejemplo relevante es

2.3.1.4.1 jsp:useBean

Permite cargar y utilizar un JavaBean en la página JSP. es una capacidad muy útil porque permite utilizar la reusabilidad de las clases Java

2.3.2 Objetos Implícitos

Son instancias que están disponibles en scripts y expresiones y NO están disponibles en declaraciones, son accesibles desde una página JSP ,no necesitan ser declarados ni instanciados.

Los más relevantes son:

2.3.2.1 request:

Es una instancia que encapsula una petición del cliente web que siendo procesado por el JSP.

2.3.2.1 Outl

objeto PrintWriter usado para devolver la respuesta al cliente

2.3.2.3 Sesión

Es el objeto de sesión HTTP asociado a la petición.

2.3.2.4 exception

Es una instancia. Representa un objeto excepción que provocará la invocación de una página de error.

2.3.3 Características de los JSP

2.3.3.1 Combinación de Lenguajes

Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts ejecutables en el servidor en sintaxis Java. Por lo tanto, las JSP podemos crearlos con editores HTML/XML habitual.

2.3.3.2 Multiplataforma

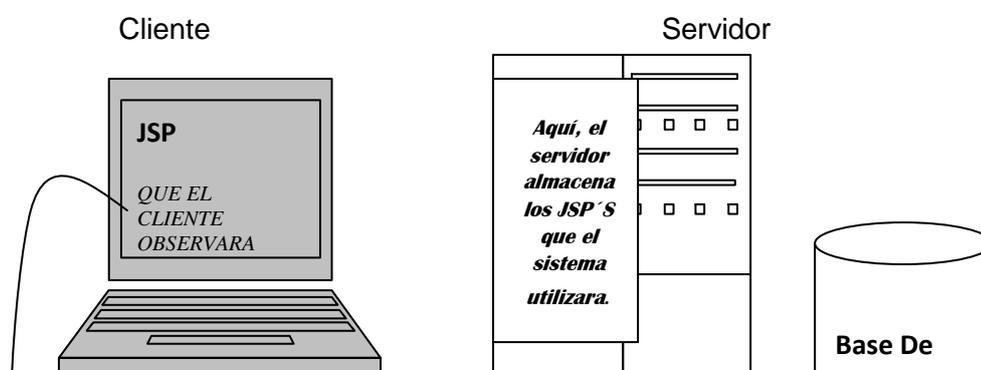
Se puede crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un

lenguaje multiplataforma. Java es en esencia un lenguaje multiplataforma es decir que puede correr en distintos tipos de Hardware o Sistemas Computacionales tales como Intel, AMD, motorola, sun, sparc, etc

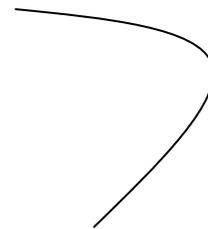
2.3.3.3 Ahorro de tiempo

Reduce el tiempo y el esfuerzo del entrenamiento, para la creación de páginas web para usuarios de experiencia en el lenguaje Java.

2.3.4 Funcionamiento de una Pagina JSP



Servidor WEB



TOMCAT ES EL MOTOR DE
JSP Y EN EL SERVIDOR

Las páginas JSP pueden ser desarrolladas en cualquier editor de páginas, para luego ser guardadas en el subdirectorio webapps de Tomcat, es necesario crearse el directorio Web-inf dentro del directorio generado (Tomcat\webapps\JSP\Web-inf) debe ser añadida la información de configuración al fichero web.xml indicando cual va a ser el directorio base de la aplicación: Después de cumplir estas especificaciones cada vez que el cliente solicita al servidor web la página JSP, pasa la petición al motor de JSP, en el cual se verifica si la página no se ha ejecutado antes ó fue modificada

después de la última compilación, tras lo cual la compila, convirtiéndola en Servlet, la ejecuta y devuelve los resultados al cliente en formato HTML.

2.3.4.1 Ejemplo de una Página JSP

```
<%String texto;  
  
texto = "Hola Mundo";  
  
out.println(texto);  
  
%>
```

2.3.5 Ventajas de los JSP

- Independencia de la plataforma
- Rendimiento mejorado
- Separación de la lógica de la aplicación de la presentación de los datos
- Uso de componentes reutilizables (Java Beans) o diferentes de otras aplicaciones pueden reutilizar los mismos componentes.
- Facilidad de administración y uso
- Tiene un importante respaldo de la sólida tecnología Java™.
- Las páginas JSP por ser similares a las de HTML, son más compatibles con las herramientas de desarrollo de Web (DreamWeaver , FrontPage, etc.).

2.3.6 Desventajas de JSP

- JSP es utilizado como interfase visual ósea como FRONT END, no posee lógica de negocios, para esto se utiliza los Beans que ayudan a manejar las reglas de negocio.

- Es necesario comprender y poseer experiencia en programación **Java**, que es un **lenguaje de programación Orientado a Objetos** por completo, además tener conocimiento básico en html.

2.4 MySQL

Es un programa interactivo que permite conectarse a un servidor MySQL, ejecutar consultas y observar los resultados de las mismas. Mysql puede ser usado también en modo batch.

MySQL es un Sistema de Gestión (o administrador) de Base se caracteriza por poner toda la información en tablas, relacionándolas entre datos añadiendo velocidad y flexibilidad. Además tiene una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías)

MySQL es un software de código abierto esto quiere decir que es accesible por cualquiera, para usarlo o modificarlo Usa licencia GPL (GNU Licencia Publica General)

Para trabajar con Mysql se puede utilizar mysql front que rápido sencillo de instalar

2.4.1 Características de Mysql

MySQL ofrece un alto rendimiento en velocidad para realizar operaciones, convirtiéndole en uno de los gestores que ofrecen mayor rendimiento.

Para poder ejecutarlo se necesita de escasos prerequisites de hardware con los cuales trabajará sin ninguna contrariedad.

Las utilidades de administración son excelentes comparados para otros gestores comerciales, debido a su gran facilidad de configuración e instalación.

El conjunto de aplicaciones Apache-PHP-MySQL son los más utilizados en Internet y buscadores de aplicaciones.

2.4.2 Modos de Manejo de Mysql

Existen dos formas:

2.4.2.1 Mysql de Manera Batch

Se escribe las consultas en un fichero de texto, y luego pueden ser ejecutados por mysql). Tiene algunas ventajas como:

Es útil cuando se ejecutan un cierto número de consultas frecuentemente, pudiendo generarse nuevas consultas similares a las existentes al copiar y editar estos scripts.

Cuando se ejecutan consultas que producen una gran cantidad de datos, es posible usar paginador para examinar los resultados de una mejor manera

2.4.2.2 Mysql de Manera Grafica

Una interfaz grafica sencilla de utilizar es mysql-Front

2.4.2.2.1 MySQL-Front

Es una aplicación utilizada para trabajar con MySQL. La administración de base de datos con esta aplicación es sencilla, se obtiene información fácil sobre las bases de datos, tablas, la estructura de las tablas y contenido. La interfaz es intuitiva con MySQL-Front se realiza acciones básicas como añadir, borrar o modificar tablas, campos, registros, y además se puede, ejecutar y matar procesos, exportar tablas a SQL-scripts o a otras bases de datos, replicar bases de datos, escribir queries en SQL, importar datos de ODBC, realizar diagnóstico de tablas (optimización, reparación, etc.), ver propiedades avanzadas de tablas (tipo, comentario, key_length, etc.)

Para el funcionamiento correcto se necesita

Microsoft C Runtime Library (msvcrt.dll).

Protocolo TCP/IP.

.ODBC (para la importación de datos).

La librería Cliente para MySQL Server incluida (libmysql.dll).

2.4.3 Ventaja de MySQL

- MySQL es muy rápido, confiable, robusto.

- Fácil de usar tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría).

- Posee la mayoría de comandos del lenguaje SQL (structured query language), que es un estándar en bases de datos. Es un servidor que funciona bajo linux teniendo una distribución gratuitamente con este sistema operativo.

- Mysql ofrece consistencia de las diferentes tablas que la componen, para conseguir esto de una forma más o menos fácil es utilizando la "Lógica Transaccional", será el propio gestor de base de datos el que proporcione mecanismos de bloqueo de ficheros y consolidación o retroceso en la operaciones con las tablas.

- La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

2.4.4 Desventajas de Mysql

- Un gran inconveniente de Mysql es el de no manejar la integridad referencial, esto lo hace un gestor pobre para considerables campos de aplicación, en especial para los programadores que provienen de otros gestores que poseen la característica.

CAPITULO III

3. TECNOLOGÍA JAVABEANS

Java proporciona facilidad para la construcción de aplicaciones, relacionados con el desarrollo que envuelve Java Server Pages (paginas dinámicas) , java aporta con componentes reutilizables, estos pueden ser insertados dentro de una página JSP cuando sean requeridos encapsulando toda la lógica de negocios, teniendo como resultado un código más simple, fácil de mantener , más legible (para los diseñadores de páginas) y eliminando el código de scripts incluido en las paginas dinámicas.

Java implementa la idea de componentes reutilizables con los JavaBeans o la tecnología JavaBeans estos son una clase java que sigue ciertas convenciones en el nombrado de sus métodos, de forma que sea posible operar con ella en tiempo de ejecución

Un JavaBean tiene propiedades y en función de que los métodos estén definidos la propiedad será de lectura, de escritura o de lectura/escritura.

3.1 Definición

JavaBeans son componente que tiene como característica principal ser reutilizados, debiendo cumplir ciertas características en cuanto a su diseño, tienen como objetivo el lograr reducir al máximo el código Java insertado en una página JSP. Los "Beans" pueden ser sencillos como un botón, complejos como el acceso a una base de datos

3.1.1 Características de Java Beans

3.1.1.1 Persistencia

Una vez que se han cambiado las propiedades de Bean, se puede guardar su estado y restaurar el Bean posteriormente.

3.1.1.2 Interoperabilidad

Esta característica forma parte del lema Java: "Write once, run everywhere" (Escríbalo una vez, ejecútelo en todos lados), después de ser creado, ejecutado y obtener un. Clase puede ser ejecutado en diversas plataformas obteniéndose el mismo resultado.

3.1.1.3 Introspección

Es la técnica de descubrimiento de propiedades, permite buscar las propiedades que coincidan con el patrón de nombres de métodos definido. Las propiedades que cumplen con esta regla son lectura y modificación.

3.1.2 Ámbito del Java Beans

Son reglas que definen el tiempo de vida y de visibilidad de un bean dentro de una aplicación basada en Web, La especificación Java Server Pages (JSP) define el ámbito usado por el javabeans.

3.1.2.1 page Beans

Son visibles dentro de una sola página JSP, para el tiempo de vida de la solicitud actual

3.1.2.2 request – Beans

Son visibles dentro de una sola página JSP, así como en cualquier página o servlet que esté incluido en esta página, o reenviado por esta página. (Atributos Request).

3.1.2.3 session – Beans

Son visibles para todas las páginas JSP y los servlets que participan en una sesión de usuario particular, a través de una o más solicitudes. (Atributos Session).

3.1.2.4 application – Beans

Son visibles para todas las páginas JSP y los servlets que forman parte de una aplicación Web. (Atributos de contexto Servlet).

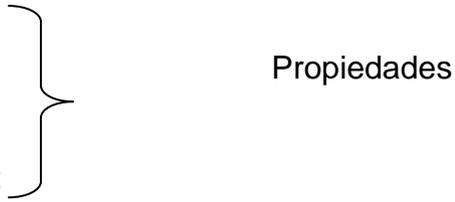
3.1 Arquitectura de un Java Beans

Los componentes de un Java Beans son:

3.2.1 Propiedades o Atributos.

Describen los datos internos, que afectan el funcionamiento y visualización del bean

```
class TestBean {  
  
    private int altura;  
    private String descripción;  
  
    protected boolean fin=false;  
  
}
```



Propiedades

3.2.2 Métodos

Son procedimiento o funciones, generalmente son utilizados para acceder a las propiedades del Beans son dos:

3.2.2.1 Get leer el valor de la propiedad

```
public void setAltura(int i) {  
    altura=i;  
  
}
```

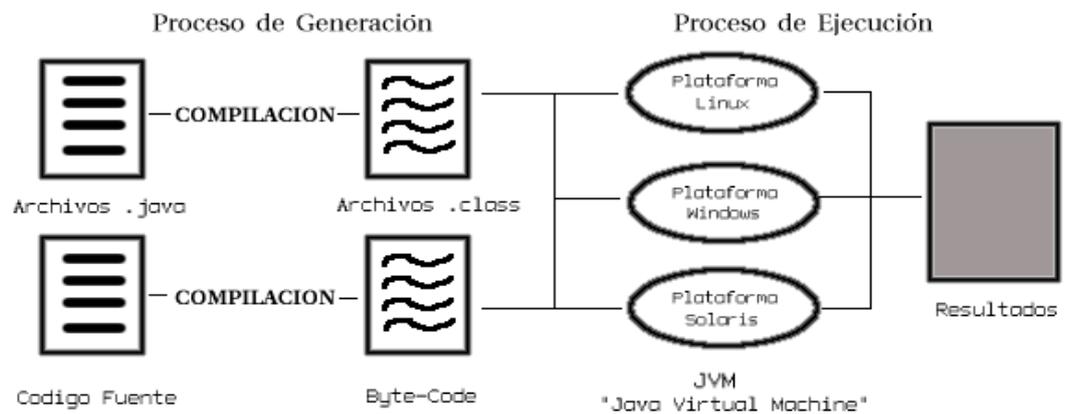
3.2.2.2 Set._ cambiar el valor de la propiedad.

```
public int getAltura() {  
    return altura;}  
}
```

3.2.3 Eventos

Sirven para la comunicación con otros Beans.

3.3. GENERACION .CLASS DE UN JAVA BEANS



Se crea el Código Fuente del Beans con las funciones requeridas por usuario, puede ser creado en JDBuilder o Jcreator, debe ser guardado en la carpeta

Tomcat\webapps\aplicacion-usuario\classes

Con extensión .java

Se compila el código fuente para generar un binario, denominado Byte-Code en Java llevando por extensión .class,

El .class puede ser ejecutado en diversas plataformas obteniendo el mismo resultado, esta ejecución del .class la realiza la máquina virtual llamada "Java Virtual Machine", existiendo diversas para Linux, Solaris, Windows, siendo utilizada por linux windows JDK

3.4 JSP (Java Server Page) y Java Beans

Jsp accede directamente a componentes Java Beans instanciándolos y estableciendo sus propiedades e invocando sus métodos directamente desde la página JSP. Permite desarrollar aplicaciones n-capas donde se separan en lo posible los datos, la lógica del negocio y la lógica de presentación, encapsulando, generalmente, en Beans el acceso a los datos. Su objetivo primordial es convertirse en software reusable, permitiendo uso de la aplicación en cualquier plataforma

3.4.1 Acciones Estándares en JSP

Son tags específicos que afectan el comportamiento en ejecución de la JSP devolviendo una respuesta al cliente, constan de un prefijo y un sufijo además de una serie de atributos. El prefijo es siempre "jsp" en las acciones estándar

Las acciones estándares usadas por Java Beans son:

<jsp: useBean>

<jsp: setProperty>

<jsp: getProperty>

3.4.1.1 jsp:useBean

Esta acción invoca a una Java beans dentro de una pagina JSP, en la página habrá que definir la clase correspondiente, creando los métodos set y get para los atributos definidos

```
<jsp:useBean id="name" scope="page"|request"|session"|application"  
class="queries.dbquery" />
```

id = Nombre del objeto. Es case sensitive

Scope = Alcance del objeto. Indica donde está disponible el objeto. El default es page. Scope

Class= Nombre completo de la clase. Class

3.4.1.2 jsp: setProperty>

Este tag, se usa junto con jsp:useBean y sirve para asignar valores a las propiedades simples del beans. Los valores de las propiedades de un bean, pueden asignarse de las siguientes formas:

- En el momento que se realiza el requerimiento, mediante los valores de parámetros del objeto request.
- En el momento que se realiza el requerimiento, mediante la evaluación de expresiones.
- Escribiendo constantes String.

Su estructura es

```
<jsp:setProperty> name="nombre del Bean" property="Nombre de la propiedad" />
```

donde

name = nombre de beans

property = propiedades a utilizar del beans

3.4.1.3 <jsp: getProperty>

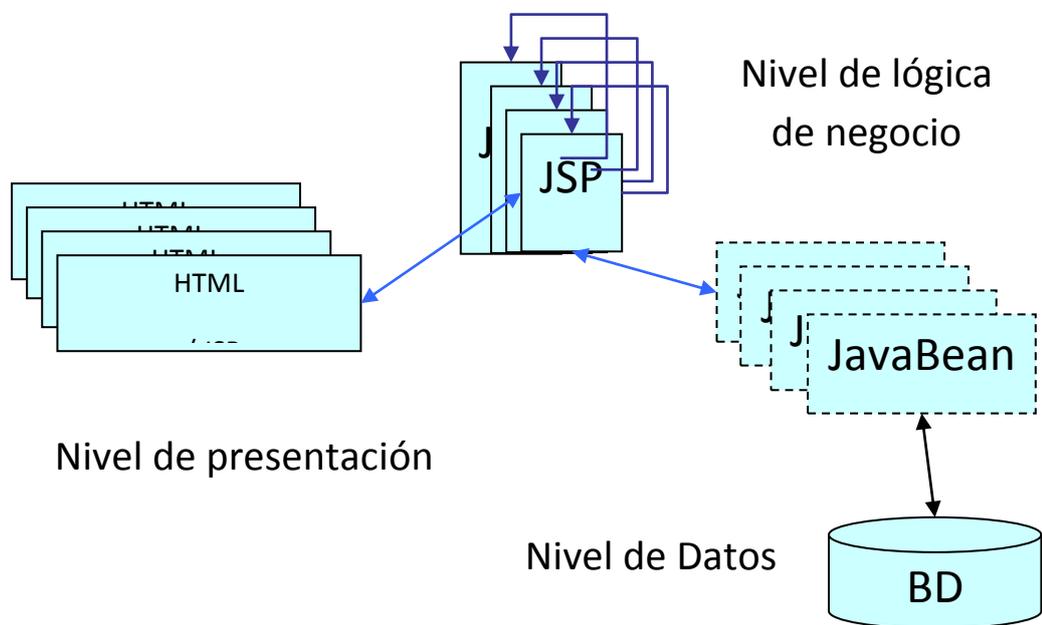
Se utiliza para obtener el valor de las propiedades de un Bean es complementaria del <jsp:setProperty> obteniendo valores de las propiedades del bean, lo convierte a String y lo imprime en la salida.

```
<jsp:getProperty> name="nombre del Bean" property="Nombre de la propiedad" />
```

name = nombre de beans

property = propiedades a utilizar del beans

3.5 Funcionamiento de Bean



El contenedor Web localiza el objeto (bean) usando el id y el scope especificado.

Si encuentra el objeto y se especificó el atributo type, entonces el contenedor Web prueba el objeto hallado, Si la prueba falla, dispara la excepción `java.lang.ClassCastException`, caso contrario ejecuta la propiedad indicada.

3.6 Ejemplo de un Java Beans dentro de una pagina JSP

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
```

```
<html>
```

```
<body>
```

Llamada al Beans

```
<jsp:useBean id="db" scope="request" class="queries.dbquery" />
```

```
<jsp:setProperty name="db" property="*" />
```

```
<%
```

```
try
```

```
{
```

```
    db.conectar(); propiedad del Javabeans
```

```
    rs = db.act_sql(descripcioncli); enviando parámetros al
```

```
Javabeans
```

```
    db.cerrar();propiedad del Javabeans
```

```
    }
```

```
}
```

```
catch(SQLException e) {};
```

```
%>
```

```
</body>
```

```
</html>
```

3.7 Ventajas

- Una de las ventajas más importantes de los Java beans es que estos encapsulan la lógica de la aplicación Web y eliminan el código de scripts incluido en las JSP.
- El uso JavaBeans tiene como resultado un código más simple, fácil de mantener y más legible (para los diseñadores de páginas).
- El uso exclusivo de Beans elimina una gran parte de código Java aunque no es suficiente para evitar todo tipo de código Java en una página JSP
- Se sabe que java es un lenguaje naciente Pero esto también está cambiando rápidamente. Los "Java Beans" son componentes que sirven para expandir las funciones de Java. Con esto, un programador reduce grandemente el tiempo que le toma crear una nueva aplicación. A pesar de su juventud, hoy día hay más aplicaciones interactivas para Internet creadas en Java que en cualquier otro lenguaje
- La portabilidad y función habilidad de Java lo hacen uno de los lenguajes de programación más significativos de la era moderna y la utilización de los Java Beans Programadores y casas de software deben considerarlo para el desarrollo de aplicaciones nuevas, particularmente si la integración con la Internet es importante.

3.8 Desventajas de los Java beans

- Para la construcción JavaBeans se debe tener un conocimiento amplio de la programación en java o base sólidas de este lenguaje.
- La gran desventaja de Java bean es que cumplen una sola función que es encapsula la lógica e negocios por el diseñador, aunque puede comunicarse entre Java beans.

CAPITULO IV

4. DESARROLLO DEL PROTOTIPO BASADOS EN LA METODOLOGÍA ORIENTDA A OBJETOS (UML)

4.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE (ERS)

4.1.1 INTRODUCCIÓN

Este documento detallara la Especificación de Requisitos de software (ERS) para el Sistema de Gestión Inventario de Producción, el mismo que ofrecerá una descripción detallada de lo que hará el Sistema, se basa en la información recolectada del departamento de producción y Sistemas de la Floricola Cultivos Premiun S.A

4.1.2 REQUERIMIENTO

El sistema estará apoyado bajo la descripción de la Especificación de Requisitos de Software estándar IEEE-830.

4.1.3 PROPÓSITO

El objetivo principal de la Especificación de Requisitos de Software es definir en una forma precisa, clara y ordenada las funcionalidades y restricciones que tendrá el Sistema.

El presente documento va dirigido al personal de desarrollo, a las personas que lo pondrán a prueba y a los usuarios finales.

Esta especificación está sujeta a revisiones y modificaciones por parte de las partes implicadas, de acuerdo a sus necesidades, hasta lograr la aprobación y agrado de todas las partes.

4.1.4 AMBITO DEL SISTEMA

Para estar acorde con el avance tecnológico la florícola Cultivos Premiun S.A toma la decisión de implantar un sistema informático para la Gestión Inventario de Producción, será denominado PSGI (Prototipo del Sistema Gestión de Inventario), el cual ofrecerá al comercialización una venta rápida y segura, permitiendo de esta manera conocer la producción actual de cada día, su disponibilidad, estadísticas de la producción , estas actividades que serán necesarias para el desarrollo optimo de su ejecución, logrando así ser competitivos en el mercado nacional e internacional.

4.1.5 DEFINICIONES ACRONIMOS Y ABREVIATURAS

4.1.5.1 DEFINICIONES

Administrador	Es la persona autorizada para extraer, ingresar modificar ,y realizar consultas de la información de usuarios almacenada en el sistema
Usuario	Es la persona autorizada para realizar ingresos, consultas , modificaciones de la información
Comercializador	Es la persona que tendrá acceso al Sistema, para realizar consultas y modificaciones de la información registrada en el Sistema útil en la comercialización

Tabla 4.1.5.1

4.1.5.2 ACRONIMOS

ERS	Especificación de Requisitos de Software
-----	--

Tabla 4.1.5.2

4.1.5.3 ABREVIATURAS

PSGI	Prototipo del Sistema Gestión de Inventario
------	---

Tabla 4.1.5.3

4.1.6 VISION GENERAL DEL DOCUMENTO

El documento presentará una introducción general de la Especificación de Requisitos de Software, acompañada de una descripción general del sistema, con el objetivo de conocer las funciones y restricciones del Sistema

4.1.7 DESCRIPCION GENERAL

Una breve descripción del Sistema de una manera global, las funciones que debe realizar, la información utilizada, las restricciones y otras funciones que afecten a la construcción y desarrollo del sistema

4.2 FUNCION DEL SISTEMA

El PSGI deberá proporcionar las siguientes tareas

- Gestión de Empaque
- Gestión de Clase
- Gestión de Grado
- Gestión Finca
- Gestión Variedad
- Gestión Producto
- Gestión Catalogo.
- Gestión Cliente
- Gestión Nota Venta
- Gestión Usuario
- Búsquedas del Sistema.

A continuación se detallará las tareas y como estarán soportadas en el sistema

4.2.1 Gestión de Empaque

El sistema permitirá realizar las funciones de alta, modificación consulta de Empaque. Para dar de alta un empaque se deberá ingresar el codigo_empaque, en caso de no existir el codigo_empaque, el sistema permitirá el ingreso de los campos:

codigoempaque

descempaque

Para la modificación el sistema permitirá la visualización de los datos de un empaque para ser modificados, los campos:

descempaque

La consulta general permitirá conocer todos los empaques registrados en el Sistema Gestión de Inventario

4.2.2 Gestión Clase

El sistema permitirá realizar las funciones de alta, modificación consulta Clase

Para dar de alta, se ingresara codigoclase en caso de no existir el sistema permitirá el ingreso de:

codigoclase

descclase

En caso de existir el sistema presentará no ingresara el registro un "Código Existente "

Para la modificación el sistema permitirá la visualización de los datos de un Clase para ser modificados, los campos:

descclase

La consulta general permitirá conocer todas las clases registradas en el Sistema Gestión de Inventario

4.2.3 Gestión de Grado

El sistema permitirá realizar las funciones de alta, modificación consulta de Grado

Para dar de alta, se ingresará codigogrado en caso de no existir el sistema permitirá el ingreso de:

codigogrado

dimension

En caso de existir el sistema presentará un mensaje de alerta "Código Existente "

Para la modificación el sistema permitirá la visualización de los datos de un Grado para ser modificados, los campos:

dimension

La consulta general permitirá conocer todos los grados registrados en el Sistema Gestión de Inventario

4.2.4 Gestión Finca

El sistema permitirá realizar las funciones de alta, modificación, consulta de la Finca

Para dar de alta, se ingresara codigofinca en caso de no existir el sistema permitirá el ingreso de:

codigofinca

nombrefinca

direccionfinca

fonofinca

emailfinca

contactofinca

En caso de existir el sistema presentará un mensaje de alerta "Código Existente "

Para la modificación el sistema permitirá la visualización de los datos de una Finca para ser modificados, los campos:

nombrefinca

direccionfinca

fonofinca

emailfinca

contactofinca

La consulta general permitirá conocer todas las fincas registrados en el Sistema Gestión de Inventario

4.2.5 Gestión Variedad

El sistema permitirá realizar las funciones de alta, modificación , consulta de la variedad de la flor.

Para dar de alta, se ingresara codigovariedad en caso de no existir el sistema permitirá el ingreso de:

codigovariedad

nombrevariiedad

observacionvariedad

En caso de existir el sistema presentará un mensaje de alerta "Código Existente "

Para la modificación el sistema permitirá la visualización de los datos de una Variedad para ser modificados, los campos:

nombrevariiedad

observacionvariedad

La consulta general permitirá conocer todas las variedades registrados en el Sistema Gestión de Inventario

4.2.6 Gestión de Producto

El sistema permitirá realizar las funciones de alta, modificación , consulta de la variedad de la flor

Para dar de alta, se ingresara `codigoproducto` en caso de no existir el sistema permitira el ingreso de :

`codigoproducto`

`codigovariedad`

`codigoempaque`

`codigogrado`

`codigofinca`

`codigoclase`

`talloproducto`

`preciounitario`

`stockmax`

`stockmin`

`stockactual`

`resistencia`

En caso de existir el sistema presentará un mensaje de alerta "Código Existente "

Para la modificación el sistema permitirá la visualización de los datos de un Producto para ser modificados, los campos:

codigovariedad

codigoempaque

codigogrado

codigofinca

codigoclase

talloproducto

preciounitario

La consulta general permitirá conocer todos los productos registrados en el Sistema Gestión de Inventario

4.2.7 Gestión Catalogo

El sistema permitirá realizar las funciones de alta, modificación , consulta del catalogo.

El lector de código de barra es el encargado de realizar la respectiva alta.

codigocatalogo

codigobarra

cantidad

fechaingreso

codigousuario

tipo

nota

tiempo

La consulta general permitirá conocer todas las variedades registrados en el Sistema Gestión de Inventario

4.2.8 Gestión Usuario

El sistema permitirá realizar las funciones de alta, modificación, consulta de usuario

Para dar de alta, se ingresara el codigousuario en caso de no existir el sistema permitira el ingreso de :

codigousu

nombreusu

passwordusu

direccionusu

fonousu

emailusu

referencia

tipo

En caso de existir el sistema presentará un mensaje de alerta “Codigo Existente ”

Para la modificación el sistema permitirá la visualización de los datos de un usuario para ser modificados, los campos:

nombreusu

passwordusu

direccionusu

fonousu

emailusu

referencia

tipo

En caso de no existir `codigousu`, el sistema presentará un mensaje de alerta “Codigo_No_Existente”

La consulta general permitirá conocer todos los usuarios registrados en el Sistema Gestión de Inventario

4.2.9 Gestión Cliente

El sistema permitirá realizar las funciones de alta, modificación, consulta de cliente

Para dar de alta, se ingresará el `codigocliente` en caso de no existir el sistema permitirá el ingreso de :

`codigocliente`

`nombrecliente`

`fonocliente`

`emailcliente`

`direcciondestino`

`direccioncobro`

En caso de existir el sistema presentará un mensaje de alerta “Codigo Existente”

Para la modificación el sistema permitirá la visualización de los datos de un Cliente para ser modificados, los campos:

nombrecliente

fonocliente

emailcliente

direcciondestino

direccioncobro

La consulta general permitirá conocer todas los clientes registrados en el Sistema Gestión de Inventario

4.2.10 Gestión Nota de Venta

El sistema permitirá realizar las funciones de alta de Nota Venta para dar de alta, se ingresara el codigoventa caso de no existir el sistema permitira el ingreso de :

codigoventa

codigocliente

codigovendedor

fechaventa

cantidad

nombrevariiedad

codigobarra

En caso de existir el sistema presentará un mensaje de alerta “Codigo Existente ”

4.2.11 Búsquedas de Producto

El usuario podrá realizar búsquedas acerca de las características del producto, accediendo a PSGI desde su navegador, pudiendo ingresar los datos sobre lo que desea averiguar, estos datos pueden combinarse entre:

Variedad

empaquete

grado

finca

clase

tallo

4.2.12 CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios deberán tener conocimiento de computación y manejo de Internet.

4.3 REQUISITOS ESPECIFICOS

En este apartado se describe los requisitos funcionales que serán satisfechos por el

Prototipo, Todos los requisitos aquí expuestos son esenciales, es decir, no sería aceptable si alguno de los requisitos aquí presentados no son resueltos por el prototipo.

4.3.1 REQUISITOS FUNCIONALES

4.3.2 Gestión Usuario

El sistema deberá permitir :

Req(01) Ingresar un nuevo Usuario

Req(02) Modificar Datos Usuario

Req(03)Acceder A Modulo Usuario

4.3.3 Gestión de Empaque

El sistema deberá permitir :

Req(04) Ingresar un nuevo Empaque

Req(05)Modificar Empaque

Req(06)Consultar Empaque

4.3.4 Gestión de Clase

El sistema deberá permitir :

Req(07) Ingresar un nuevo Clase

Req(08)Modificar Clase

Req(09)Consultar Clase

4.3.5 Gestión de Grado

El sistema deberá permitir :

Req(10) Ingresar un nuevo Grado

Req(11) Modificar Grado

Req(12) Consultar Grado

4.3.6 Gestión Finca

El sistema deberá permitir :

Req(13) Ingresar un nuevo Finca

Req(14) Modificar Finca

Req(15) Consultar Finca

4.3.7 Gestión Variedad

El sistema deberá permitir :

Req(16) Ingresar un nuevo Variedad

Req(17) Modificar Variedad

Req(18)Consultar Variedad

4.3.8 Gestión Producto

El sistema deberá permitir :

Req(19) Ingresar un nuevo Producto

Req(20)Modificar Producto

Req(21)Consultar Producto

Req(22) El sistema permitirá imprimir el código de barra (o etiqueta) de un producto seleccionado dependiendo de la cantidad necesitada por el mismo

Req(23) El sistema permitira realizar búsquedas del producto

4.3.9 Gestión Catalogo.

El sistema deberá permitir :

Req (24) El sistema permitirá el ingreso de los datos de un producto a través de un lector de código de barra.

Req(25) El sistema permitirá consultar disponibilidad diaria de la flor

4.3.10 Gestión Cliente

El Sistema debe permitir:

Req(26) Ingresar un nuevo Cliente

Req(27)Modificar Cliente

Req(28)Consultar Cliente

4.3.11 Gestión Nota de Venta

Req(29) Ingresar nueva nota de venta

4.3.12 Búsquedas de producto

Req(30) Realizar búsquedas acerca de las características del producto, accediendo a PSGI desde su navegador,

Req(31)Ingresar los datos sobre lo que desea averiguar, estos datos pueden combinarse entre:

Variedad

empaque

grado

finca

clase

tallo

4.4 REQUISITOS DE INTERFACES EXTERNAS

4.4.1 Interfaces de Usuario

La interfaz del sistema debe ser orientado a ventanas y el manejo del programa se realizará a través del teclado, lector de barras y ratón

4.4.2 Interfaces Software

PSGI tiene una interfaz cliente que brindará su servicio a un sistema cliente.

4.4.3 Interfaces de Comunicación

PSGI está orientado para funcionar en línea sobre Internet

4.5 REQUISITOS DE DESARROLLO

El ciclo de vida elegido para desarrollar el prototipo es el interactivo incremental, por su flexibilidad para incorporar cambios.

4.6 REQUISITOS TECNOLÓGICOS DE HADWARE Y SOTFWARE

	Cliente	Servidor
Sistema Operativo	Windows XP	Linux o Windows XP en adelante
Memoria (MB)	64 o superior	128 o superior
Procesador	Pentium III o superior	Pentium III o superior
Velocidad	800 Mhz o superior	800 Mhz o superior
Conexión a Internet	Dial Up como mínimo	128 Kbps simétrico
Browser	IE5.5 o superior (para windows)	IE 5.5 o superior (para windows)
Programas instalados	_____	Tomcat

		JVM
Dispositivos adicionales		CD ROM

4.7 Atributos

4.7.1 Seguridad

El sistema permitirá el manejo de la información únicamente a las personas autorizadas, es decir al administrador, usuarios y comercializador los mismos que poseerán un login y un password, la información será verificada por el prototipo para comprobar que se trata de un usuario registrado en el prototipo. Si el usuario introducido corresponde a un no identificado, no podrá ingresar al prototipo.

4.8 FASE DE ANALISIS

4.8.1 Formularios De Entrada

4.8.1.1 Ingreso de Datos Grado

DATOS INFORMATIVOS DE GRADO

4.8.1.2 Ingreso de Datos Clase

DATOS INFORMATIVOS DE CLASE

4.8.1.3 Ingreso de Datos Empaque

DATOS INFORMATIVOS DE EMPAQUE

4.8.1.4 Ingreso de Datos Variedad

DATOS INFORMATIVOS DE VARIEDAD

4.8.1.5 Ingreso de Datos Finca

DATOS INFORMATIVOS DE FINCA	
Código	
Nombre	
Dirección	
E-mail	

4.8.1.6 Datos informativos Producto

DATOS INFORMATIVOS DE PRODUCTO	
Código barra	
Grado	
Clase	
Empaque	
Variedad	
Finca	
Tallo	

Ingreso de Datos Usuario

Datos informativos Usuario

4.8.1.7 Datos informativos Usuario

DATOS INFORMATIVOS DE USUARIO	
Código	
Nombre	
Password	
Dirección	

4.9. CASOS DE USO EXPANDIDOS

4.9.1 MODELO DE CASO DE USO DEL ACTOR ADMINISTRADOR

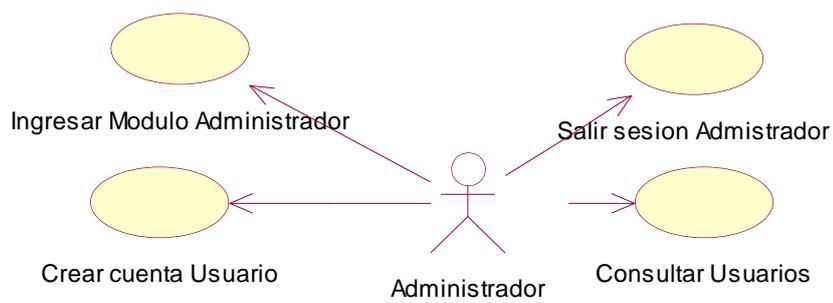


FIGURA 4.9.1.1 INGRESAR MODULO AMINISTRADOR

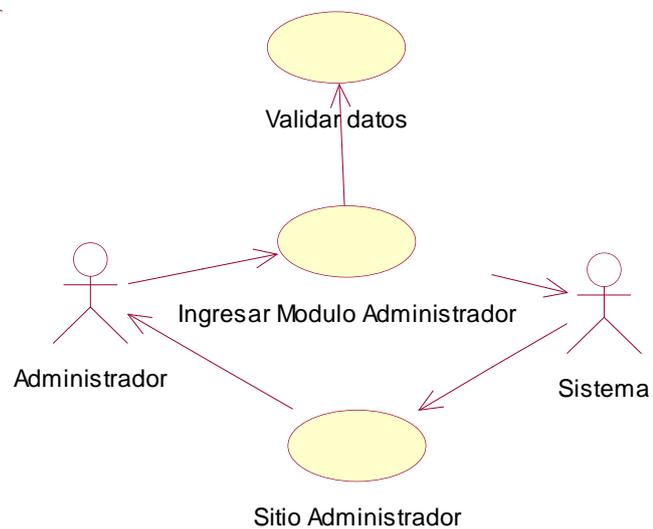


FIGURA 4.9.1.1 CASO DE USO INGRESAR MODULO ADMINISTRADOR

Nombre caso de Uso: Ingresar Modulo Administrador

Propósito: Permitir a un Administrador Ingresar al sitio de Administradores.

Visión General: El Administrador solicita la operación Ingresar Modulo Administrador, el sistema presenta ventana de ingreso de modulo del Administrador, el administrador ingresa su login y password el sistema valida los datos si son correctos el administrador podrá acceder al Sitio de Administradores, el sistema confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El Administrador solicita la operación Ingresar Modulo Administrador.
2. El sistema presenta ventana de Ingreso a Modulo Administrador.
3. El Administrador ingresa su login y password.

4. El sistema valida los datos ingresados por el administrador y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe ventana de ingreso a Modulo de administradores, termina caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.1.2 CREAR CUENTA USUARIO

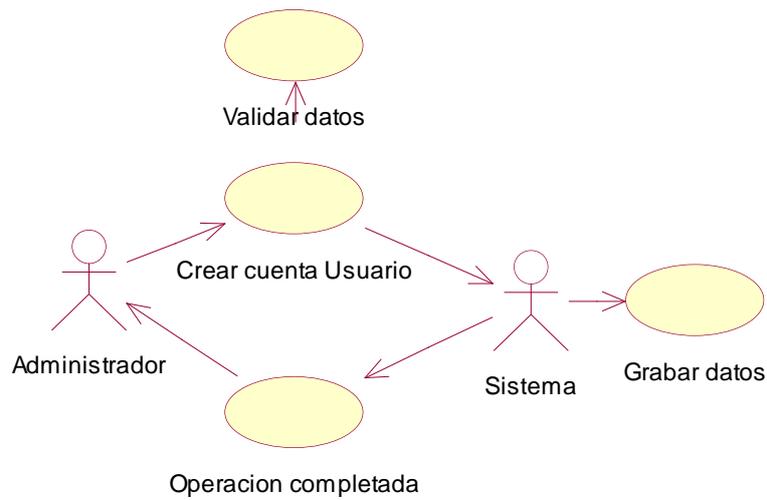


FIGURA 4.10.1.2 CASO DE USO CREAR CUENTA USUARIO

Nombre caso de Uso: Crear Cuenta Usuario

Propósito: Crear una nueva cuenta de Usuario para el sistema.

Visión General: El Administrador solicita la operación Crear Cuenta Usuario, el sistema presenta formulario de Registro de un nuevo Usuario, el administrador ingresa uno a uno los datos necesarios para la creación de una nueva cuenta de Usuario, el sistema almacena los datos del usuario y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Administrador solicita la operación Crear Cuenta Usuario.
2. El sistema presenta formulario de registro de una nueva cuenta de usuario.
3. El Administrador ingresa los datos solicitados por el sistema.
4. El sistema almacena los datos del usuario y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de una nueva cuenta de usuario, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.1.3 CONSULTAR USUARIOS

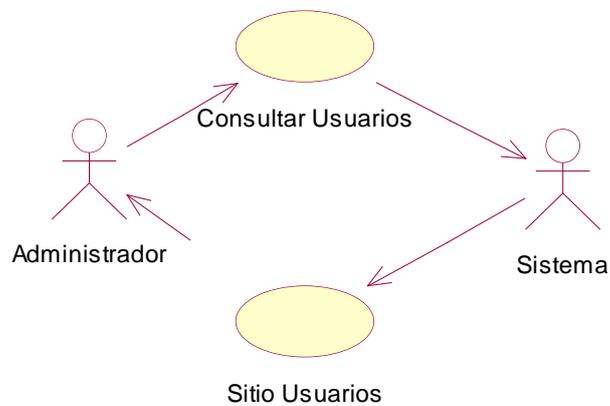


FIGURA 4.10.1.3 CASO DE USO CONSULTAR USUARIOS

Nombre caso de Uso: Consultar Usuarios

Propósito: Observar los usuarios registrados en el sistema.

Visión General: El Administrador solicita la operación Consultar Usuarios, el sistema presenta una lista de los usuarios registrados en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Administrador solicita la operación Consultar Usuarios.
2. El sistema presenta una lista de los usuarios en el sistema.
3. El sistema confirma operación.

Cursos Alternativos

- Curso típico 2: No existe usuarios registrados en el sistema, termina el caso de uso.

4.10.1.4 SALIR SESION ADMINISTRADOR



FIGURA 4.10.1.4 CASO DE USO SALIR SESION ADMINISTRADOR

Nombre caso de Uso: Salir Sesión Administrador.

Propósito: Permitir a un Administrador abandonar el Sitio Administrador.

Visión General: El Administrador solicita la operación Salir Sesión Administrador, el sistema confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Administrador solicita la operación Salir Sesión Administrador.
2. El sistema confirma operación.

FIGURA 4.10.2 MODELO DE CASO DE USO DEL ACTOR USUARIO

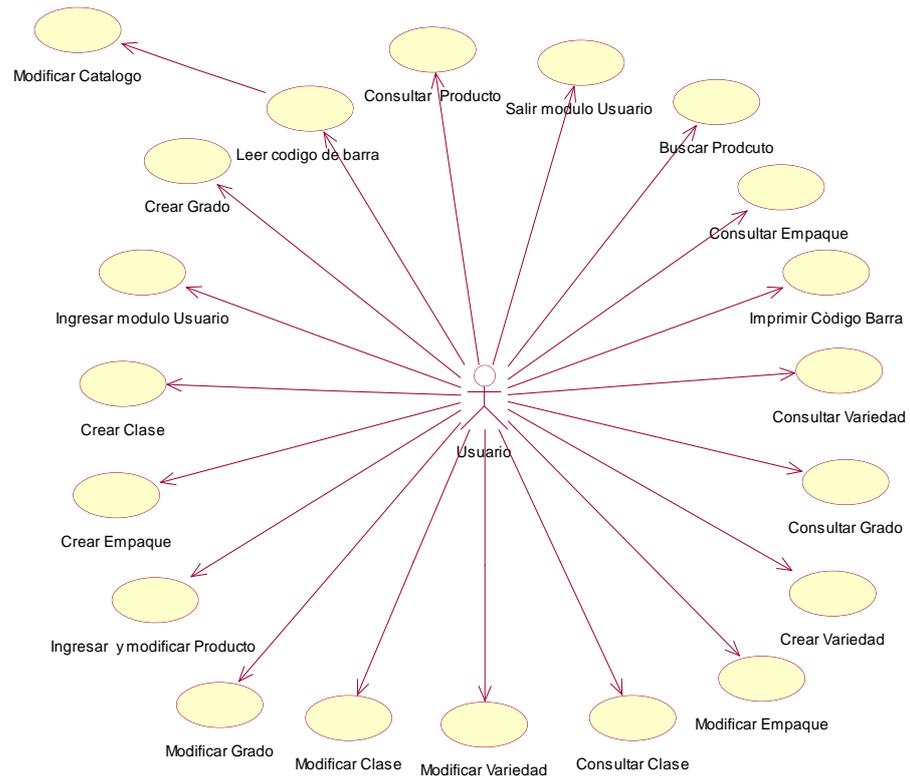


FIGURA 410.2 CASOS DE USO ACTOR USUARIO

4.10.2.1 INGRESAR MODULO USUARIO

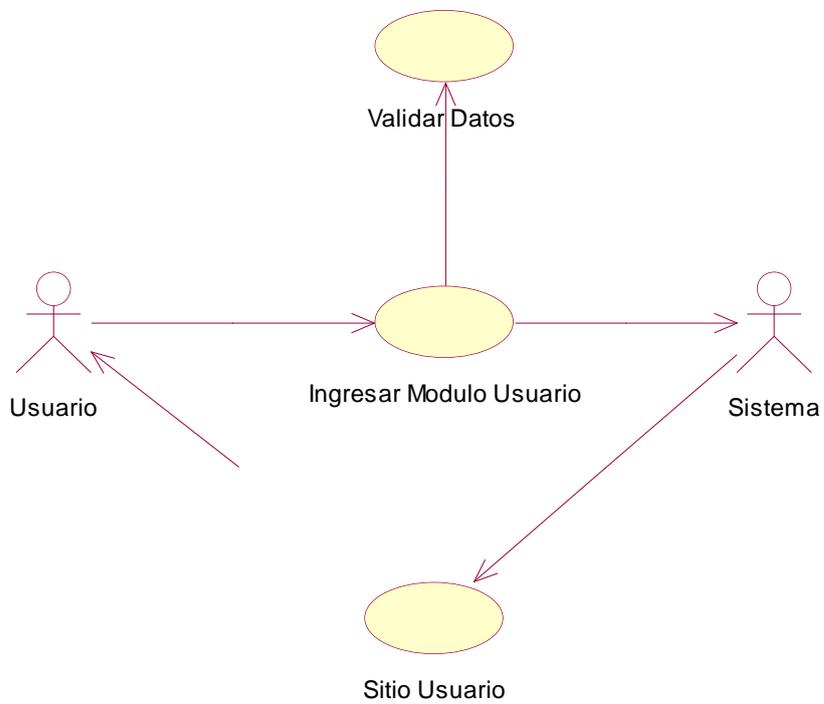


FIGURA 4.10.2.1 CASO DE USO INGRESAR MODULO USUARIO

Nombre caso de Uso: Ingresar Modulo Usuario.

Propósito: Permitir a un Usuario Ingresar al sitio de Usuarios.

Visión General: El Usuario solicita la operación Ingresar Modulo Usuario, el sistema presenta ventana de ingreso de modulo del Usuario, el usuario ingresa su login y password el sistema valida los datos si son correctos el usuario podrá acceder al Sitio de Usuarios, el sistema confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El usuario solicita la operación Ingresar Modulo Usuario.

2. El sistema presenta ventana de Ingreso a Modulo Usuario.
3. El Usuario ingresa su login y password
4. El sistema valida los datos ingresados por el usuario y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe ventana de ingreso a Modulo de usuario, termina caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.2.2 CREAR GRADO

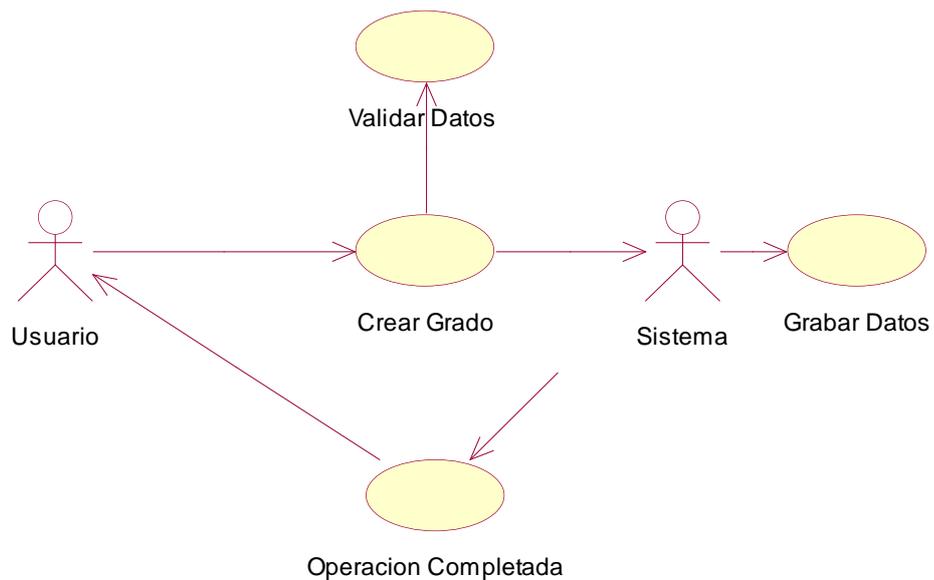


FIGURA 4.10.2.2 CASO DE USO CREAR GRADO

Nombre caso de Uso: Crear Grado

Propósito: Crear un nuevo Grado para el sistema.

Visión General: El Usuario solicita la operación Crear Grado, el sistema presenta formulario de Registro de un nuevo Grado, el usuario ingresa los campos(código grado, dimensión) al sistema, el sistema almacena los datos de grado y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Crear Grado.
2. El sistema presenta formulario de registro de un nuevo grado.
3. El usuario ingresa los campos (código grado, dimensión) al sistema.
4. El sistema almacena los datos de grado y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de un nuevo grado, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.2.3 MODIFICAR GRADO

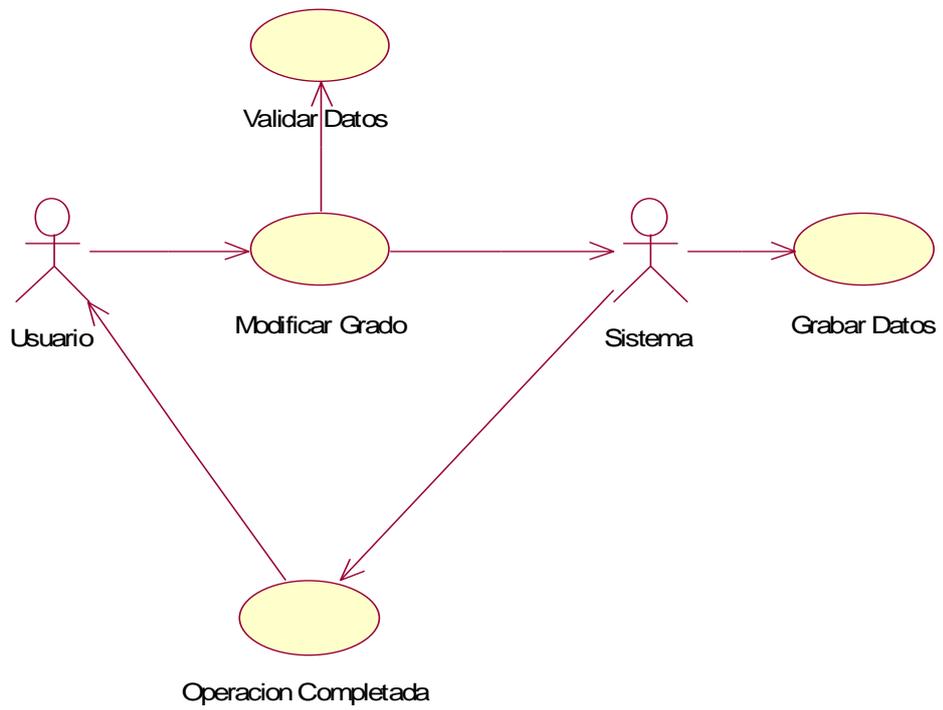


FIGURA 4.10.2.3 CASO DE USO MODIFICAR GRADO

Nombre caso de Uso: Modificar Grado

Propósito: Modificar Grado.

Visión General: El Usuario solicita la operación Modificar Grado, el sistema presenta formulario de Edición de Grado, el usuario podrá modificar los campos(código grado, dimensión) de grado, el sistema almacena los datos modificados de grado y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Modificar Grado.
2. El sistema presenta formulario edición de grado.
3. El usuario tendrá la opción de modificar los campos (código grado, dimensión).
4. El sistema almacena los datos modificados y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.2.4 CONSULTAR GRADO

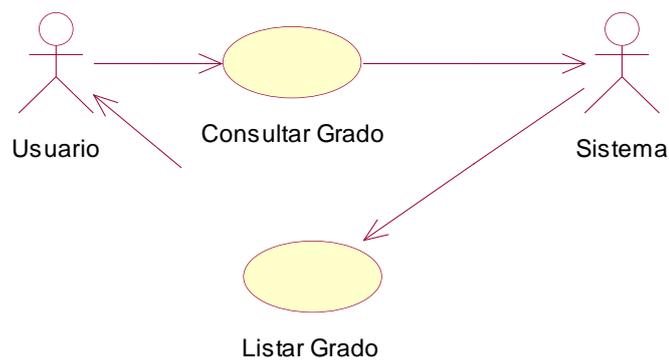


FIGURA 4.10.2.4 CASO DE USO CONSULTAR GRADO

Nombre caso de Uso: Consultar Grado

Propósito: Observar cuales son los grados registrados en el sistema.

Visión General: El Usuario solicita la operación Consultar Grado, el sistema presenta la lista de grados registrados en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Consultar Grado.
2. El sistema presenta la lista de grados registrados en el sistema.

3. El sistema confirma operación.

Cursos Alternativos

- Curso típico 2: No existe grados registrados, termina el caso de uso.

4.10.2.5 CREAR CLASE

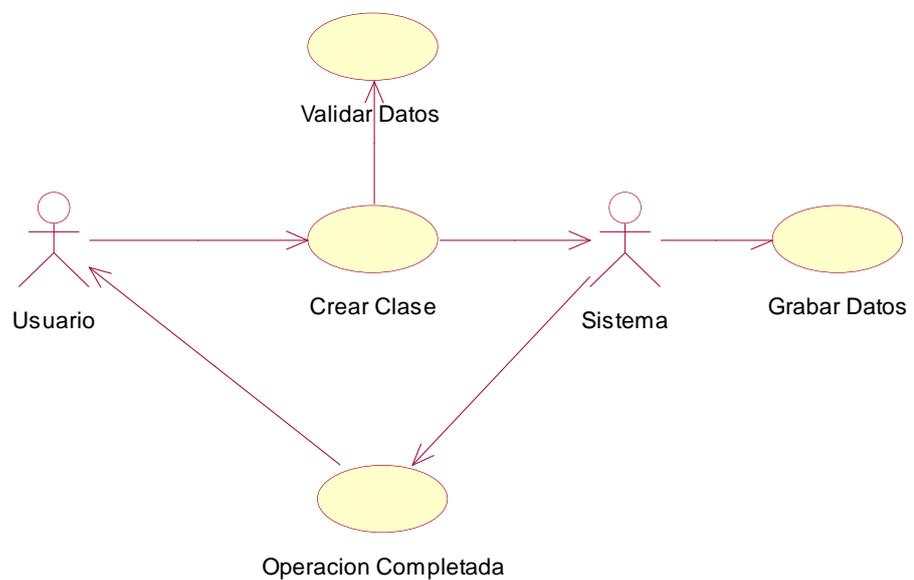


FIGURA 4.10.2.5 CASO DE USO CREAR CLASE

Nombre caso de Uso: Crear Clase

Propósito: Crear una nueva Clase para el sistema.

Visión General: El Usuario solicita la operación Crear Clase, el sistema presenta formulario de Registro de una nueva Clase, el

usuario ingresa los campos(código clase, descripción clase) al sistema, el sistema almacena los datos de clase y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Crear Clase.
2. El sistema presenta formulario de registro de una nueva clase.
3. El usuario ingresa los campos (código clase, descripción clase) al sistema.
4. El sistema almacena los datos de clase y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de una nueva clase, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.2.6 MODIFICAR CLASE

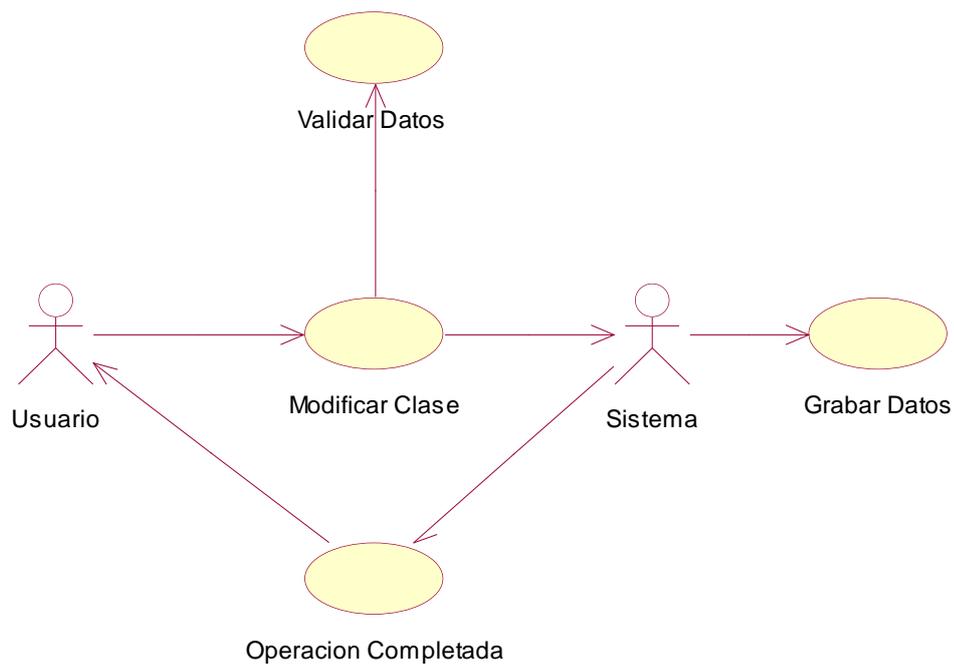


FIGURA 4.10.2.6 CASO DE USO MODIFICAR CLASE

Nombre caso de Uso: Modificar Clase

Propósito: Modificar Clase.

Visión General: El Usuario solicita la operación Modificar Clase, el sistema presenta formulario de Edición de Clase, el usuario podrá modificar los campos(código clase, descripción clase) de clase, el sistema almacena los datos modificados de clase y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Modificar Clase.
2. El sistema presenta formulario edición de clase.
3. El usuario tendrá la opción de modificar los campos (código clase, descripción clase).
4. El sistema almacena los datos modificados y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.2.7 CONSULTAR CLASE

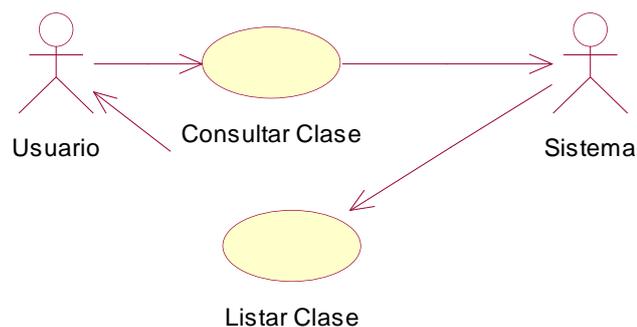


FIGURA 4.10.2.7 CASO DE USO CONSULTAR CLASE

Nombre caso de Uso: Consultar Clase.

Propósito: Observar cuales son las clases registrados en el sistema.

Visión General: El Usuario solicita la operación Consultar Clase, el sistema presenta la lista de clases registrados en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1 El caso de uso comienza cuando el Usuario solicita la operación Consultar Clase.

2 El sistema presenta la lista de clases registrados en el sistema.

3 El sistema confirma operación.

Cursos Alternativos

- Curso típico 2: No existe clases registrados, termina el caso de uso.

4.10.2.8 CREAR EMPAQUE

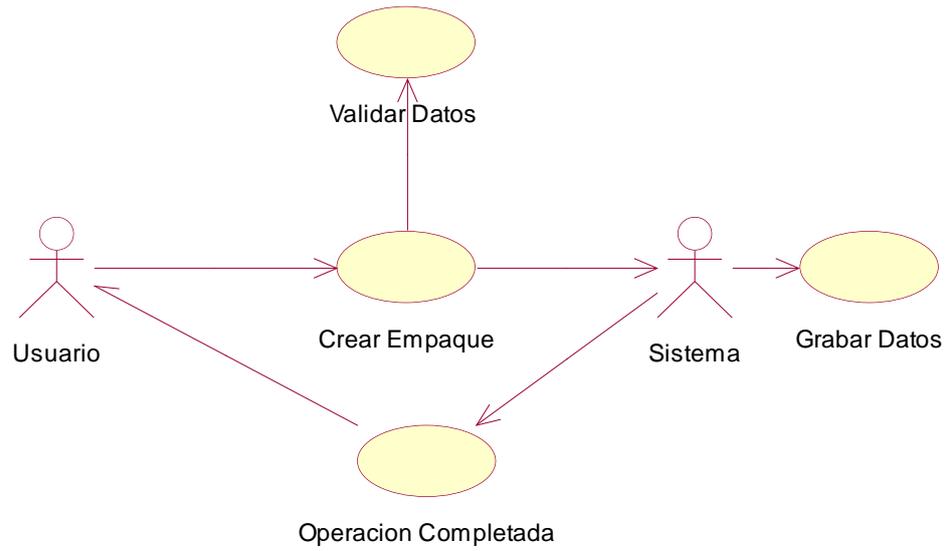


FIGURA 4.10.2.8 CASO DE USO CREAR EMPAQUE

Nombre caso de Uso: Crear Empaque

Propósito: Crear un nuevo Empaque.

Visión General: El Usuario solicita la operación Crear Empaque, el sistema presenta formulario de Registro de un nuevo Empaque, el usuario ingresa los campos(código empaque, descripción empaque) al

sistema, el sistema almacena los datos de empaque y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Crear Empaque.
2. El sistema presenta formulario de registro de un nuevo empaque.
3. El usuario ingresa los campos (código empaque, descripción empaque) al sistema.
4. El sistema almacena los datos de empaque y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de un nuevo empaque, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso,

4.10.2.9 MODIFICAR EMPAQUE

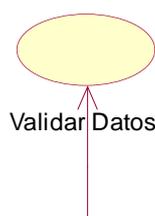


FIGURA 4.10.2.9 CASO DE USO MODIFICAR EMPAQUE

Nombre caso de Uso: Modificar Empaque

Propósito: Modificar Empaque.

Visión General: El Usuario solicita la operación Modificar Empaque, el sistema presenta formulario de Edición de Empaque, el usuario podrá modificar los campos(código empaque, descripción empaque) de empaque, el sistema almacena los datos modificados de empaque y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Modificar Empaque.
2. El sistema presenta formulario edición de empaque.
3. El usuario tendrá la opción de modificar los campos (código empaque, descripción empaque).
4. El sistema almacena los datos modificados y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.2.10 CONSULTAR EMPAQUE

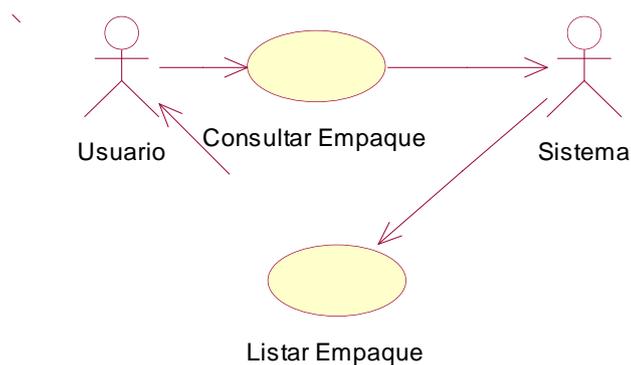


FIGURA 4.10.2.10 CASO DE USO CONSULTAR EMPAQUE

Nombre caso de Uso: Consultar Empaque.

Propósito: Observar cuales son los empaque registrados en el sistema.

Visión General: El Usuario solicita la operación Consultar Empaque, el sistema presenta la lista de empaques registrados en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Consultar Empaque.
2. El sistema presenta la lista de empaques registrados en el sistema.
3. El sistema confirma operación.

Cursos Alternativos

- Curso típico 2: No existe empaques registrados, termina el caso de uso.

4.10.2.11 CREAR VARIEDAD

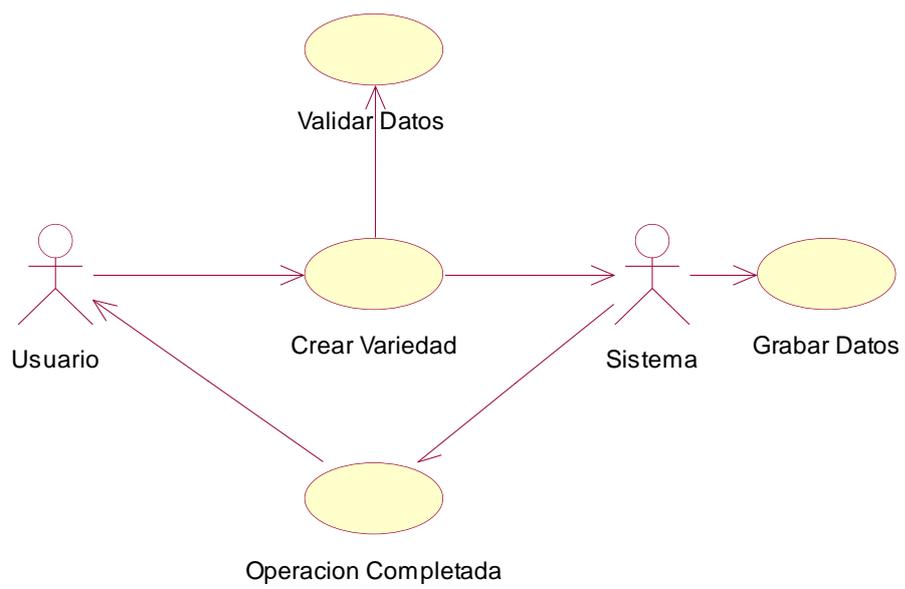


FIGURA 4.10.2.11 CASO DE USO CREAR VARIEDAD

Nombre caso de Uso: Crear Variedad

Propósito: Crear una nueva Variedad.

Visión General: El Usuario solicita la operación Crear Variedad, el sistema presenta formulario de Registro de una nueva Variedad, el usuario ingresa los campos(código variedad, nombre) al sistema, el sistema almacena los datos de variedad y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Crear Variedad.
2. El sistema presenta formulario de registro de una nueva variedad.
3. El usuario ingresa los campos (código variedad, nombre, tallo) al sistema.
4. El sistema almacena los datos de variedad y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de una nueva variedad, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.2.12 MODIFICAR VARIEDAD

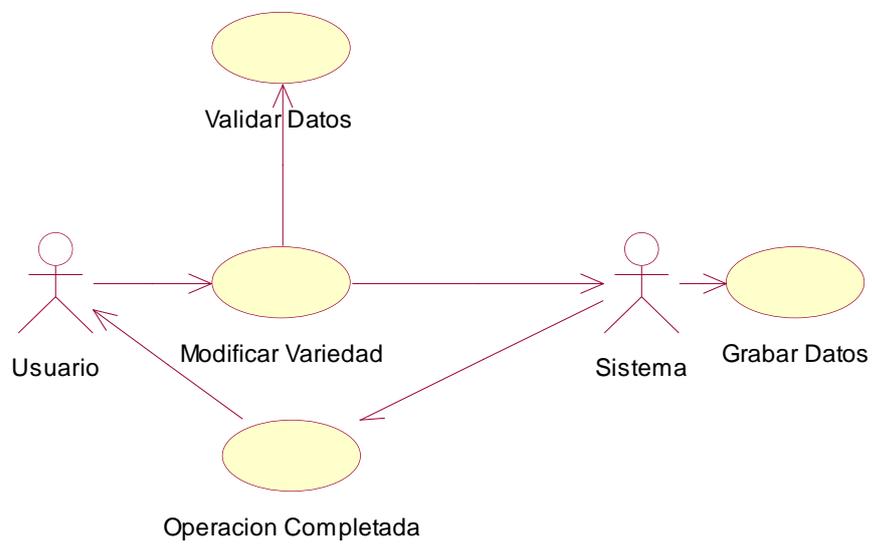


FIGURA 4.10.2.12 CASO DE USO MODIFICAR VARIEDAD

Nombre caso de Uso: Modificar Variedad

Propósito: Modificar Variedad.

Visión General: El Usuario solicita la operación Modificar Variedad, el sistema presenta formulario de Edición de Variedad, el usuario podrá modificar los campos(código variedad, nombre) de variedad, el sistema almacena los datos modificados de variedad y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Modificar Variedad.
2. El sistema presenta formulario edición de variedad.
3. El usuario tendrá la opción de modificar los campos (código variedad,nombre).
4. El sistema almacena los datos modificados y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.

- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.2.13 CONSULTAR VARIEDAD

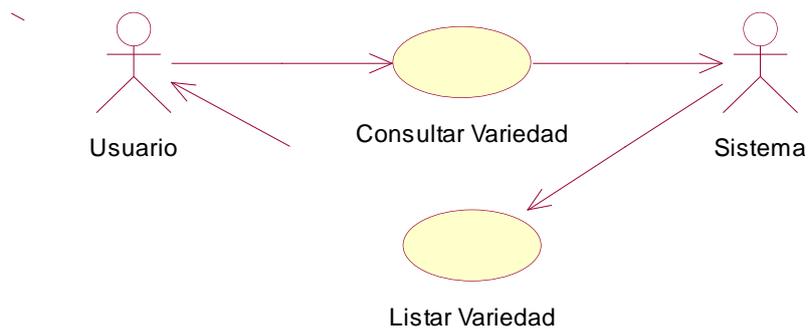


FIGURA 4.10.2.13 CASO DE USO CONSULTAR VARIEDAD

Nombre caso de Uso: Consultar Variedad

Propósito: Observar cuales son las variedades registrados en el sistema.

Visión General: El Usuario solicita la operación Consultar Variedad, el sistema presenta la lista de variedades registrados en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Consultar Variedad.
2. El sistema presenta la lista de variedades registradas en el sistema.
3. El sistema confirma operación.

Cursos Alternativos

- Curso típico 2: No existe variedades registrados, termina el caso de uso.

4.10.2.14 INGRESAR Y MODIFICAR PRODUCTO

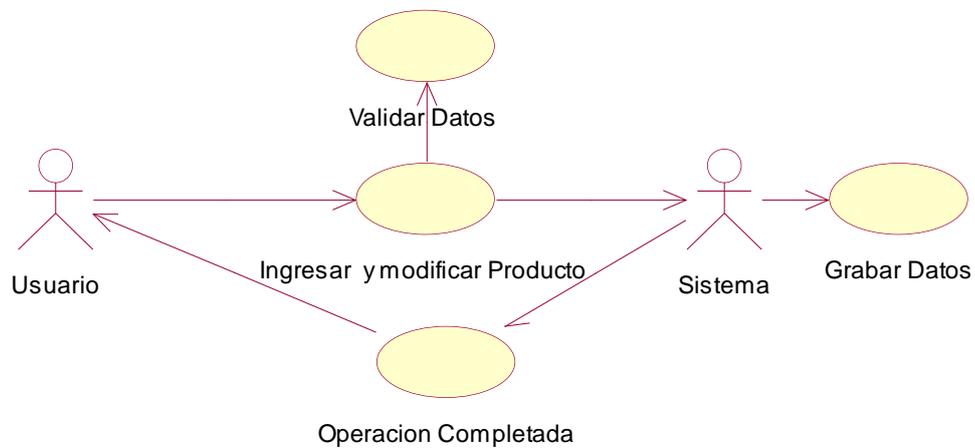


FIGURA 4.10.2.14 CASO DE USO INGRESAR Y MODIFICAR PRODUCTO

Nombre caso de Uso: Ingresar Y Modificar Producto.

Propósito: Que el sistema permita Ingresar y Modificar Producto.

Visión General: El Usuario solicita la operación Ingresar Producto, el sistema presenta formulario de Creación de producto, el usuario ingresa los campos(código variedad, nombre) al sistema, para modificar un producto selecciona la opción editar el usuario podrá modificar los campos () de producto, el sistema almacena los datos de producto y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Ingresar Producto.
2. El sistema presenta formulario de creación de producto.
3. El usuario ingresa los campos (código variedad, nombre, tallo) al sistema.
4. Para modificar producto selecciona la opción enlace.
5. El sistema presenta formulario de Edición de producto.
6. El usuario podrá modificar los campos () de producto.
7. El sistema almacena la información de los productos y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Curso típico 3: Los datos no están bien ingresados, termina caso de uso.

4.10.2.15 LEER CODIGO DE BARRA

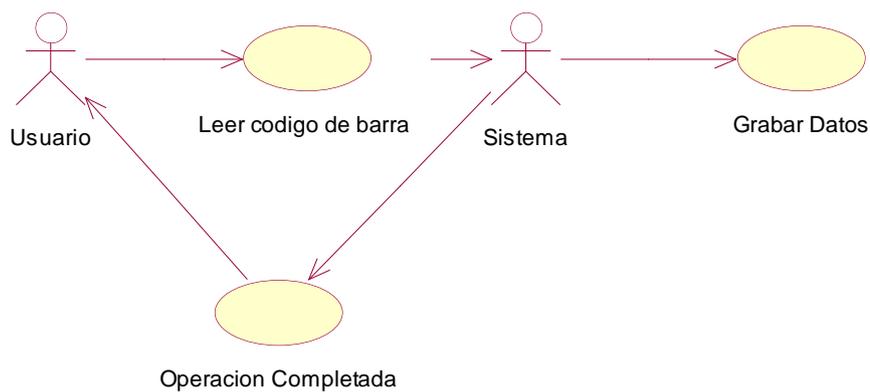


FIGURA 4.10.2.15 CASO DE USO LEER CODIGO DE BARRA

Nombre caso de Uso: Leer Código de barra.

Propósito: Que el sistema permita Ingresar el código de barra con la información respectiva de un producto.

Visión General: El Usuario solicita operación leer código de barra, el sistema presenta formulario de ingreso de código de barra, el usuario lee la etiqueta con el lector de código de barra e ingresa la información

correspondiente al producto y el sistema presenta operación completada .

Tipo: Primario real

Referencias:

Curso Típico de eventos:

8. El caso de uso comienza cuando el Usuario solicita la operación leer código de barra.
9. El sistema presenta formulario de leer código de barra.
10. El usuario ingresa el código de barra a través del lector al sistema.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.

4.10.2.16 IMPRIMIR CODIGO DE BARRA

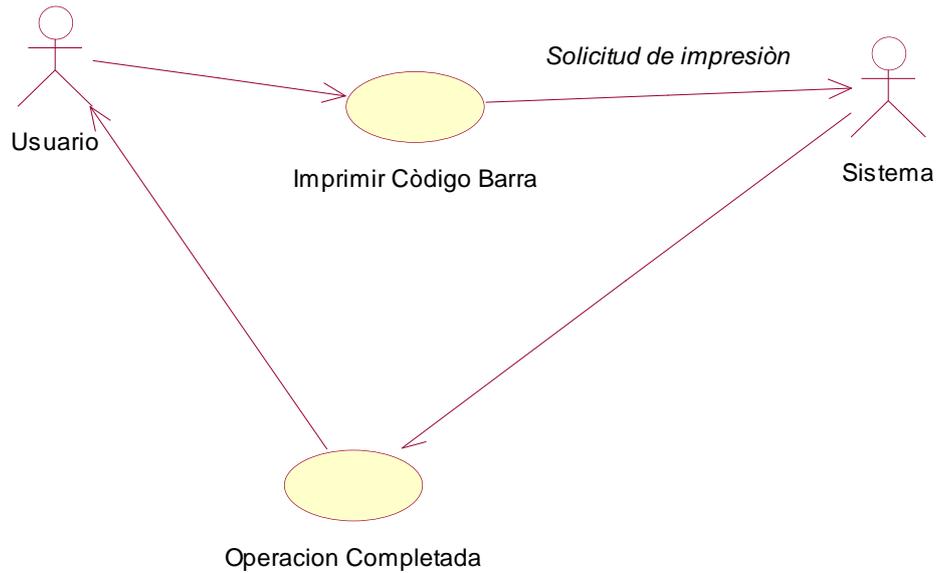


FIGURA 4.10.2.16 CASO DE USO IMPRIMIR CODIGO DE BARRA

Nombre caso de Uso: Imprimir Código de barra.

Propósito: Que el sistema permita imprimir el código de barra con la información respectiva de un producto.

Visión General: El Usuario solicita operación imprimir código de barra, el sistema presenta formulario donde el usuario puede visualizar la lista de producto que puede imprimir, el usuario selecciona y el sistema presenta operación completada.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

11. El caso de uso comienza cuando el Usuario solicita la operación imprimir código de barra.
12. El sistema presenta formulario de imprimir código de barra.

13. El usuario selecciona el producto que desea imprimir y el sistema imprime.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Cuando no exista papel en la impresora

4.10.2.17 BUSCAR PRODUCTO

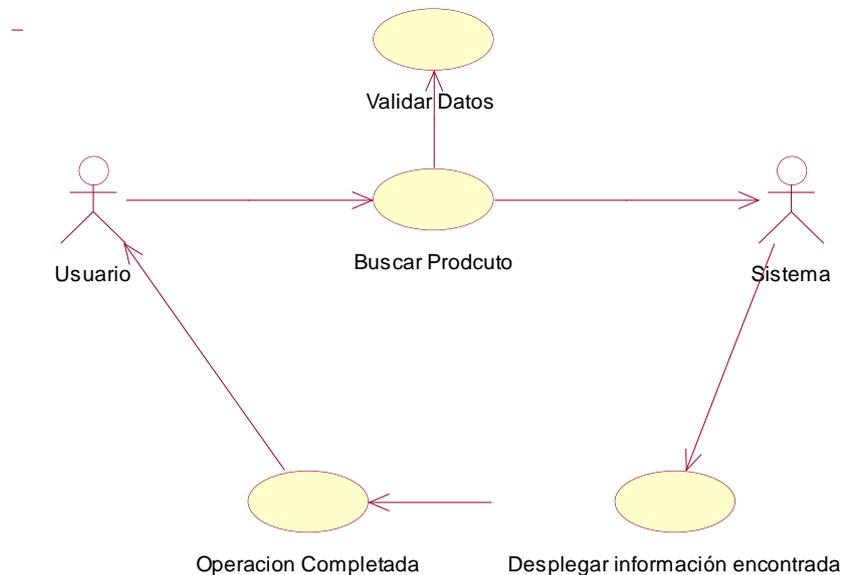


FIGURA 4.10.2.17 CASO DE BUSCAR PRODUCTO

Nombre caso de Uso: Buscar Producto

Propósito: Que el sistema permita Buscar Producto dependiendo a la información enviada por el usuario.

Visión General: El Usuario solicita Buscar producto el sistema, presenta formulario donde ingresa el parámetro por el cual desea buscar y el sistema despliega la información y termina el caso de uso

Tipo: Primario real

Referencias:

Curso Típico de eventos:

14. El caso de uso comienza cuando el Usuario solicita la operación buscar producto.
15. El sistema presenta formulario de buscar producto.
16. El usuario ingresa el parámetro por el cual desea buscar y el sistema presenta la información que encontró.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- No exista información

4.10.2.18 SALIR SESION USUARIO

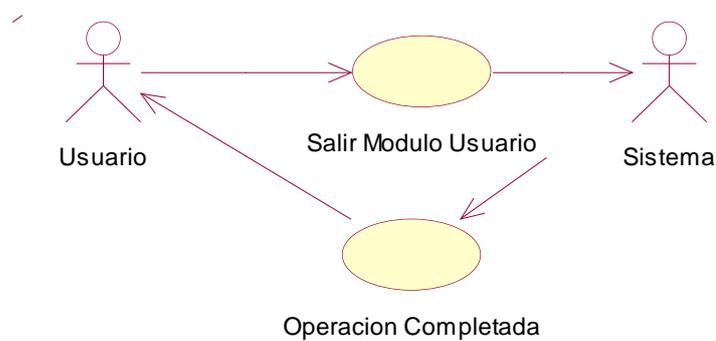


FIGURA 4.10.2.18 CASO DE USO SALIR SESION USUARIO

Nombre caso de Uso: Salir Sesión Usuario.

Propósito: Permitir a un Usuario abandonar el Sitio Usuario.

Visión General: El Usuario solicita la operación Salir Sesión Usuario, el sistema confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Usuario solicita la operación Salir Sesión Usuario.
2. El sistema confirma operación.

4.10.3 MODELO DE CASO DE USO DEL ACTOR COMERCIALIZADOR

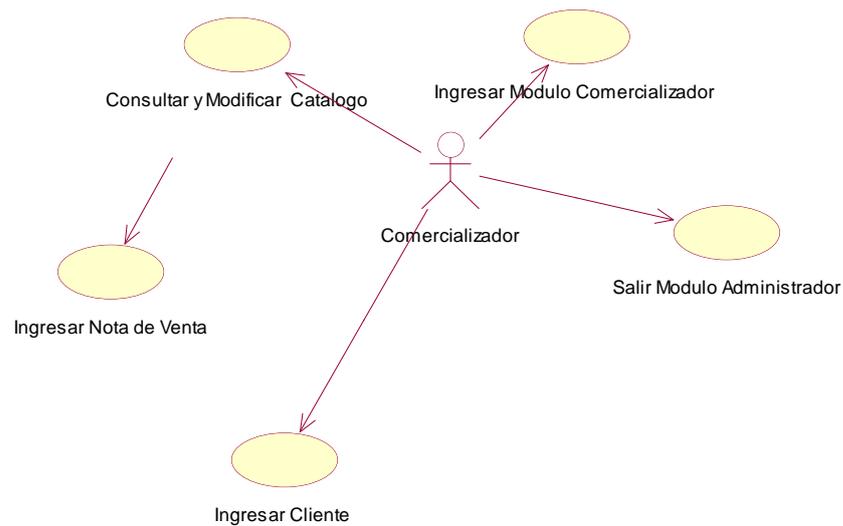


FIGURA 4. 10.3 CASO DE USO DE ACTOR COMERCIALIZADOR

4.10.3.1 INGRESAR MODULO COMERCIALIZADOR

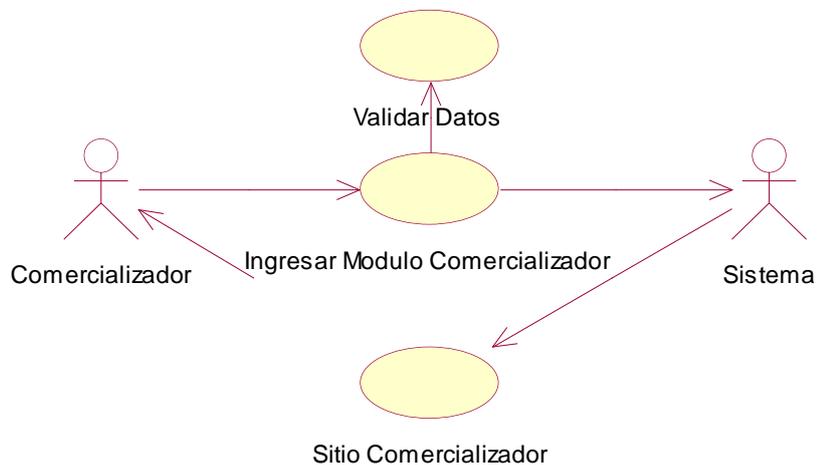


FIGURA 4.10.3.1 INGRESAR MODULO COMERCIALIZADOR

Nombre caso de Uso: Ingresar Modulo Comercializador.

Propósito: Permitir a un Comercializador Ingresar al sitio de Comercializadores.

Visión General: El Comercializador solicita la operación Ingresar Modulo Comercializador, el sistema presenta ventana de ingreso de modulo del Comercializador, el comercializador ingresa su login y password el sistema valida los datos si son correctos el

comercializador podrá acceder al Sitio de Comercializadores, el sistema confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El Comercializador solicita la operación Ingresar Modulo Comercializador.
2. El sistema presenta ventana de Ingreso a Modulo Comercializador.
3. El comercializador ingresa su login y password.
4. El sistema valida los datos ingresados por el comercializador y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe ventana de ingreso a Modulo de comercializador, termina caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.3.2 INGRESAR CLIENTE

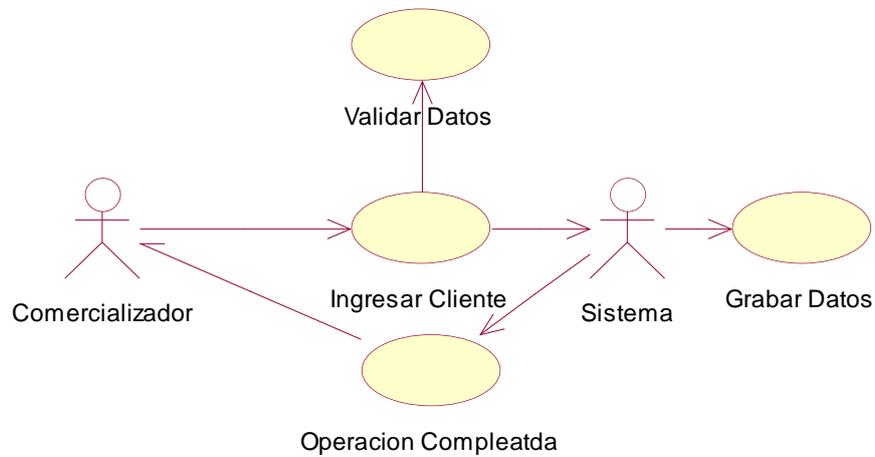


FIGURA 4.10.3.2 CASO DE USO INGRESAR CLIENTE

Nombre caso de Uso: Ingresar Cliente

Propósito: Ingresar un nuevo Cliente.

Visión General: El Comercializador solicita la operación Ingresar Cliente, el sistema presenta formulario de Registro de un nuevo Cliente, el comercializador ingresa los campos() al sistema, el sistema almacena los datos de cliente y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el comercializador solicita la operación Ingresar Cliente.
2. El sistema presenta formulario de registro de un nuevo cliente.
3. El comercializador ingresa los campos () al sistema
4. El sistema almacena los datos de cliente y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario de registro de un nuevo cliente, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.3.5 INGRESAR NOTA DE VENTA

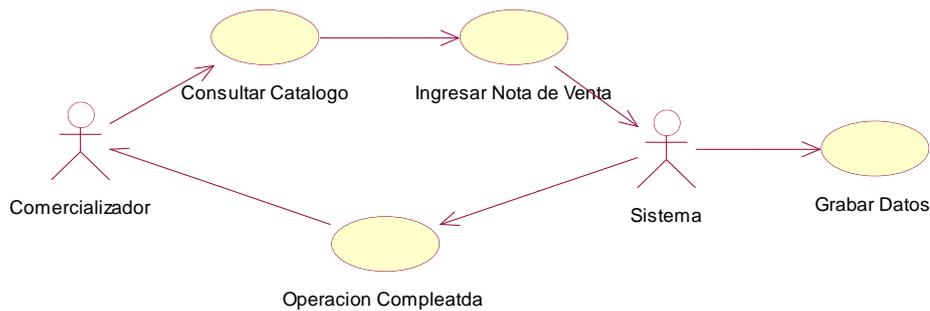


FIGURA 4.10.3.5 CASO DE USO INGRESAR NOTA DE VENTA

Nombre caso de Uso: Ingresar Nota de Venta Factura

Propósito: Ingresar nota de Venta.

Visión General: El Comercializador solicita consultar catalogo con esta operación se puede visualizar los productos existentes en inventario, selecciona los productos que serán vendidos , con los cuales se ingresa la nota de venta y se termina la operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el comercializador consultar catalogo.
2. El sistema presenta un formulario de todas los productos existentes en inventario
3. El comercializador selecciona los productos que serán vendidos

4. El sistema almacena los datos que selecciono en catalogo en la nota de venta

Cursos Alternativos

- Curso típico 2: No existe formulario de consultar catalogo termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso.

4.10.3.6 MODIFICAR CATALOGO

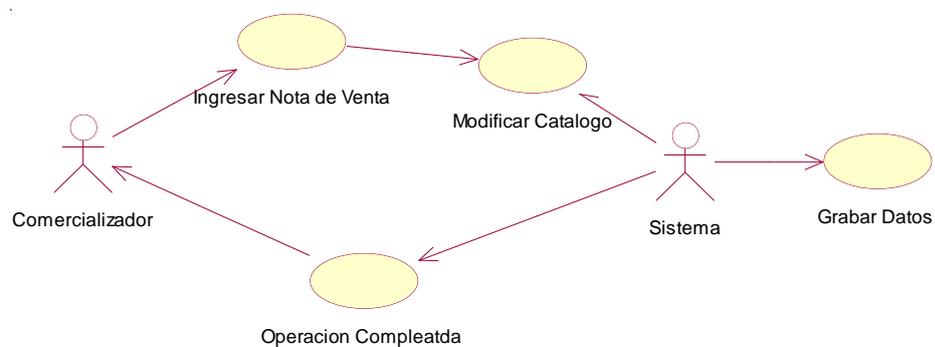


FIGURA 4.10.3.6 CASO DE USO MODIFICAR CATALOGO

Nombre caso de Uso: Modificar Catalogo

Propósito: Modificar Catalogo.

Visión General: El Comercializador solicita la operación Ingresar Nota de venta con el fin de poder Modificar Catalogo, el sistema presenta formulario de Disponible de Catalogo, el comercializador selecciona los productos a vender el sistema modifica la información dependiendo de los productos seleccionados y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Comercializador solicita la operación Ingresar Nota de venta.
2. El sistema presenta formulario disponibilidad
3. El sistema modifica catalogo dependiendo de los productos que se han seleccionado por el comercializador y confirma operación.

Cursos Alternativos

- Curso típico 2: No existe formulario, termina el caso de uso.
- Curso típico 4: Los datos no están bien ingresados, termina caso de uso

4.10.3.7 CONSULTAR CATALOGO

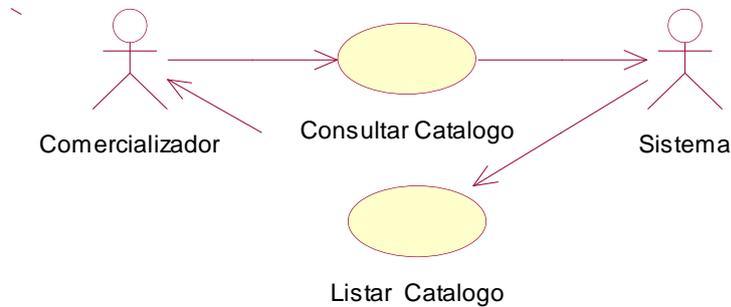


FIGURA 4.10.3.7 CASO DE USO CONSULTAR CATALOGO

Nombre caso de Uso: Consultar Catalogo.

Propósito: Observar cuales son los disponibles de los productos registrados en el sistema.

Visión General: El Comercializador solicita la operación Consultar Catalogo con el fin de vender, el sistema presenta la lista de productos disponibles en el sistema y confirma operación.

Tipo: Primario real

Referencias:

Curso Típico de eventos:

1. El caso de uso comienza cuando el Comercializador solicita la operación Consultar Catalogo.

2. El sistema presenta el listado de productos disponibles en el sistema.

3. El sistema confirma operación.

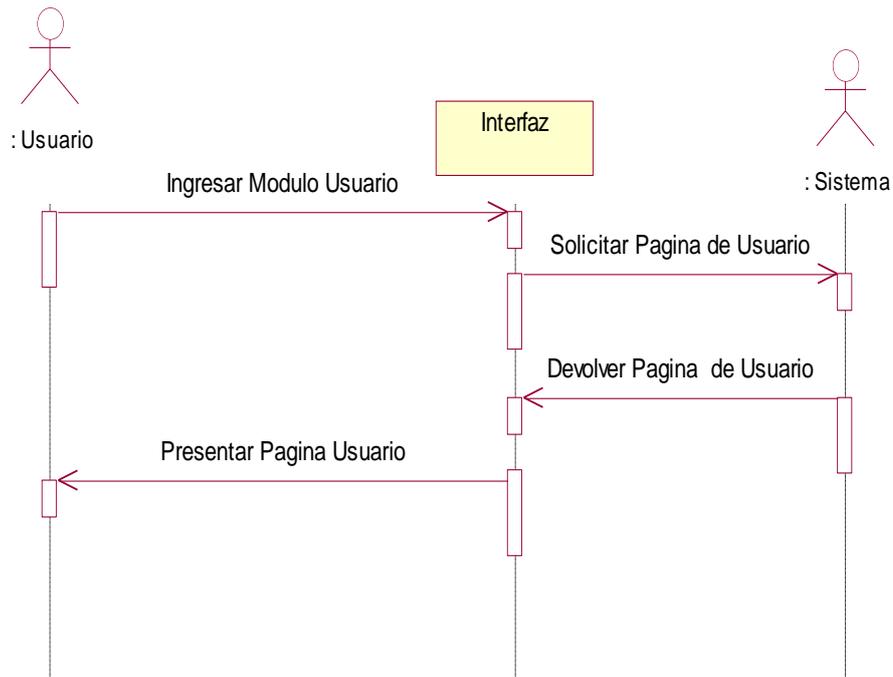
Cursos Alternativos

- Curso típico 2: No existe productos disponibles registrados, termina el caso de uso.

4.11 FASE DE DISEÑO

4.11.1 Diagramas de Secuencia

4.11.1.1 Ingresar Modulo Usuario



Nombre: Ingresar Modulo Usuario

Responsabilidades: Deberá validar los Datos de Usuario y tipo de los datos al momento de ingresarlos

Tipo: PSGI

Caso de uso: Ingresar Modulo Administrador

Notas:

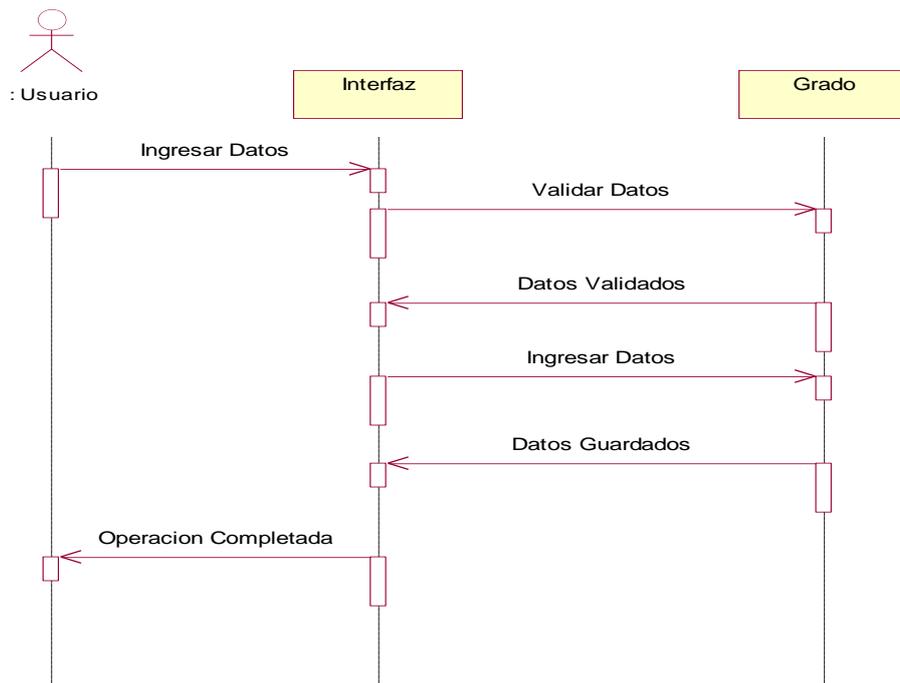
Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

4.11.1.2 Crear Grado



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Crear Grado

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Grado (Datos)

Nombre: Guardar Grado(Datos)

Responsabilidades: Deberá guardar los datos de grado en el Irepositorio

Tipo: PSGI

Caso de uso: Crear Grado

Notas:

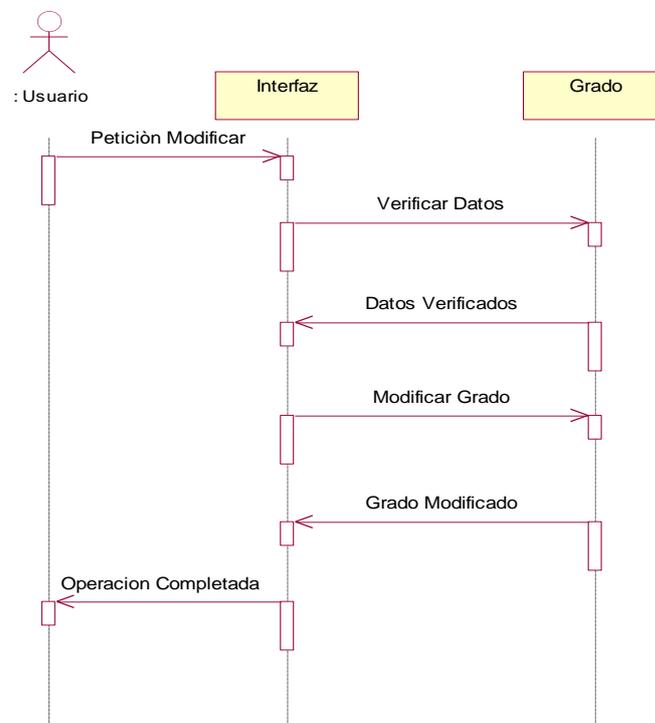
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Grado guardado

4.11.1.3 Modificar Grado



Verificar Grado

Nombre: Verificar Grado

Responsabilidades: Deberá verificar si los datos del Grado a modificar tienen o no otros datos que dependan de ellos.

Tipo: PSIG

Caso de uso: Modificar Grado

Notas:

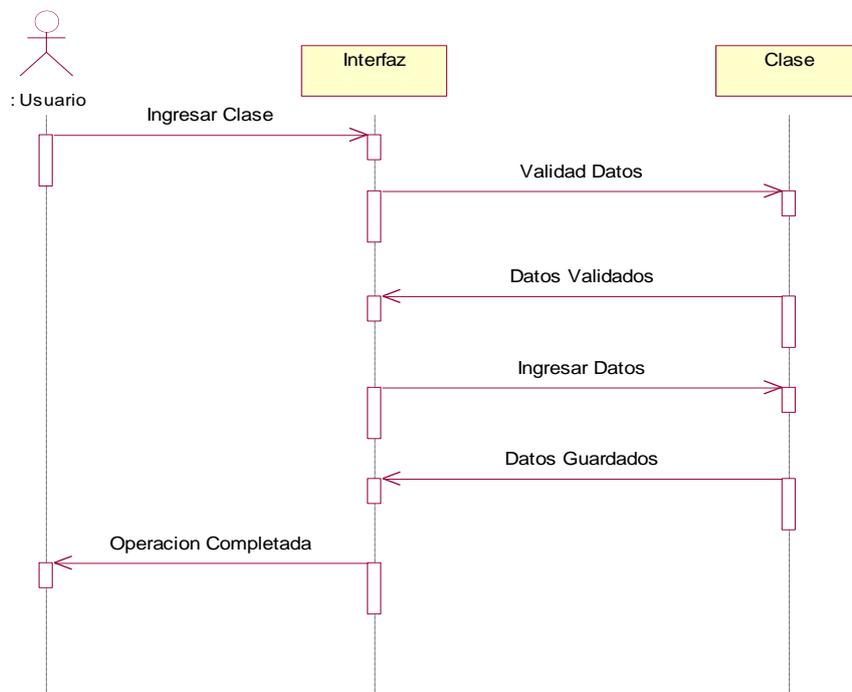
Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Grado verificado

4.11.1.4 Crear Clase



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Crear Clase

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Clase (Datos)

Nombre: Guardar Clase (Datos)

Responsabilidades: Deberá guardar los datos de Clase en el
Irepositorio

Tipo: PSGI

Caso de uso: Crear Clase

Notas:

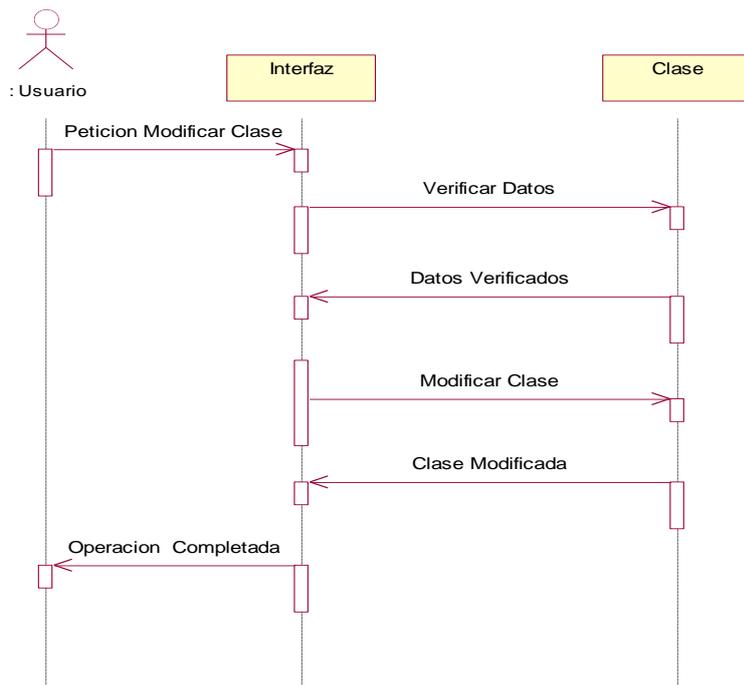
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Clase guardada

4.11.1.5 Modificar Clase



Verificar Clase

Nombre: Verificar Clase

Responsabilidades: Deberá verificar si los datos del Clase a modificar tienen o no otros datos que dependan de ellos.

Tipo: PSIG

Caso de uso: Modificar Clase

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Clase verificado

Modificar Clase

Nombre: Modificar Clase

Responsabilidades: Deberá modificar el Clase solicitado del repositorio

Tipo: PSIG

Caso de uso: Modificar Clase

Notas:

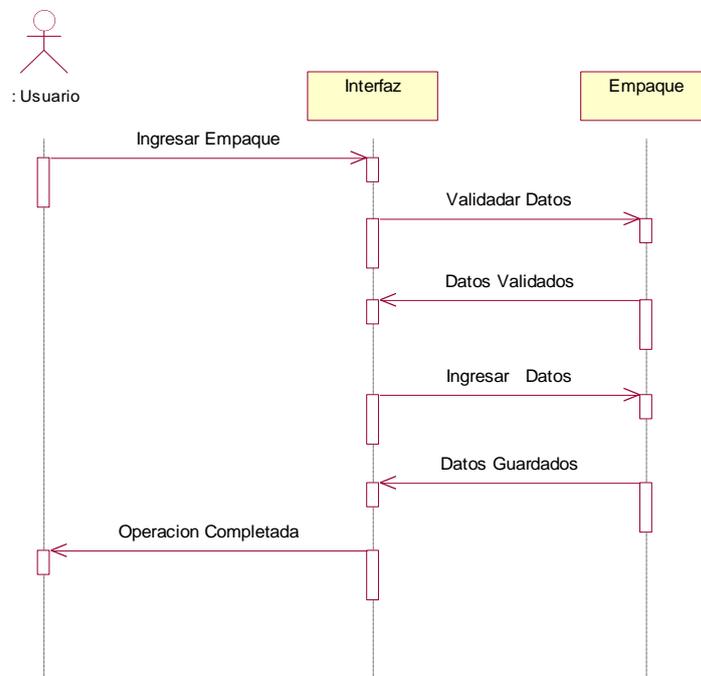
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Clase modificado

4.11.1.6 Crear Empaque



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Crear Empaque

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Empaque (Datos)

Nombre: Guardar Empaque (Datos)

Responsabilidades: Deberá guardar los datos de Empaque en el Irepositorio

Tipo: PSGI

Caso de uso: Crear Empaque

Notas:

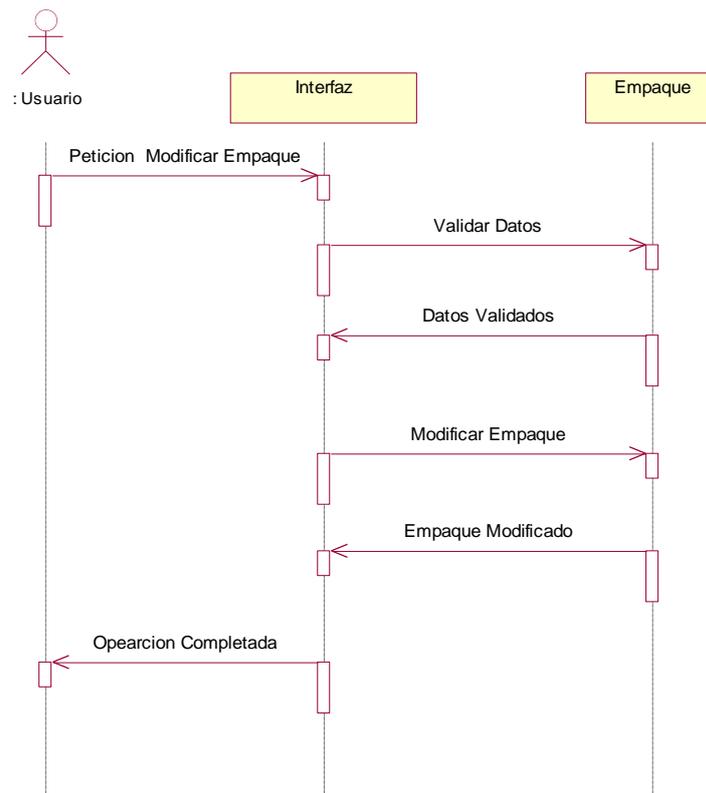
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Empaque guardada

4.11.1.7 Modificar Empaque



Verificar Empaque

Nombre: Verificar Empaque

Responsabilidades: Deberá verificar si los datos del Empaque a modificar tienen o no otros datos que dependan de ellos.

Tipo: PSIG

Caso de uso: Modificar Empaque

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Empaque verificado

Modificar Empaque

Nombre: Modificar Empaque

Responsabilidades: Deberá modificar el Empaque solicitado del repositorio

Tipo: PSIG

Caso de uso: Modificar Empaque

Notas:

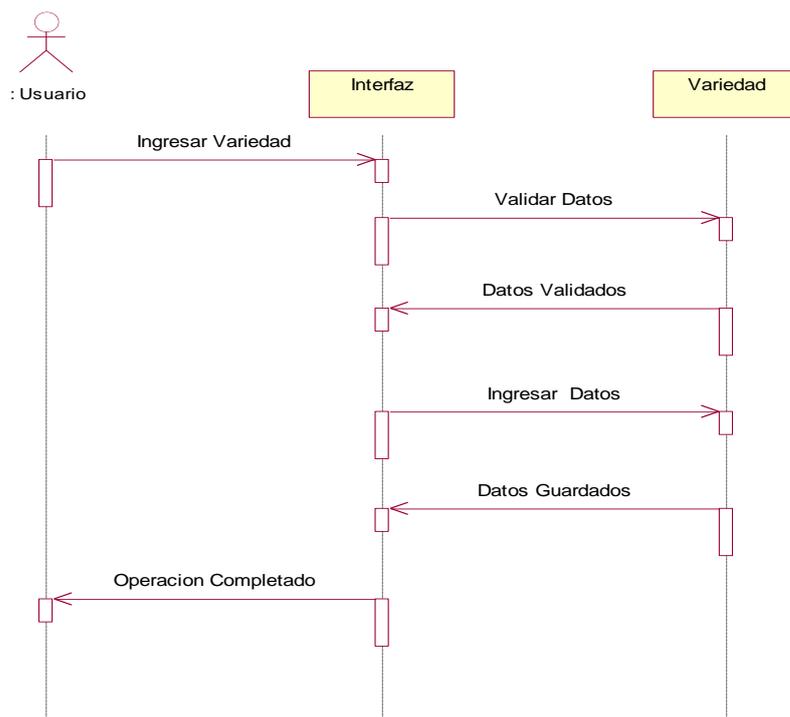
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Empaquemodificado

4.11.1.8 Crear Variedad



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Crear Variedad

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Variedad (Datos)

Nombre: Guardar Variedad (Datos)

Responsabilidades: Deberá guardar los datos de Variedad en el Irepositorio

Tipo: PSGI

Caso de uso: Crear Variedad

Notas:

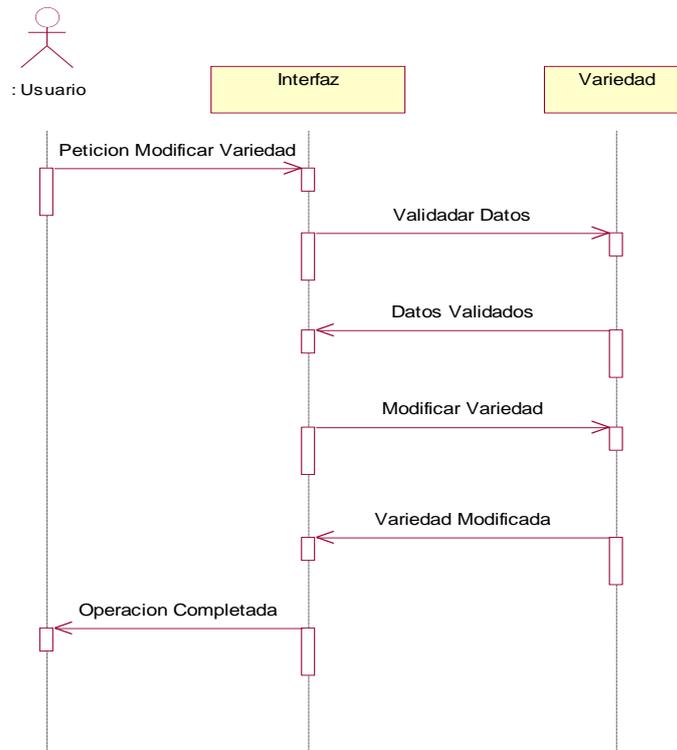
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Variedad guardada

4.11.1.9 Modificar Variedad



Verificar Variedad

Nombre: Verificar Variedad

Responsabilidades: Deberá verificar si los datos del Variedad a modificar tienen o no otros datos que dependan de ellos.

Tipo: PSIG

Caso de uso: Modificar Variedad

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Variedad verificado

Modificar Variedad

Nombre: Modificar Variedad

Responsabilidades: Deberá modificar el Variedad solicitado del repositorio

Tipo: PSIG

Caso de uso: Modificar Variedad

Notas:

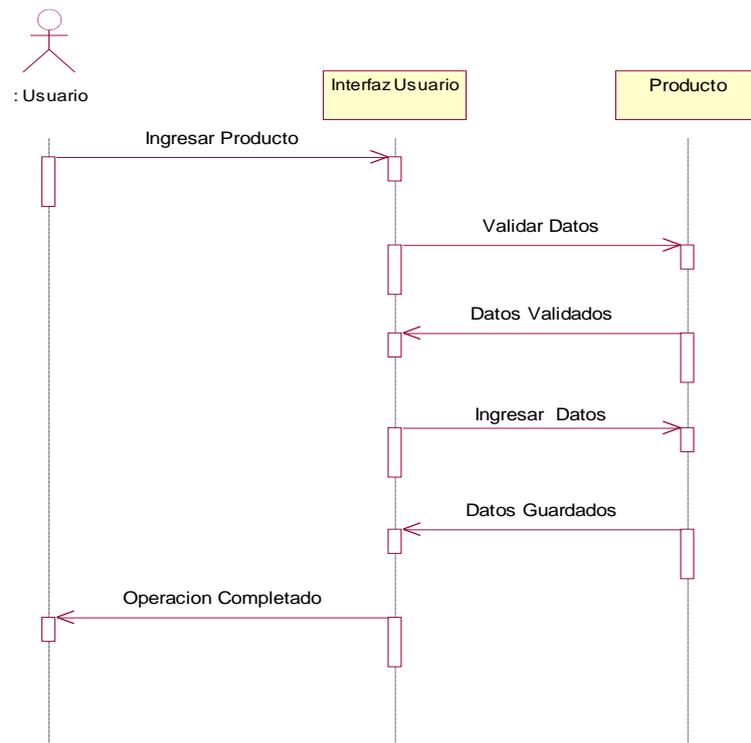
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Variedad modificado

4.11.1.10 Crear Producto



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Crear Producto

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Producto (Datos)

Nombre: Guardar Producto (Datos)

Responsabilidades: Deberá guardar los datos de Producto en el Irepositorio

Tipo: PSGI

Caso de uso: Crear Producto

Notas:

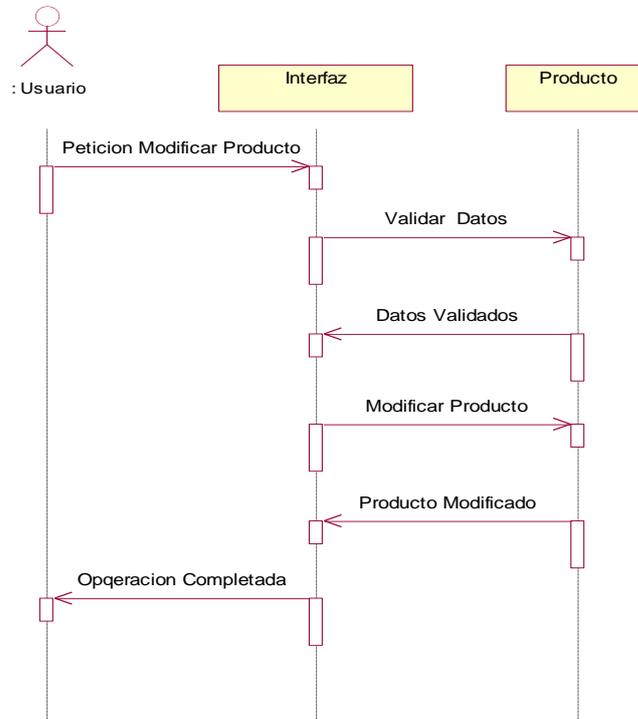
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Variedad guardada

4.11.1.11 Modificar Producto



Verificar Producto

Nombre: Verificar Producto

Responsabilidades: Deberá verificar si los datos del Producto a modificar tienen o no otros datos que dependan de ellos.

Tipo: PSIG

Caso de uso: Modificar Producto

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Producto verificado

Modificar Producto

Nombre: Modificar Producto

Responsabilidades: Deberá modificar el Producto solicitado del repositorio

Tipo: PSIG

Caso de uso: Modificar Producto

Notas:

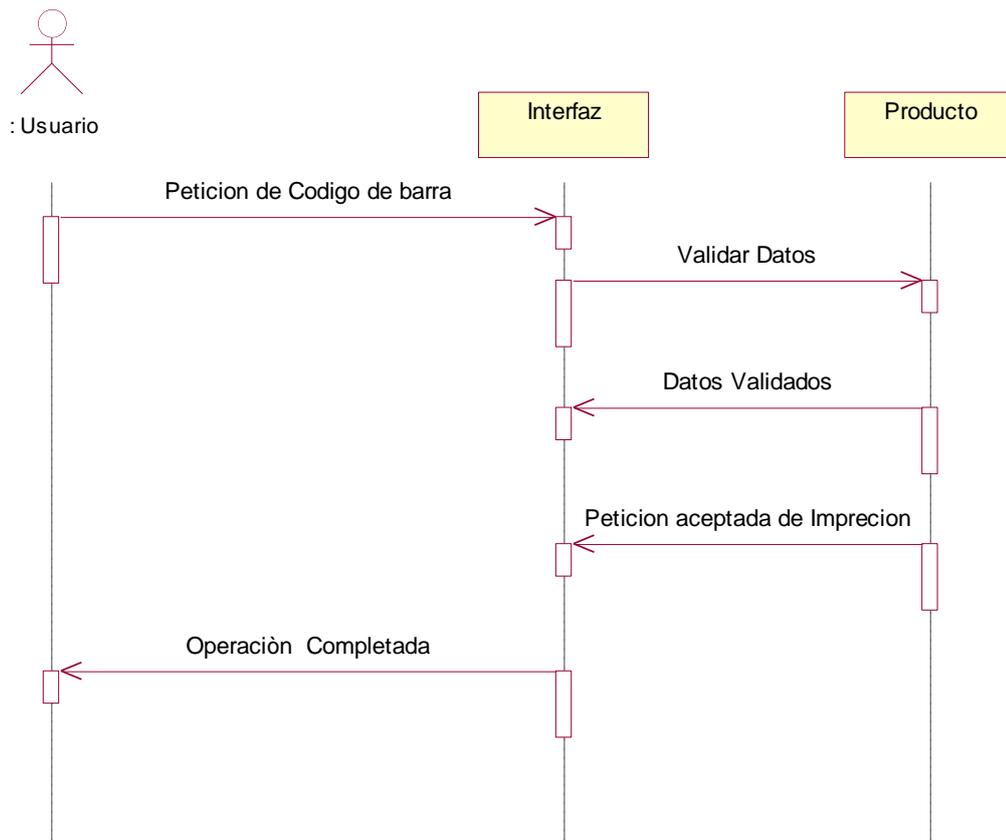
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Producto modificado

4.11.1.12 ImprimirCodigo de Barra



Imprimir código de Barra

Nombre: Verificar Datos Barra

Responsabilidades: Deberá Imprimir el código de barra seleccionado por el usuario

Tipo: PSIG

Caso de uso: Imprimir código de Barra

Notas:

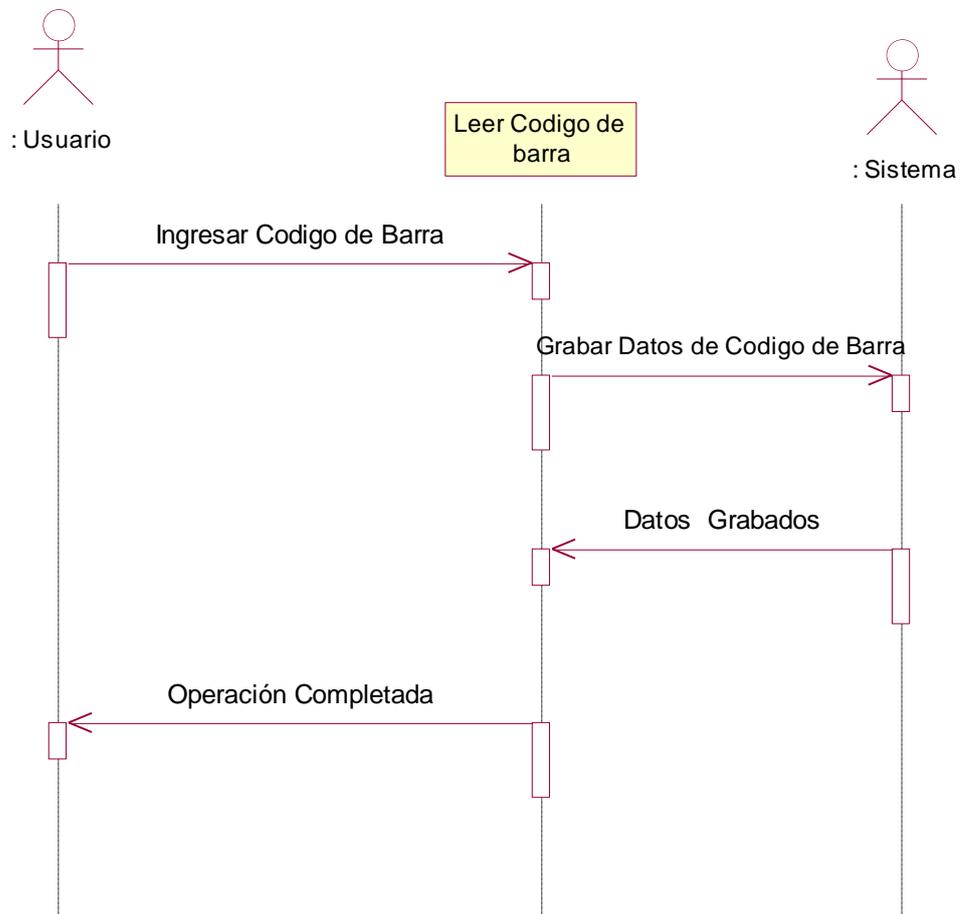
Excepciones:

Salidas: Etiqueta

Precondiciones: Almacenar producto Petición modificar

Postcondiciones: Impresión de la etiqueta del Producto

4.11.1.13 Leer Código de Barra



Leer Código de Barra

Nombre: Leer código de Barra

Responsabilidades: Deberá Leer el código de barra ingresado por el usuario con la ayuda del lector)

Tipo: PSIG

Caso de uso: Leer código de Barra

Notas:

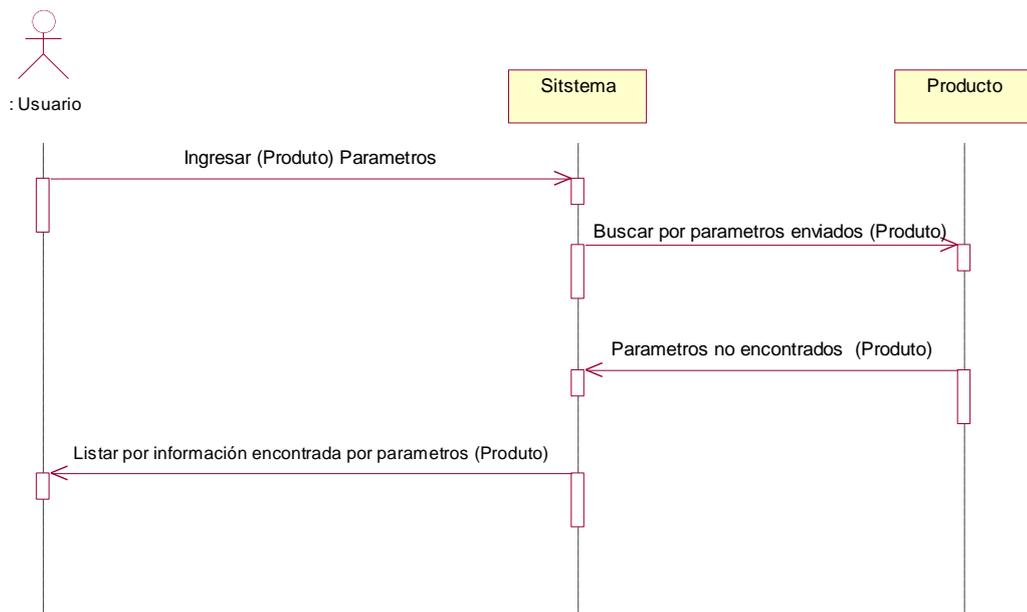
Excepciones:

Salidas:

Precondiciones: Tener etiqueta del producto almacenar

Postcondiciones:

4.11.1.14 Buscar Producto



Buscar Producto

Nombre: Buscar Producto

Responsabilidades: Deberá buscar la información dependiendo al parámetro ingresado de producto.

Tipo: PSGI

Caso de uso: Buscar Producto

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos de búsqueda

Postcondiciones: Producto encontrado o no encontrado

Presentar Producto encontrado

Nombre: Presentar producto encontrado

Responsabilidades: Deberá presentar en pantalla de producto

Tipo: PSGI

Caso de uso: Buscar Producto

Notas:

Excepciones:

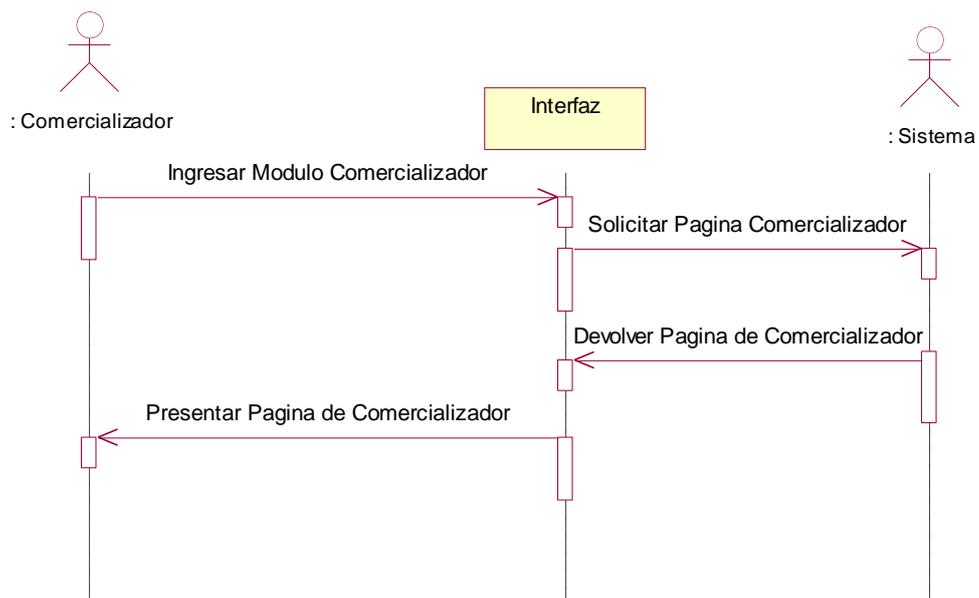
Salidas:

Precondiciones: Busca Producto

Postcondiciones:

4.11.2 Diagramas de Secuencia Comercialización

4.11.2.1 Ingresar Modulo Comercialización



Nombre: Ingresar Modulo Comercialización

Responsabilidades: Deberá validar los Datos de Comercialización y tipo de los datos al momento de ingresarlos

Tipo: PSGI

Caso de uso: Ingresar Modulo Comercialización

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Nombre: Presentar Pantalla

Responsabilidades: Deberá presentar la pantalla de comercializador
Tipo: PSGI

Caso de uso: Ingresar Modulo Comercialización

Notas:

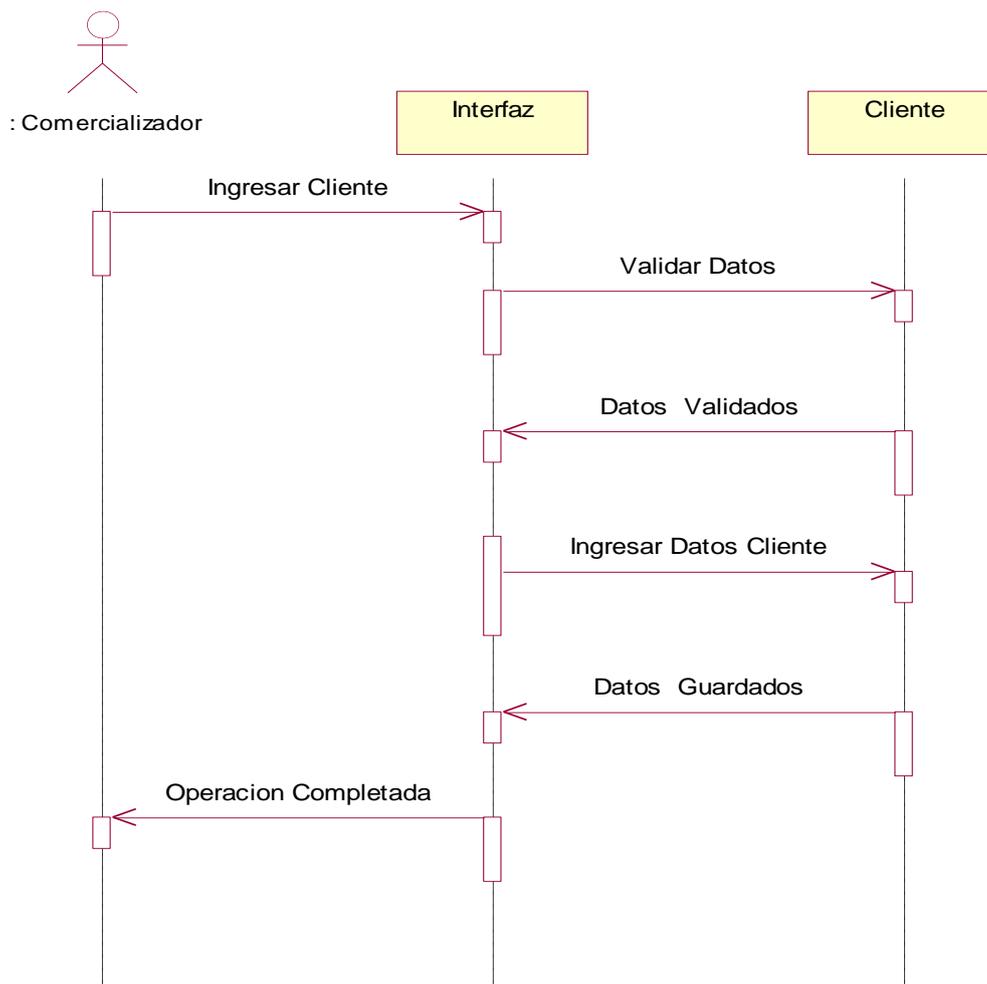
Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Presentar Pantalla

4.11.2.3 Ingresar Cliente



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Ingresar Cliente

Notas:

Excepciones:

Salidas:

Precondiciones:

Postcondiciones: Datos validados

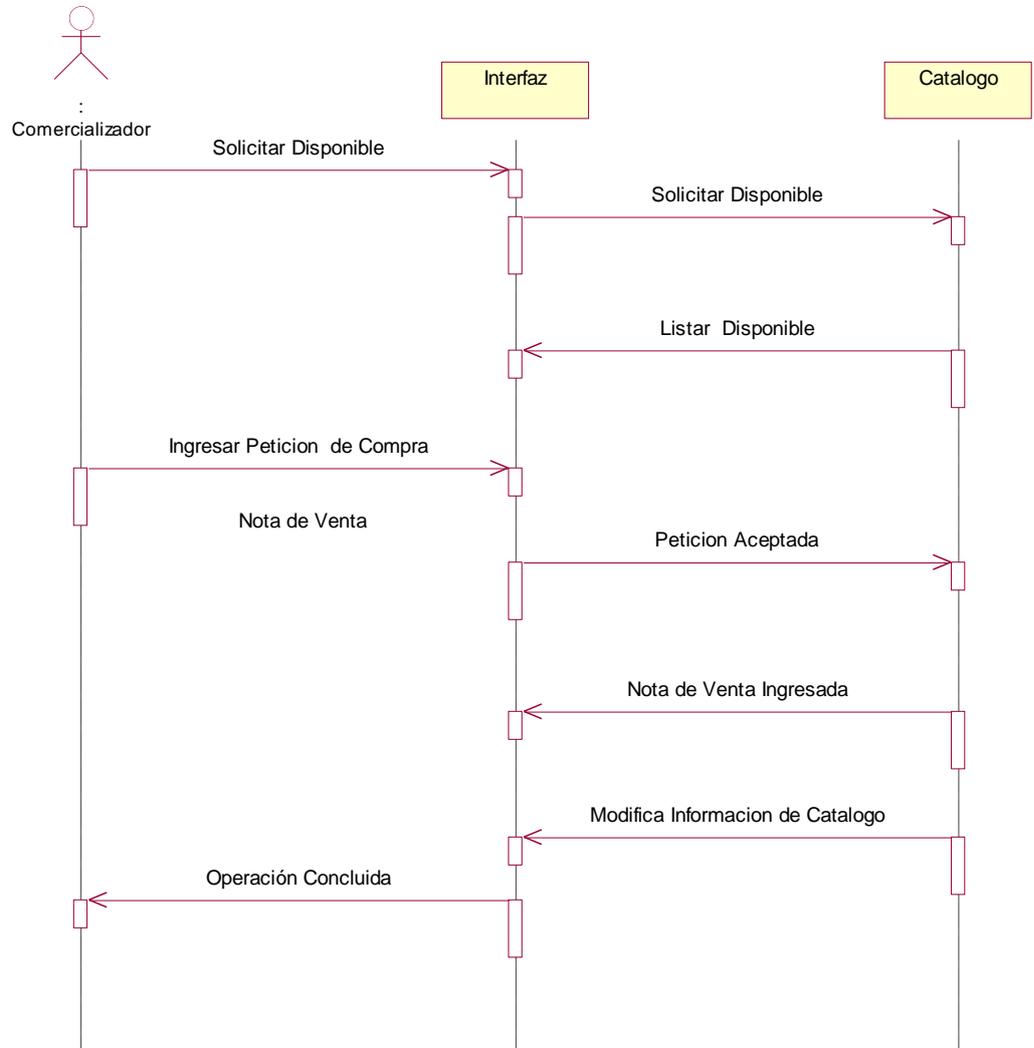
Guardar Cliente (Datos)

Nombre: Guardar Cliente (Datos)

Responsabilidades: Deberá guardar los datos de Cliente en el
Irepositorio

Tipo: PSGI

4.11.2.4 Ingresar Nota de venta



Solicitar Disponible

Responsabilidades: Deberá desplegar toda la información de todo el disponible

Tipo: PSIG

Caso de uso: Ingresar Nota de Venta

Notas:

Excepciones:

Salidas:

Precondiciones: Existir datos en Catologo

Postcondiciones: Nota de Venta Registrada, Catologo Modificado

Ingresar Petición de Nota de Venta

Responsabilidades: Deberá ingresar la petición de nota de venta

Tipo: PSIG

Caso de uso: Ingresar Nota de Venta

Notas:

Excepciones:

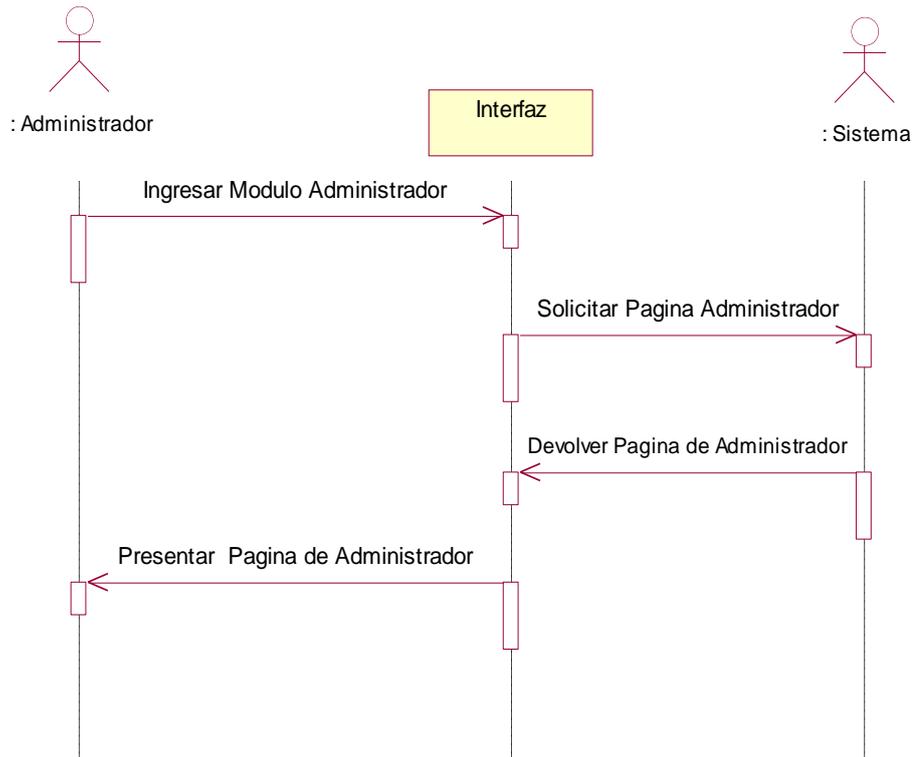
Salidas:

Precondiciones: Existir datos en Catologo

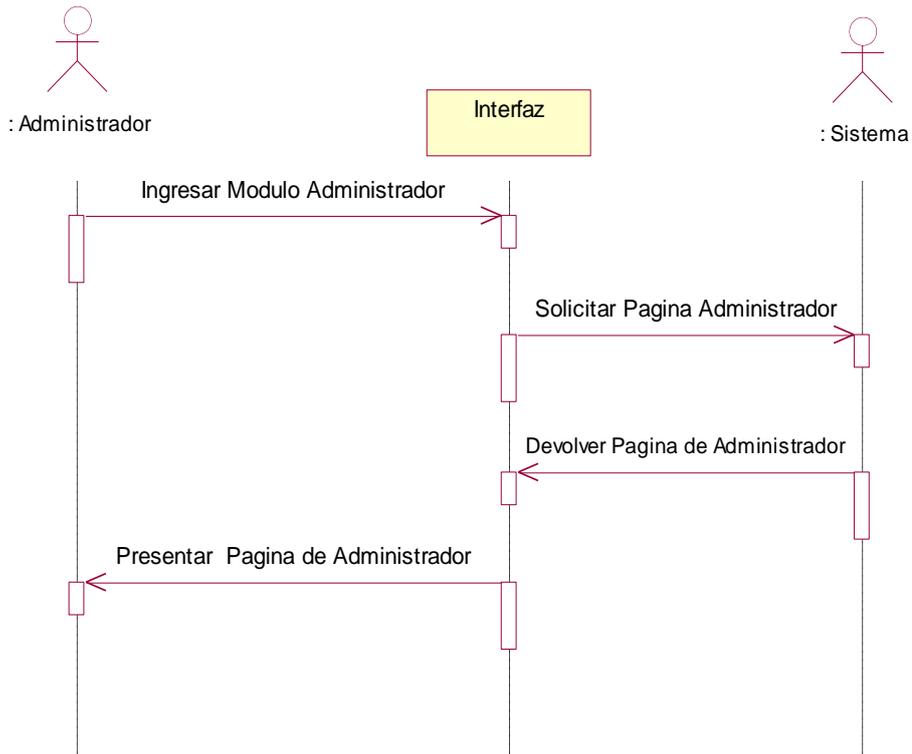
Postcondiciones: Nota de Venta Registrada.

4. 11.3 DIAGRAMAS DE SECUENCIA DE ACTOR ADMINISTRADOR

4.11.3.1 Ingresar Modulo Administrador



4.11.3.1 Ingresar Modulo Administrador



Nombre: Ingresar Modulo Administrador

Responsabilidades: Deberá validar los Datos de Administrador y tipo de los datos al momento de ingresarlos

Tipo: PSGI

Caso de uso: Ingresar Modulo Administrador

Notas:

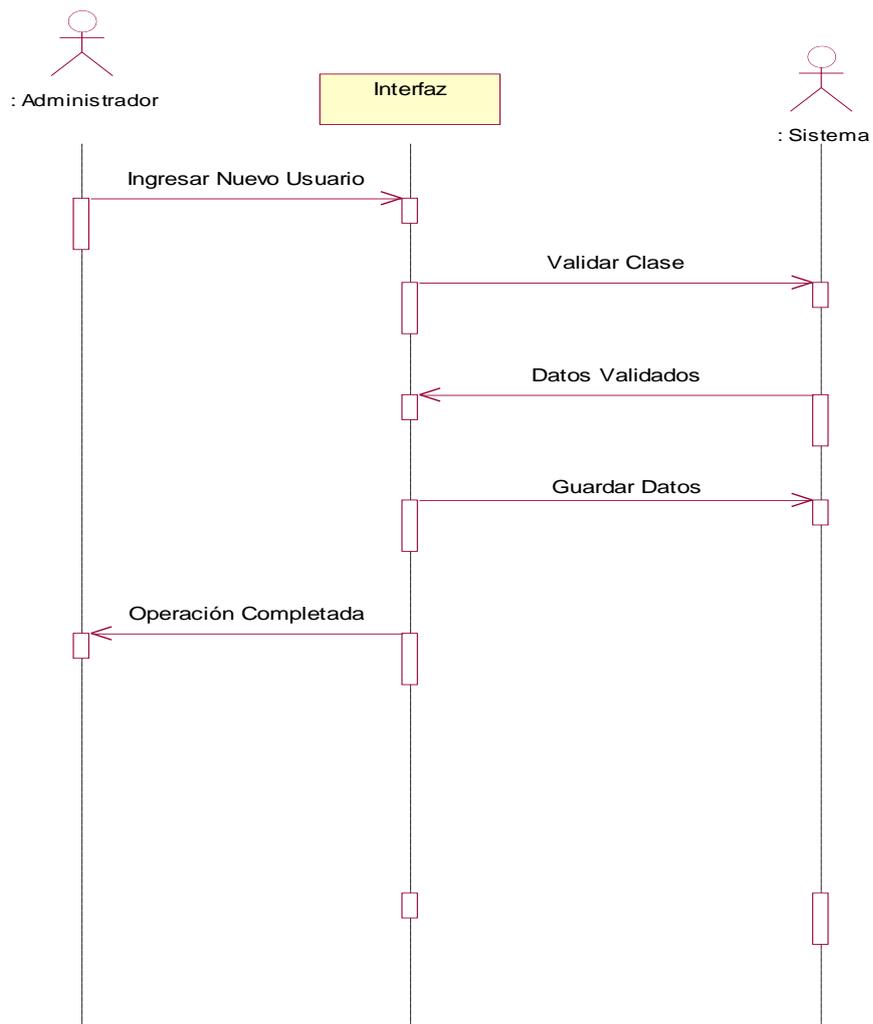
Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

4.11.3.2 Ingresar Usuario



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos

Tipo:PSGI

Caso de uso: Ingresar Usuario

Notas:

Excepciones:

Salidas:

Precondiciones:

Postcondiciones: Datos validados

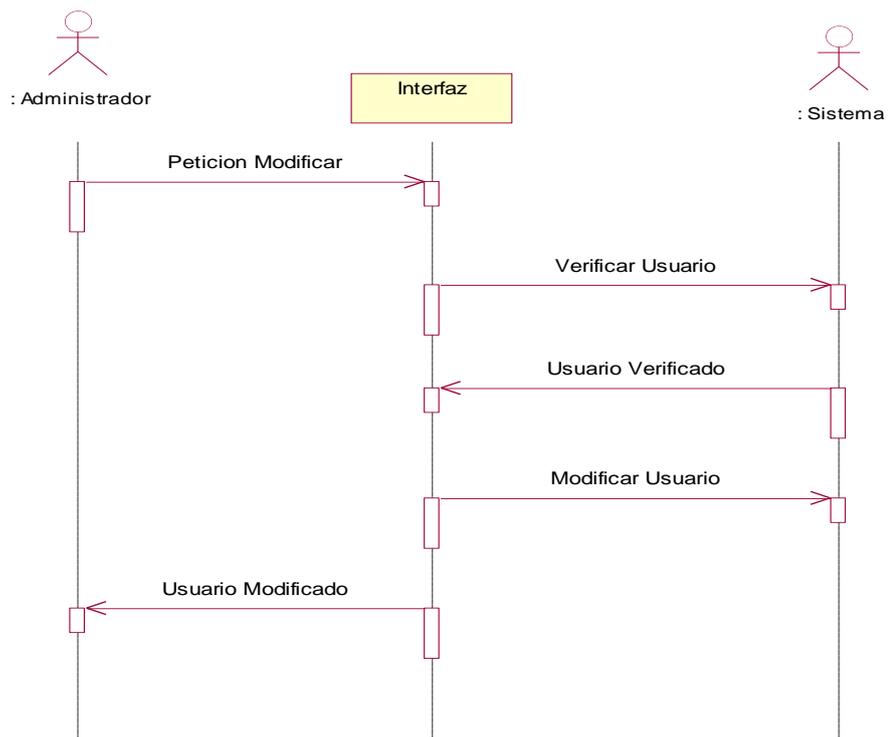
Guardar Usuario (Datos)

Nombre: Guardar Usuario (Datos)

Responsabilidades: Deberá guardar los datos de Usuario en el
Irepositorio

Tipo: PSGI

4.11.3.3 Modificar Usuario



Modificar Usuario

Nombre: Modificar Usuario

Responsabilidades: Deberá modificar el Usuario solicitado del repositorio

Tipo: PSIG

Caso de uso: Modificar Usuario

Notas:

Excepciones:

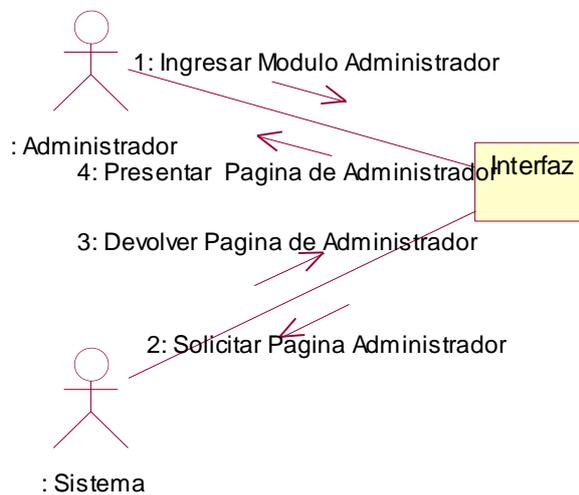
Salidas:

Precondiciones: Verificar datos

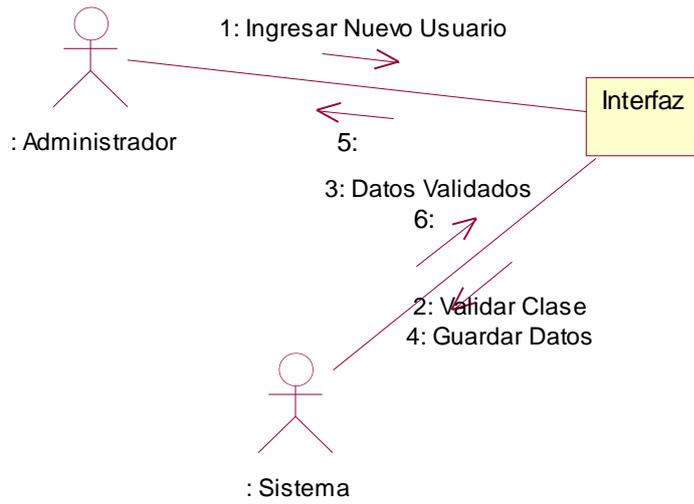
Postcondiciones: Usuario Modificado

4.12 DIAGRAMAS DE COLABORACIÓN

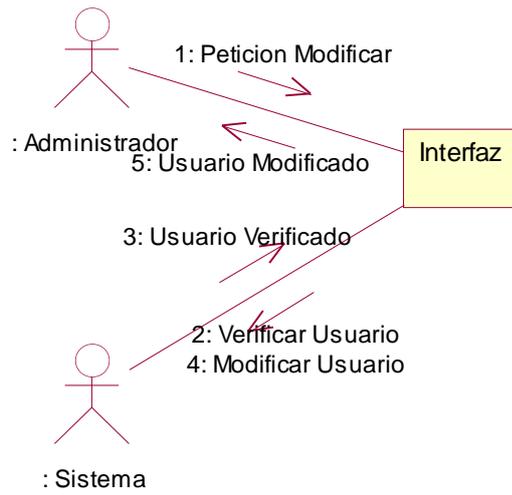
4.12.1 Ingresar Modulo Administrador



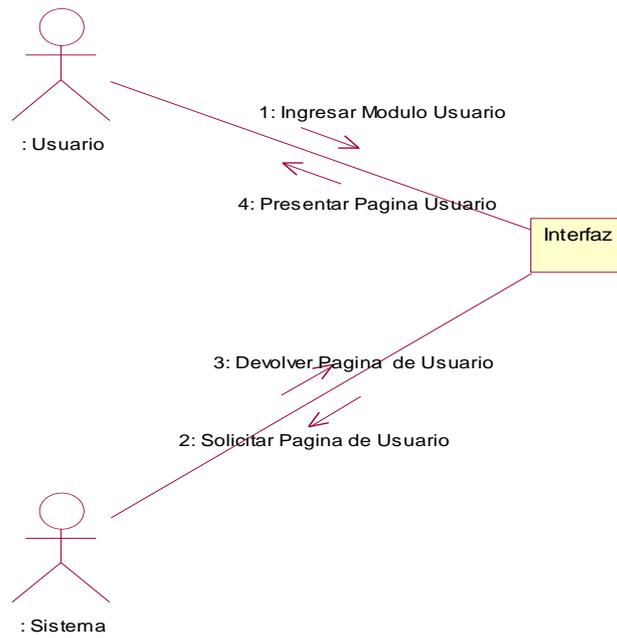
4.12.2 Ingresar Usuarios



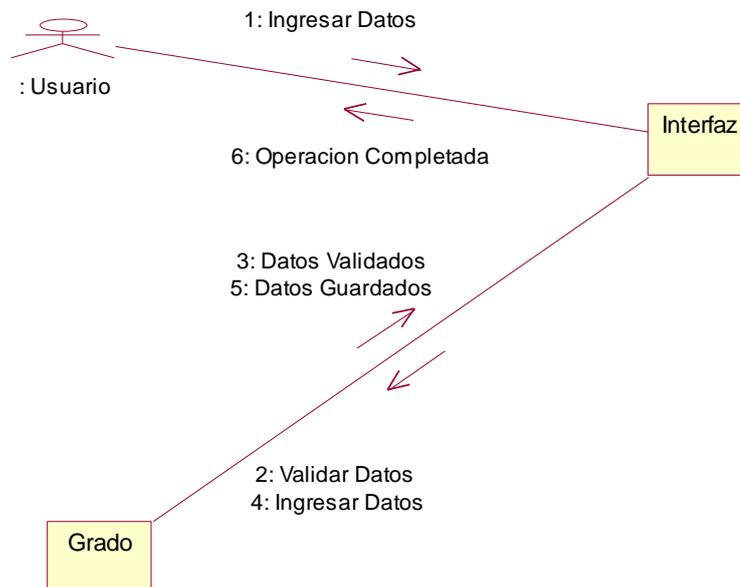
4.12.3 Modificar Usuario



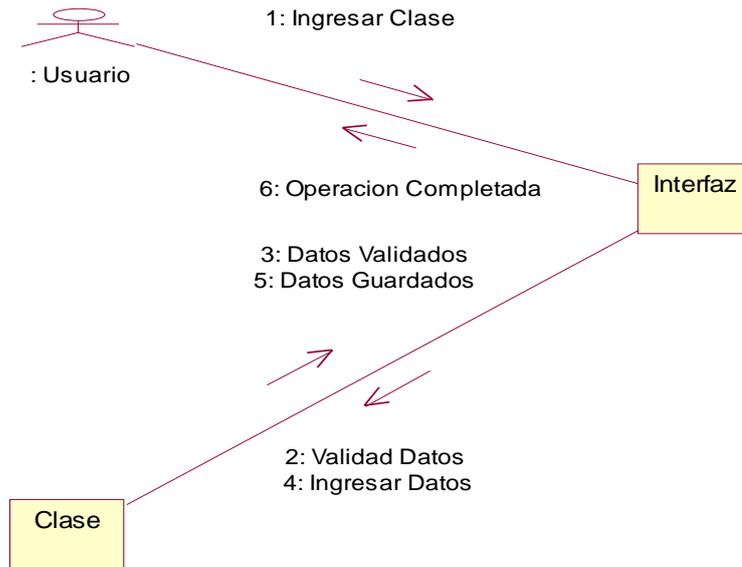
4.12.5 Ingresar Modulo Usuario



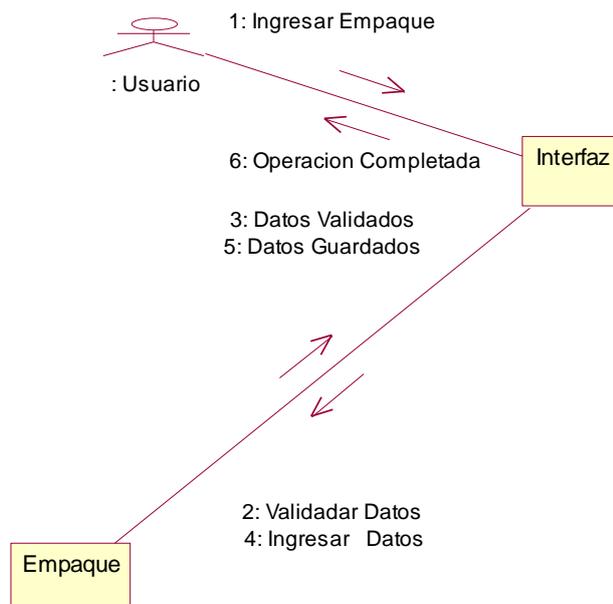
4.12.6 Ingresar Grado



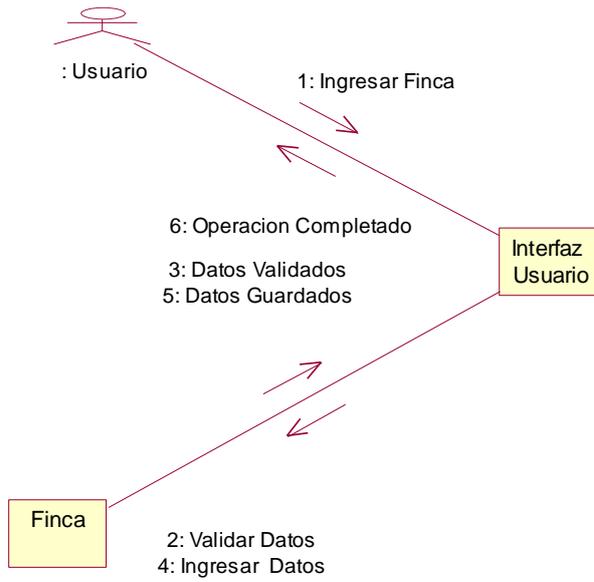
4.12.7 Ingresar Clase



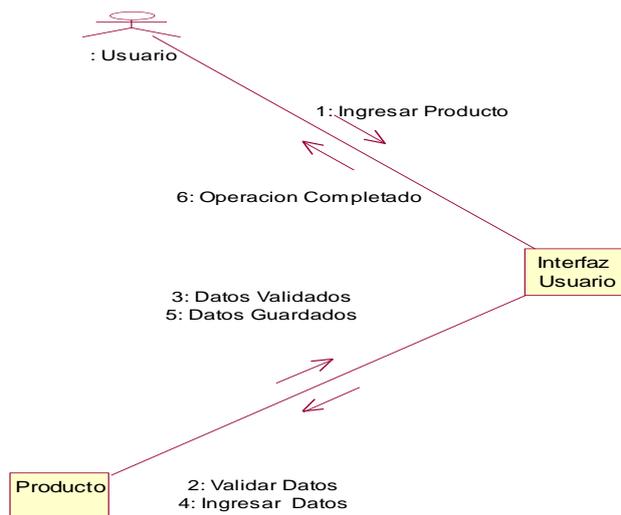
4.12.8 Ingresar Empaque



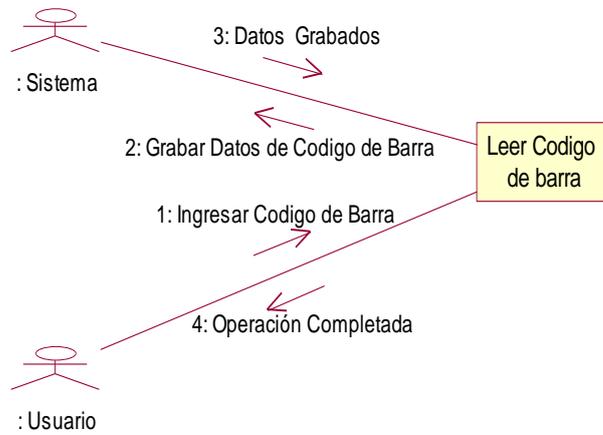
4.12.9 Ingresar Finca



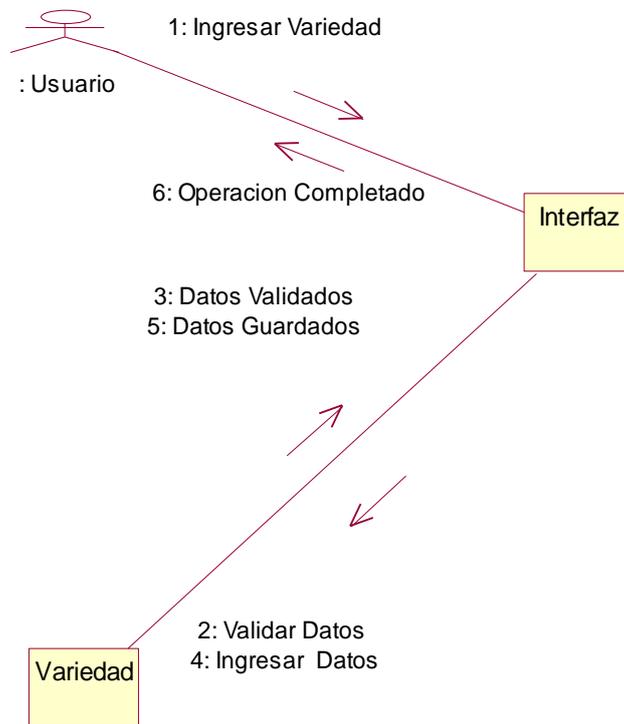
4.12.10 Ingresar Producto



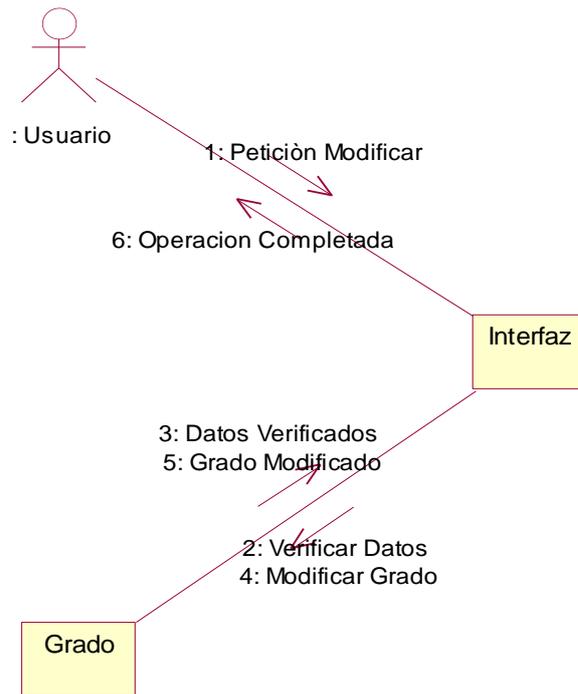
4.12.11 LeerCodigo de Barra



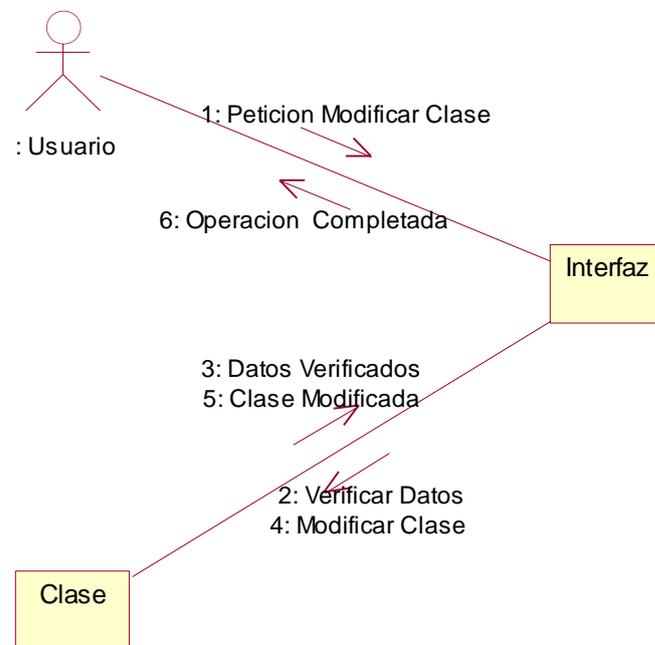
4.12.12 Ingresar Variedad



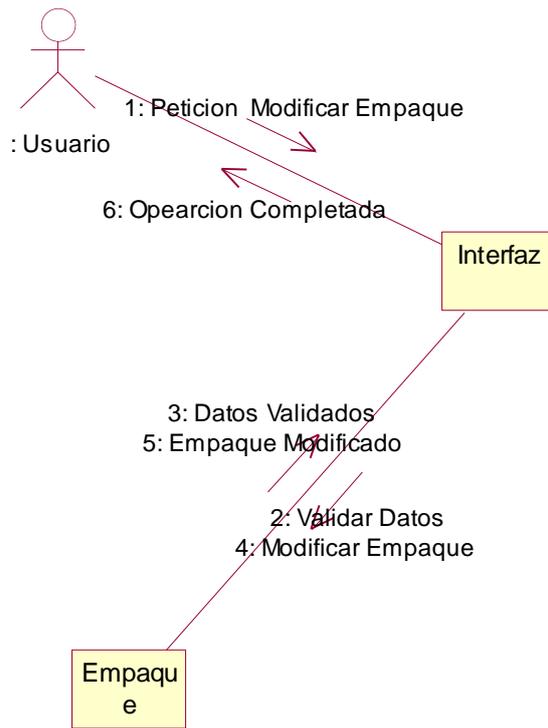
4.12.13 Modificar Grado



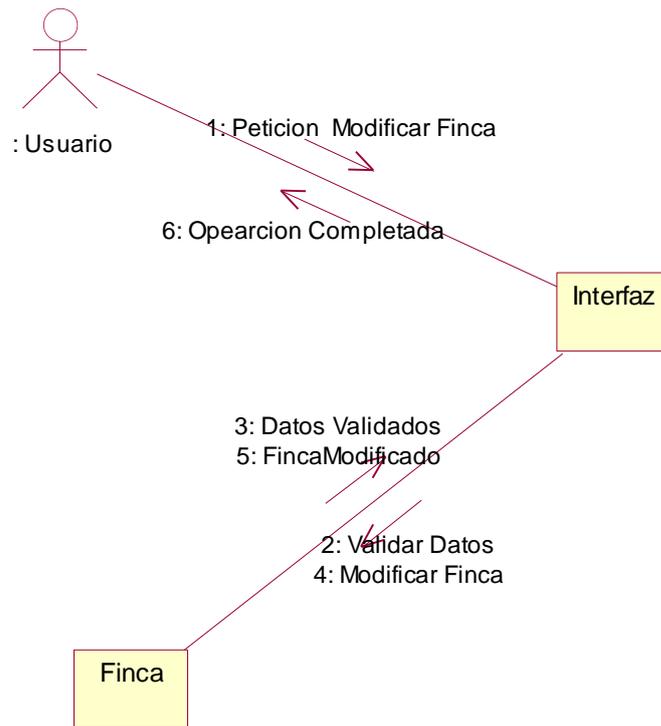
4.12.14 Modificar Clase



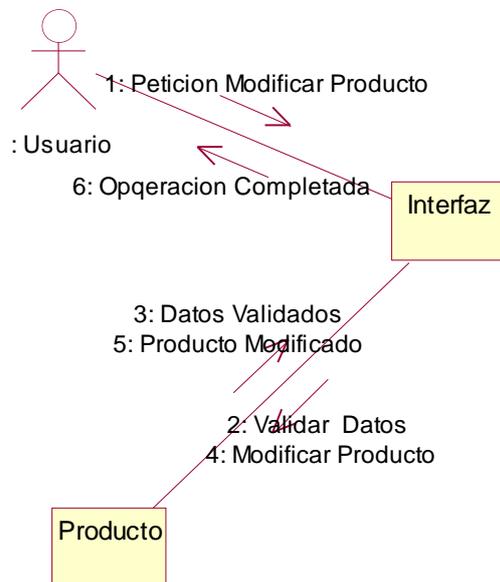
4.12.15 Modificar Empaque



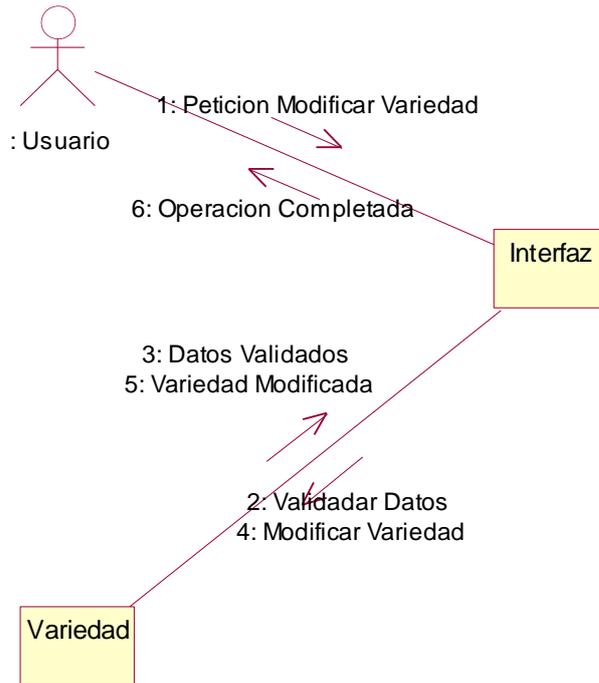
4.12.16 Modificar Finca



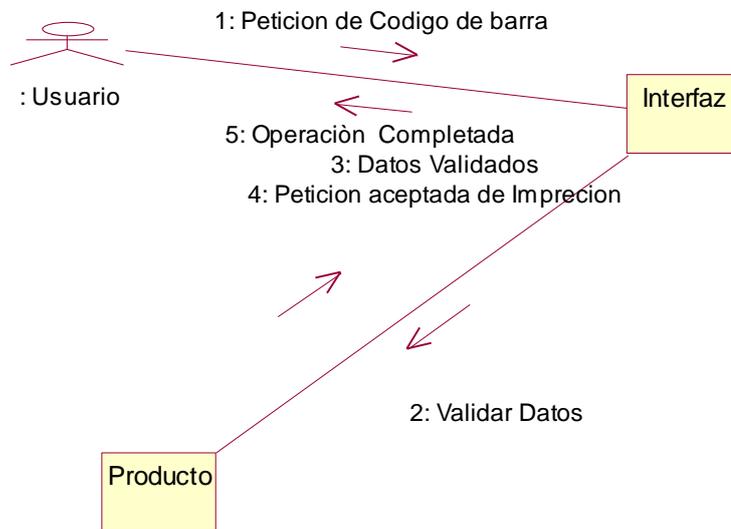
4.12.17 Modificar Producto



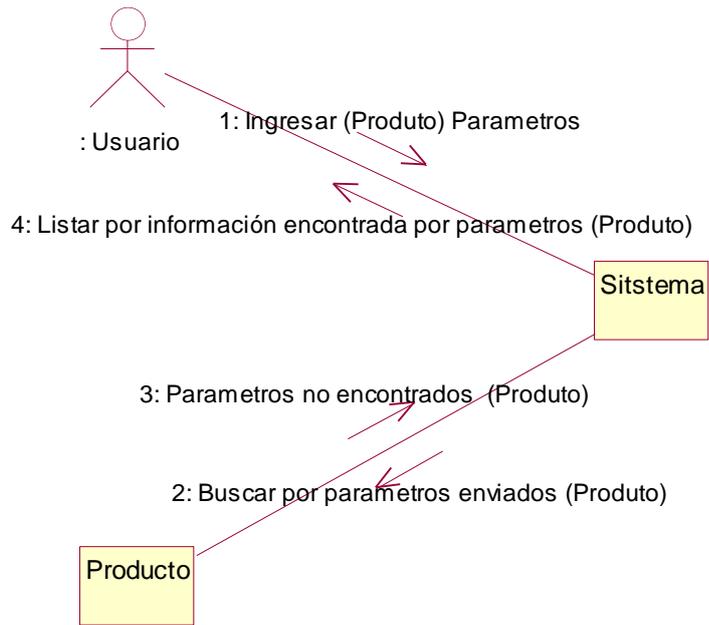
4.12.18 Modificar Variedad



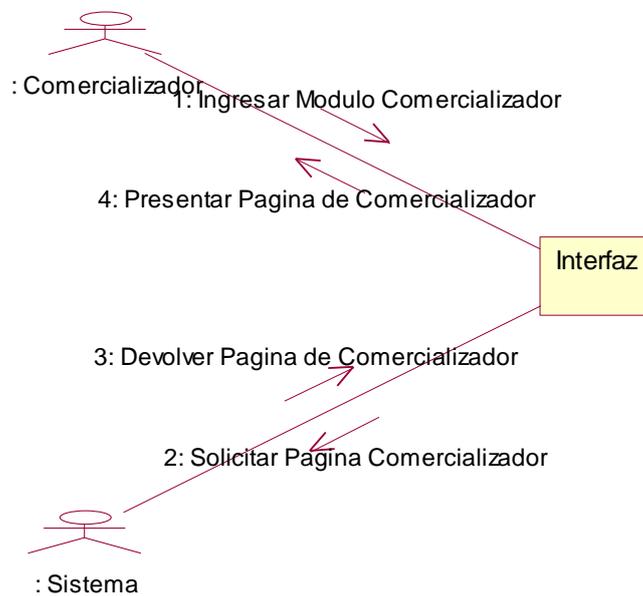
4.12.19 Imprimir Código de Barra



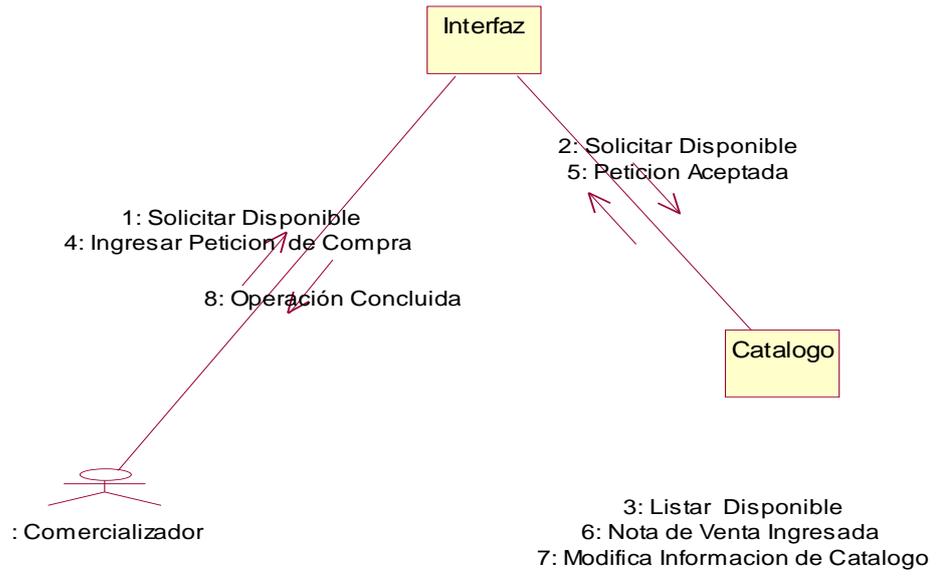
4.12.20 Buscar Producto



4.12.1 Ingresar a Modulo Comercializador



4.12.2 Ingresar Nota de Venta



CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Al finalizar el presente trabajo de investigación podemos concluir lo siguiente:

- Para crear aplicaciones Web de calidad debemos familiarizarnos con las tecnologías que son aplicables no solamente para el mercado de hoy en día sino también para el de mañana, a saber JSP
- La estructura de las páginas JSP hace más fácil el diseño de páginas a manos de los diseñadores Web.
- Java es un lenguaje muy potente, pero un poco más complicado de usar porque es orientado a objetos y la manera de escribir los programas es más rígida
- Los JSP es multiplataforma y abierta y puede ser fácilmente actualizados o cambiados sin que afecte a las aplicaciones basadas en la tecnología JSP.
- JSP hace mayor énfasis en los componentes que en los Scripts, esto hace que sea más fácil revisar el contenido sin que afecte a la lógica o revisar la lógica sin cambiar el contenido.
- Los JavaBeans nos permite re-utilizar componentes de software que pueden ser manipulados en herramientas de desarrollo y son sencillos.
- El uso de software libre utilizado en esta investigación como Tomcat, Msql-Front, JCreator, fue importante a nivel legal y económico, porque los costos en cuanto a software fueron de menos costos que si hubiésemos utilizado software propietario .

5.2 Recomendaciones

Al finalizar el presente trabajo de investigación recomendamos lo siguiente:

- Utilizar páginas jsp en lo posible porque son más entendibles y permiten la modificación rápida.

- Utilizar Javabeans incrustado en páginas jsp, con el fin de no reutilizar un código ya existente y no ser tan compleja la programación.

- Realizar pruebas en cada una de las fases de Desarrollo del Proyecto con el fin de comprobar el buen funcionamiento de la aplicación.

- Utilizar software libre, ya que a más de no necesitar licencia, son robustos, estables, y además no obligan a usar una plataforma en particular, como ocurre con otros tipos de software.

- Utilizar una metodología orientada a objetos al desarrollar un software ya que representa un ahorro de tiempo, permitiendo tener un enfoque claro del software desde el principio

REFERENCIAS BIBLIOGRÁFICAS

- **HAHN, Harley.** “Internet Manual de Referencia 2° Edición”. Edit. McGraw Hill
- **BUYENS, Jim.** “Aprenda Desarrollo de Base de Datos Web YA”. Edit. McGraw Hill
- **Rational Unified Process,** Best Practices for Software Development Teams Rational Software Corporation White Paper

DIRECCIONES WEB

<http://www.revista.unam.mx/vol.2/num1/art3/>

<http://www.elrincondelprogramador.com/default.asp?>

<http://www.programacion.net/java/>

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/fundamentos.htm>

">

<http://www.lalibriedigital.com.>

<http://www.almagesto.com.>

<http://www.algoritmodigital.com>

<http://www.eidos.es/alquimia>

<http://sestud.uv.es/manual.esp/>

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/intro.htm>">

href="http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/javaBeans/fundamento.htm#Definición de JavaBean">

GLOSARIO

APACHE TOMCAT : Sw utilizado en los servidores Web de tecnología Java

JSP. es una tecnología para generar páginas web de forma dinámica en el servidor. (mezcla de Html con JSP)

JDK (Java Development Kit) es un paquete para desarrollar cualquier aplicación Java, incluye el API de Java

COMPILADOR JAVA. Genera *Byte-Code*.

JVM ("Java Virtual Machine"), capaz de ejecutar *Byte-Code*.

UML. Lenguaje Modeling Unified, lenguaje de Modelado Unificado

CÓDIGO DE BARRAS. símbolos con imágenes formadas por combinaciones de barras y espacios paralelos, de anchos variables.

OPEN SOURCE libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

LINUX es un Sistema Operativo basado en UNIX, por lo que se trata de un sistema operativo muy seguro y poderoso, con ventajas adicionales: es gratuito y abierto.

VARIABLES DE ENTORNO almacenan valores que describen las propiedades del entorno de trabajo pueden ser diferentes para los usuario

MYSQL-FRONT ._ es una aplicación utilizada para trabajar con MySQL.

JAVABEANS._ son componente que tiene como característica principal ser reutilizados cumplen ciertas características en cuanto a su diseño, tienen como objetivo el lograr reducir al máximo el código Java

INTEROPERABILIDAD deberá ser escrito una vez, y puede ser ejecutado en diversas plataformas obteniéndose el mismo resultado.

INTROSPECCIÓN._ es la técnica de descubrimiento de propiedades, permite buscar las propiedades que coincidan con el patrón de nombres de métodos definido.

ANEXOS

ANEXO 1

Manual de Usuario

Información CD # 1

ANEXO 2

Manual de Programador

Información CD # 1

Latacunga, Septiembre 2005

Srta. Lilia Pilar Chingo Chiluisa

C.I 0502529258

Srta. Gladys Yolanda Masapanta Aimara

C.I 0502517519

Ing. Santiago Jácome

DECANO DE LA FACULTAD DE SISTEMAS E INFORMATICA

Dr.Eduardo Vázquez Alcázar

SECRETARIO ACADEMICO

