

Licenciatura en Informática

INGENIERÍA DE SOFTWARE

Nancy Obed Martínez Miguel

Luis Fernando Vásquez Fabián

Fernando López Salvador

M.T.C.A. ROLANDO PEDRO GABRIEL

Índice de contenido

0.1	Introducción	4
0.2	Propósito del proyecto	4
0.3	Viabilidad	4
0.4	Factibilidad	4
0.5	Ámbito del proyecto	5
0.5.1	Validación y Registro de usuario	5
0.5.2	Punto de venta	5
0.5.3	Ticket	5
0.5.4	Agregar registro de los productos	5
0.5.5	Modificación del producto	6
0.5.6	Eliminar registros	6
0.6	Mapa de Navegación	7
0.7	Interfaces	7
0.8	Requerimientos Funcionales	12
0.9	Requerimientos no Funcionales	14
0.10	Estándar de Codificación	14
0.11	Documentación y Casos de Uso	21
0.12	Personal Involucrado	33
0.13	Script	38
0.14	Diagrama de Secuencia	39
0.15	Diagrama de Despliegue	47
0.16	Diagrama de Componentes	49

Índice de Tablas

1	Requerimientos Funcionales 1	12
2	Requerimientos Funcionales 2	12
3	Requerimientos Funcionales 3	12
4	Requerimientos Funcionales 4	13
5	Requerimientos Funcionales 5	13
6	Requerimientos no Funcionales 1	14
7	Requerimientos no Funcionales 2	14
8	Requerimientos no Funcionales 3	14
9	Requerimientos no Funcionales 3	14
10	Personal Involucrado	33

0.1 Introducción

En la actualidad las herramientas tecnología se han convertido en uno de los factores más importantes en cuanto a los usos institucionales de las empresas, estas son utilizados diariamente para el manejo de información y su proyección social. Así que dentro del amplio ámbito de la informática se busca que día día se puedan lograr los avances necesarios para poder facilitar la vida de las personas, ya que conforme pasa tiempo surgen las necesidades de cambiar algunas cosas. Por lo que en este proyecto se enfoca en implementar un software de ámbito comercial, este software será diseñado para el control de un sistema de ventas de una tienda de abarrotes , buscando una optimización de los datos que esta maneja.

Por lo que estas aplicaciones mejoran el control administrativo llevando un seguimiento preciso de todas las transacciones que se realizan dentro de la tienda en un tiempo real, para generar reportes detallados de las cuentas que van permitir a los administradores la cantidad correcta de productos, en el momento adecuado lo cual va permitir al negocio o empresa mejorar servicio al cliente reduciendo cualquier anomalía. Los sistemas de ventas tienen la ventaja de ser personalizados para cumplir con las necesidades específicas de un negocio, por ejemplo, en la tienda las organizaciones de venta al menudeo pueden ayudar a localizando precios de venta de los productos y costos actuales de los productos para poder trabajar más rápidamente. En años recientes el mercado ha incrementado y se han ido generando diversos negocios o empresas en todas partes y eso ha llevado a las mismas ofrecer una mejor atención por lo que es muy importante contar con un sistema de ventas rápido y sencillo.

0.2 Propósito del proyecto

El sistema de ventas está orientada a la gestión de empresas comerciales o pequeños negocios, es un software practico que va permitir un mejor control del negocio. Se trata de un software para la gestión de clientes y productos. Donde puedes ingresar como administrador y realizar los que se necesite así sea ingresar, eliminar o modificar un producto o de ser un vendedor buscar los productos que el cliente desee y checar precios, cantidades en existencia y realizar su venta. Así mismo generara un informe de la venta y su respectivo ticket. El sistema se hace cargo de registrar las ventas del producto o servicio de la organización, evalúa sus características o necesidades del negocio, llevando un detalle del registro de todas las transacciones.

0.3 Viabilidad

Este proyecto es viable debido a que si contamos con las posibilidades para poder trasladarnos a dichos establecimientos al que se realizara el proyecto, todo esto con la finalidad de platicar con el cliente y así nos pueda proporcionar cada una de las necesidades o requerimientos que este quiera que se implemente al software.

0.4 Factibilidad

El proyecto es factible debido a que en la universidad de la sierra sur contamos con un centro de tecnologías de la informática y comunicación donde contamos con las herramientas necesarias ya que se cuenta con computadoras internet, y la asesoría de los docentes quienes

están altamente calificados para ayudarnos a desarrollar dicho software. A si mímoo también durante nuestra carrera hasta el día de hoy hemos aprendido conocimientos en distintos lenguajes de programación como C, Java, Css, Html y lenguaje ensamblador los cuales podemos aplicar en el proyecto a resolver.

0.5 **Ámbito del proyecto**

En la actualidad debido a la falta de conocimiento de muchas personas que administran establecimientos o pequeñas empresas dedicadas a las ventas no tienen en orden la administración de las mismas, como podrían ser las ventas, registros o actualizaciones de productos. Por lo que muchas veces suelen tener pérdidas de ganancias o desperdicio de la misma mercancía.

Pero también se sabe que en un establecimiento deben de quedar de manera ordenada los roles que se desempeñan dentro del lugar, es decir, ver que cosas puede hacer el administrador del negocio y que puede realizar el vendedor. Así que se desea implementar un sistema de control de ventas en una tienda de abarrotes y así dar solución a la problemática ya antes mencionada. Evitando así pérdidas monetarias y descontrol de su inventario, este sistema de venta estará diseñado de manera que el usuario pueda interactuar fácilmente y obtenga buenos resultados al utilizarlo.

Por lo que se describirá la estructura ya pensada de como se va ir desarrollando el sistema, el cual se va a distribuir de la siguiente manera:

0.5.1 **Validación y Registro de usuario**

En esta parte se implementará implantará la validación de usuario con la finalidad de mantener seguro los datos del usuario, evitando así que otras personas no autorizadas puedan acceder al sistema haciendo un mal uso del sistema de ventas. También ahora va depender del rol que entren al sistema, así se le irán mostrando sus partes por ejemplo si entra del lado del vendedor:

0.5.2 **Punto de venta**

Donde el vendedor va generar una venta es decir el cliente comienza a tomar los productos que serán mostrados desde una tabla y así puede ir viendo si tiene existencia de dicho producto, el precio y entre otras cosas.

0.5.3 **Ticket**

En esta parte se piensa que después de que el vendedor genere su venta le muestre como resultado un mensaje donde le detalle el nombre del producto, el precio y de igual manera le muestre la suma total de su compra. En caso contrario y se iniciará sesión como administradores pues lo primero que le mostraría es una ventana con tres botones que le darán acceso a agregar, modificar o eliminar algún producto.

0.5.4 **Agregar registro de los productos**

El usuario podrá realizar sus registros diarios de las ventas y adquisición de dispositivos nuevos, además podrá registrar las características de cada producto como nombre, id, cuantos hay existentes, precio entre otras.

0.5.5 Modificación del producto

En cualquier momento el usuario podrá acceder a sus registros para poder hacer las modificaciones necesarias siempre y cuando se requiera. En este caso se le mostrará un pequeño formulario muy similar al de agregar y así podrá modificar lo que necesite y así podrá dar en su botón guardar cambios.

0.5.6 Eliminar registros

El usuario podrá en cualquier momento eliminar los registros pasados que ya no necesite, esto con la finalidad de que no requiera de más espacio de memoria evitando de esta manera la necesidad de hacer más gastos innecesarios. Este apartado estará diseñado de manera simple para que el usuario pueda navegar sin problemas, además de que contará con aviso de alerta cuando el usuario elimine un registro, de esta forma el usuario estará seguro si desea eliminar tal registro, evitando así el riesgo de eliminar registros importantes.

Este sistema tendrá el objetivo de ser una gran herramienta que pueda ayudar al usuario llevar una mejor organización de sus operaciones en su negocio y también está claro que esta también debe cumplir las expectativas de cliente en cuanto al diseño del software ya que el diseño se realizará de forma que pueda ser más agradable y entendible para el usuario, este sistema contará con diferentes herramientas, como el registro de ganancias en donde podrá ver si su negocio está resultando así como también contará con página en donde puedan consultar los modelos productos que ha vendido y así el usuario podrá ver que compras son más recomendables les hará a sus proveedores.

0.6 Mapa de Navegación

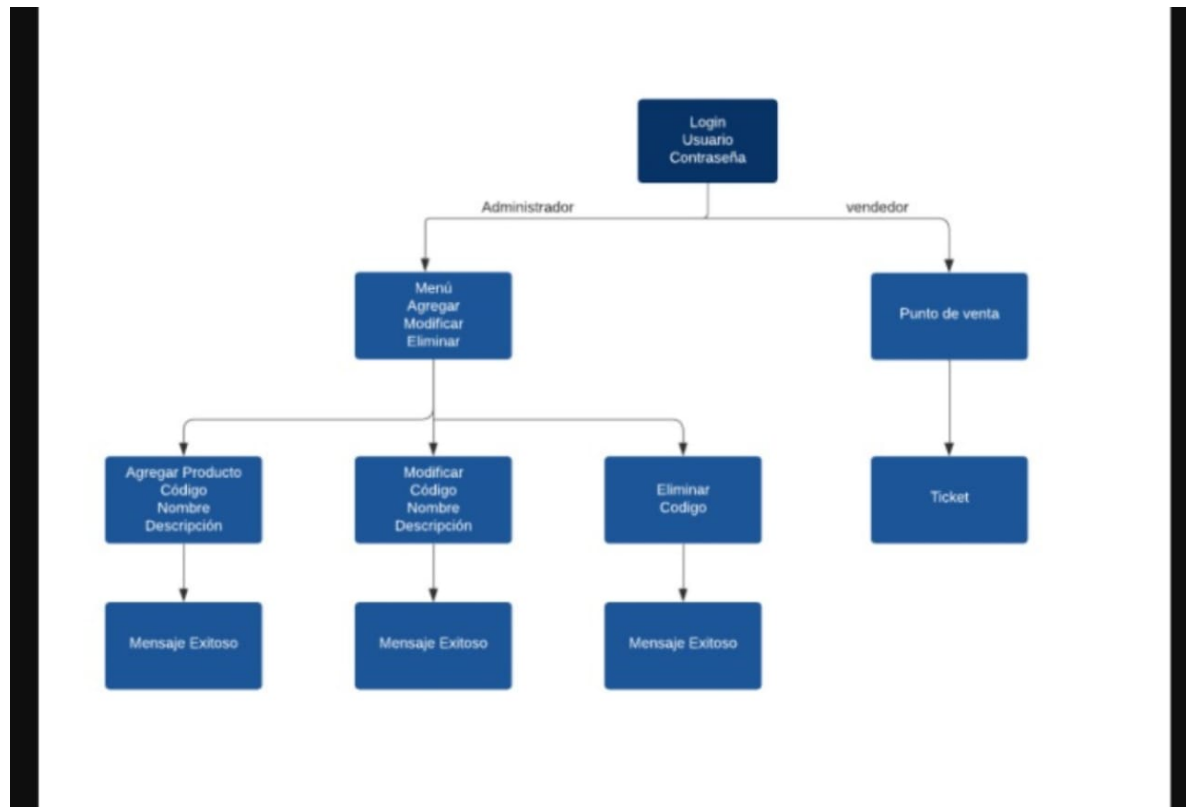



Figure 1: Mapa de Navegación

0.7 Interfaces

Una interfaz se utiliza en informática para nombrar a la conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles, permitiendo el intercambio de información. Por lo que este proyecto tiene las siguientes interfaces.

Validación de usuario.

En la siguiente ventana lo que nos muestra es el Login o el inicio de sesión en el cual es necesario ingresar los datos correctos para si poder ingresar al sistema, ya sea como Vendedor o como administrador para realizar las acciones necesarias.

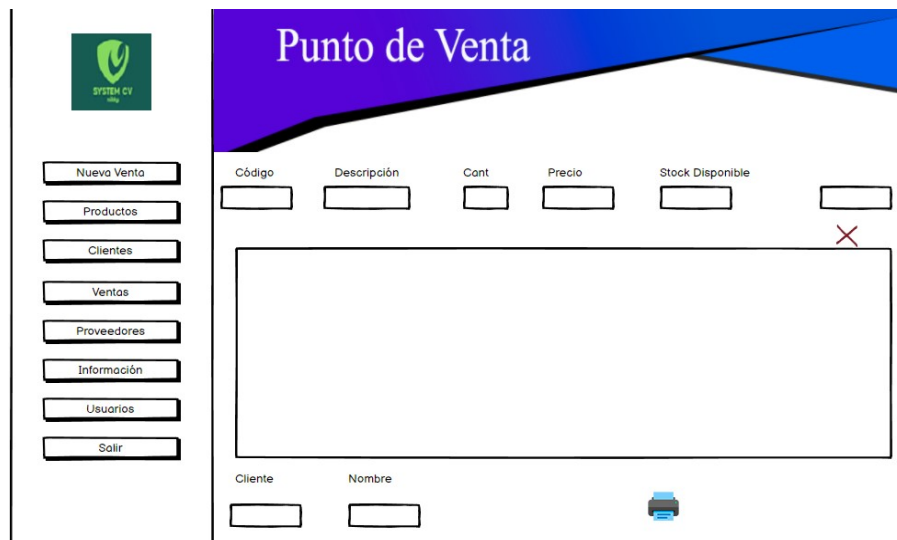


The login window is titled "Login" in the top left corner. It features a green logo with a shield and the text "SYSTEM CV" on the left side, with "Sistema de Ventas" written below it. On the right side, there is a blue padlock icon with a keyhole. Below the icon, there are two input fields: the first is labeled "Usuario" and the second is labeled "Contraseña". At the bottom right, there is a "Login" button.

Figura 1: Login

Punto de venta.

Una vez validado el inicio de sesión se muestra la siguiente ventana donde se realizaran la ventas las opciones que encontramos en esta ventana son *Nueva Venta*, *Productos*, *Cientes*, *Ventas*, *Proveedores*, *Información*, *Usuarios*, *Salir*. Para realizar una venta se necesita ingresar el código en la opción de código a continuación se muestra la descripción del producto, posteriormente se ingresa la cantidad de producto a comprar y se muestra el precio en caso de no contar con el producto se muestra un mensaje de Stock no disponible, la X es para eliminar un producto que no se desea comprar, en la parte inferior encontramos con el cliente en el cual ingresamos el código del cliente el cual desea realizar la venta una vez ingresado el código se muestra el nombre del cliente y para finalizar la venta se presiona en finalizar, finalizada la venta se muestra el tiket y para salir se presiona la opción salir que se encuentra al final en la izquierda de la ventana.



The "Punto de Venta" window has a blue header with the title "Punto de Venta". On the left side, there is a vertical menu with buttons for "Nueva Venta", "Productos", "Cientes", "Ventas", "Proveedores", "Información", "Usuarios", and "Salir". The main area contains a table with columns: "Código", "Descripción", "Cant", "Precio", and "Stock Disponible". Below the table, there are input fields for "Cliente" and "Nombre". A printer icon is located at the bottom right. A red "X" is visible in the top right corner of the main area.

Figura 2: Punto de Venta

Ventana de Clientes.

En caso de que el usuario ingrese como administrador la ventana que se muestra es la de clientes con la opción de agregar modificar o eliminar clientes, en caso de ingresar como vendedor solo estará visible la tabla que contiene a los clientes para así poder realizar consultas.



Figura 3: Clientes

Ventana de Inventario.

En la opción de inventario nuevamente si ingresamos como administrador estarán habilitadas las opciones para agregar, modificar o eliminar productos, en caso de ingresar como vendedor solo esta habilitado la tabla que contiene la información de los productos para facilitar las consultas.




Figura 4: Inventario

Ventana de Proveedores.

También contamos con la ventana de Proveedores en la cual almacenamos a todos los proveedores a los cuales compramos mercancía en la cual guardamos el nombre, dirección, teléfono y en la parte derecha contamos con una tabla la cual nos muestra el total de proveedores con los cuales contamos actualmente teniendo así un mejor control de nuestros proveedores.

Punto de Venta

Usuario

Nuevo Venta
Productos
Clientes
Ventas
Proveedores
Información
Usuarios
Salir

Registro Proveedores

Código

Nombre

Telefono

Dirección

+ -

✖

+

Figura 5: Proveedores

Ventana de Ventas.

En esta opción nos permitirá mostrar o consultar las ventas totales realizadas, en dicha tabla nos muestra el ID de la venta el nombre del cliente, el nombre del vendedor y el total a pagar, en esta ventana no están habilitadas las modificaciones simplemente es para poder hacer las consultas de la ventas realizadas previamente.

Punto de Venta

Usuario

Nuevo Venta
Productos
Clientes
Ventas
Proveedores
Información
Usuarios
Salir

Ventas

Figura 6: Ventas

Ventana de Usuarios.

En la opción de Usuarios nos permite ingresar nuevos usuarios al sistema dándole cierto privilegio en caso de ser administrador y en caso de ser vendedor simplemente los privilegios básicos.

The image shows a web application interface for a 'Punto de Venta' (Point of Sale) system. The header is blue with the title 'Punto de Venta' in white. Below the header, there is a sidebar on the left and a main content area on the right.

Sidebar Menu:

- Usuario
- Nueva Venta
- Productos
- Clientes
- Ventas
- Proveedores
- Información
- Usuarios
- Salir

Main Content Area:

The main content area is titled 'Registro Usuarios' (User Registration). It contains a form with the following fields:

- Usuario: Text input field.
- Password: Text input field.
- Nombre: Text input field.
- Rol: Dropdown menu with 'Rol Usuario' selected.

A 'Registrar' (Register) button is located at the bottom of the form.

The right side of the main content area is a large table with 10 empty rows, intended for displaying a list of users.

Figura 7: Usuarios

Ventana de Información.

En la opción de Información nos permite acceder a los datos de la tienda así como el nombre la dirección el teléfono o también un mensaje que desee mostrar la tienda al momento de realizar una compra , en caso de ingresar como administrador este nos permitirá modificar dicha información, si ingresa como vendedor no tendrá permitido modificar los datos en esta ventada pero se mostraran de igual manera al terminar una venta.

The image displays a web application interface for a Point of Sale (Punto de Venta) system. The header is a solid blue bar with the text "Punto de Venta" in white. Below the header, the main content area is divided into three sections. On the left is a sidebar menu with a green logo at the top that says "SYSTEM CV". The menu is titled "Usuario" and contains eight buttons: "Nueva Venta", "Productos", "Clientes", "Ventas", "Proveedores", "Información", "Usuarios", and "Salir". The central section is titled "Datos de la Tienda" and contains a form with a "Datos" dropdown menu. Below the dropdown are four input fields arranged in a 2x2 grid: "Código", "Nombre", "Dirección", and "Dirección". At the bottom of this section is a "Mensaje" label and a long text input field, followed by an "Actualizar" button. On the right side of the interface is a decorative illustration of a city skyline with blue buildings, a red sun, and birds. The overall design is clean and modern, using a color palette of blue, green, and white.

Figura 8: Información de la tienda

0.8 Requerimientos Funcionales

Código de requerimiento	RF01
Nombre	Envío y recepción de archivos
Propósito	Permitir el acceso de información a la base de datos así como la facilidad para enviar información
Descripción	El usuario realiza la encuesta para la información, a continuación acepta los términos para así poder enviar su información a la base de datos
Entrada	Formulario de perfil del cliente
Salida	Mensaje de acción satisfactoria
Prioridad	Alta

Table 1: Requerimientos Funcionales 1

Código de requerimiento	RF02
Nombre	Identificación
Propósito	Para saber si al iniciar sesión es cliente o dueño.
Descripción	Primero da clic en el botón iniciar sesión después podrá poner su contraseña y mirar como administrador o usuario
Entrada	Contraseña
Salida	Regresar a la pantalla principal
Prioridad	Alta

Table 2: Requerimientos Funcionales 2

Código de requerimiento	RF03
Nombre	Operaciones Crud
propósito	Crear, Actualizar, borrar usuarios
descripción	Una vez ubicados en la página inicial del software el usuario debe capturar los datos de cliente
Entrada	Formulario de entrada de llenado de información del cliente
Salida	Mensaje que indique que el registro fue exitoso
Prioridad	Alta

Table 3: Requerimientos Funcionales 3

Código de requerimiento	RF04
Nombre	Facturación de pedidos de venta
propósito	Crear la factura de la venta
descripción	La facturación de pedidos de venta se realizara por medio de una pantalla de pedidos pendientes de facturación.
Entrada	Formulario de entrada de llenado de información de la compra
Salida	Mensaje que indique que el proceso fue exitoso
Prioridad	Media

Table 4: Requerimientos Funcionales 4

Código de requerimiento	RF05
Nombre	Maestro de clientes
propósito	Mantener un mejor control sobre la información de los clientes
descripción	El sistema manejará un registro maestro de clientes. Sólo los usuarios autorizados podrán ingresar nuevos clientes, modificar los datos o eliminarlos.
Entrada	Formulario de entrada de llenado de información del cliente
Salida	Mensaje que indique que el proceso fue exitoso
Prioridad	Alta

Table 5: Requerimientos Funcionales 5

0.9 Requerimientos no Funcionales

Código de requerimiento	RNF01
Nombre	Actualización del software
propósito	Realizar mantenimiento y actualizaciones del software
descripción	Ir a la empresa a darle mantenimiento al software y actualizar bibliotecas

Table 6: Requerimientos no Funcionales 1

Código de requerimiento	NRF02
Nombre	Mantenimiento
propósito	Un software con un buen funcionamiento
descripción	Las acciones que se harán para mejorar el software (cambio de contraseña, alguna mejora, limpieza de registro de sistema.

Table 7: Requerimientos no Funcionales 2

Código de requerimiento	NRF03
Nombre	Aplicación para celular
propósito	El software no contara con aplicación para celulares
Prioridad	baja

Table 8: Requerimientos no Funcionales 3

Código de requerimiento	NRF04
Nombre	Ordenes de entrega
propósito	El sistema no contara con opcion para envio a domicilio
Prioridad	baja

Table 9: Requerimientos no Funcionales 3

0.10 Estándar de Codificación

Estándar de codificación Java

Nomenclatura

El idioma por defecto a la hora de dar sentido funcional al nombre de clases, variables, constantes, etc. será una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada.

Resumiendo, aquella codificación que por estandarización y/o aceptación se pueda escribir en inglés se mantendrá así por convenio, casos como insert, update, delete, create, retrieve, list, set, get, newInstance, Delegate.

Para la parte funcional se utilizara castellano, por lo tanto la nomenclatura de los métodos será: getListEmpresa en sustitución de getListCompany o insertBanco en lugar de insertarBanco.

Paquetes

Por defecto todos los paquetes se escribirán en minúsculas y sin utilizar caracteres especiales. El paquete base queda definido como es.gobcantabria, en este paquete no se definirá ninguna clase.

Se tendrá, así mismo, otro nivel extra dentro del paquete definido como el nombre del proyecto o del módulo (Ej. es.gobcantabria.xxxx).



Si existiera una parte común a varios de estos módulos, el nombre de los paquetes comenzarán por: es.gobcantabria.common

La estructura en árbol de los paquetes siguientes a partir de este último se define como sigue: (presuponiendo que se trata de una aplicación web multicapa)

- bussines
 - dao
 - domain
 - service
 - helper
 - exception
- util
- web
 - controller
 - model
 - view

Nombres de Interfaces

Los nombres de interfaces utilizarán el sufijo **Interface** y estarán compuestos por palabras con la primera letra en mayúscula (CamelCase). Se debe evitar el uso de abreviaciones que dificulten la comprensión del código.

Ejemplo: ConexionInterface, ComponenteTablaInterface

Nombres de clases

Los nombres de clases deben ser mezclas de mayúsculas y minúsculas, con la primera letra de cada palabra interna en mayúsculas (CamelCase).

Debemos intentar mantener los nombres de clases simples y descriptivos.

Debemos usar palabras completas y evitar acrónimos y abreviaturas (se permiten DAO, DTO, URL, HTML, etc.).

Si la clase cumpliera algún patrón determinado o tuviera una funcionalidad específica es recomendable definirlo en el nombre.

Paquete	Funcionalidad	Nombre
bussines.dao	Data Access Object (Interface)	<nombre> DAO
bussines.dao.impl	Data Access Object (Implementation)	<nombre> DAOImpl
bussines.exception	Excepciones	<nombre> Exception
bussines.service	Service	<nombre> Service
bussines.helper	Helper	<nombre> Helper
bussines.dto	Data Transfer Objects	<nombre> DTO
util	Clases de constantes.	<scope> Keys <nombre> Keys
web.controller	Controller	<nombre> Controller
web.filter	Filter	<nombre> Filter
web.model	Model	<nombre> Model
web.listener	Listener	<nombre> Listener

Nombres específicos de gestiones

Cuando se trata de gestionar una entidad determinada (Ej. Usuario) se definen los nombres de clases, demás ficheros implicados con la siguientes reglas:

Clase: <<FuncionalidadGenerica>><<Entidad>><<Especificación de Clase>>

Ejemplo: Usuario

UsuarioAction, FindUsuarioAction

Métodos

Los métodos deberán ser verbos (en infinitivo), en mayúsculas y minúsculas con la primera letra del nombre en minúsculas, y con la primera letra de cada palabra interna en mayúsculas (lowerCamelCase).

No se permiten caracteres especiales.

El nombre ha de ser lo suficientemente descriptivo, no importando a priori la longitud del mismo.

Variables

Los nombres de las variables tanto de instancia como estáticas reciben el mismo tratamiento que para los métodos, con la salvedad de que aquí sí importa más la relación entre la regla mnemónica y la longitud del nombre.

Ejemplo:

- Correctos: diaCalculo, fechaIncorporacion
- Incorrectos: dC, Dcal, fl, Fl¿

Se evitará en la medida de lo posible la utilización de caracteres especiales, así como nombre sin ningún tipo de significado funcional.

Las excepciones son las variables utilizadas en bucles for, para esos casos se permite utilizar i, j, k, l y siempre en ese orden de anidamiento.

El primer bucle siempre será el que tenga la variable i como iterador. (Esta variable se definirá para el bucle en cuestión).

Constantes

Los nombres de constantes de clases deberían escribirse todo en mayúsculas con las palabras separadas por subrayados ("_"). Todas serán declaradas como public static final

```
public static final String PROPERTY_URL_SERVICIO = "urlServicio";
```

Estilo de codificación

Comentarios

Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase. Se tiene, por tanto, que evitar referencias al diseño funcional de la misma.

El uso abusivo de los comentarios es desaconsejable, principalmente por el trabajo extra necesario para su correcto mantenimiento. Es preferible rediseñar el código para una mejor compresión del mismo.

Estándar de Codificación JEE 13ESTÁNDAR DE CODIFICACIÓN JEE

Se tienen que evitar el uso de caracteres especiales dentro de los comentarios así como el uso de cajas u otro tipo de gráfico creado mediante códigos ASCII.

La estructura de los diferentes tipos de comentarios y su uso general se presenta en la clase base de codificación.

Comentarios de Documentación (JavaDoc)

Como norma es obligatorio proporcionar un comentario de documentación por cada clase / interface, método, propiedad o constante creado.

Son obligatorios los siguientes comentarios de documentación:

Comentario de la clase / interface:

- Prescripción genérica de la clase y su responsabilidad.
- Autor.

Todas las variables tipo `private` o `protected` han de ser obligatoriamente comentadas.

Reglas generales a la hora de escribir comentarios de documentación.

- Siempre se escribe en tercera persona.
- Los caracteres especiales tales como tildes y eñes se han de codificar con su código HTML correspondiente.
- Las descripciones siempre deberían empezar por un verbo.

En cuanto a los tag's utilizados el orden de los mismos es el siguiente:

- **@param** seguido de el nombre del parámetro e indentada la descripción del mismo. Usualmente esta descripción será una frase corta que comienza definiendo el tipo del parámetro.

`@param alturaCaja` Entero que define la altura de la caja en píxeles

`@param longitudCaja` Flotante que define la longitud de la caja en píxeles

- **@return** este tag no aparece para aquellos métodos que retornan `void`. Por lo demás se comporta como el tag anterior
- **@throws** Descripción breve de la posible causa de la excepción.
- **@see** Su uso queda restringido en cuanto a cantidad de los mismos como con el atributo **@link**.

Los demás tags permitidos (`@since`, `@serial`, etc..) el uso es menos común y por lo tanto no se define una manera de utilizarlos.

Usar el atributo `<code>` para las palabras reservadas de java, nombres de clases, métodos, interfaces, propiedades, argumentos y ejemplos de código.

El uso del atributo **@link** tendrá que ser mínimo, para evitar llenar el documento de enlaces. Como norma general solo se incluirán links cuando la referencia sea necesaria y sólo en la primera aparición de la misma.

Para distinguir entre dos métodos con el mismo nombre pero diferentes parámetros de entrada se utilizarán el nombre del método seguido por los argumentos del mismo entre paréntesis.

Declaraciones

Para la declaración de las variables se utiliza una declaración de cada vez y no se permiten dejar variables locales sin inicializar salvo en el caso de que sean propiedades de un objeto bean.

La codificación correcta sería:

```
public static Integer entero = new Integer(0);
```

La declaración de las variables locales a una clase, método o bloque de código se realizan al principio del mismo y no justo antes de necesitarse la utilización de la variable.

La única excepción a esta regla son las variables que gestionan los bucles for.

Las variables de avance de bucles for no podrán ser modificadas de ninguna manera fuera de la propia sentencia del bucle.

La duplicidad de los nombres de variables en diferentes niveles dentro de la misma clase se tiene que evitar.

Sentencias

Normas básicas son :

- Una sentencia por línea de código.
- Todo bloque de sentencias entre llaves, aunque sea una sola sentencia después de un if.

La declaración de los bucles for serán usualmente de la forma :

```
for (int i = 0; i < condicion ; i++)
```

Son obligatorias las tres condiciones del bucle for: inicialización, condición de finalización y actualización del valor de la variable de avance.

La variable de avance del bucle nunca podrá ser modificada dentro del propio bucle.

Buenas prácticas

Constantes

Como norma general todas las constantes numéricas no deberían codificarse directamente, salvo la excepción de -1, 0 y 1.

Propiedades

El acceso/modificación de las propiedades de una clase (no constantes) siempre mediante métodos de acceso get/set.

La asignación de variables / propiedades no podrá ser consecutiva.

Variable1 = variable2 = "hola mundo"

No válido

No utilizar el operador asignación en sitios donde se pueda confundir con el operador igualdad. Ni dentro de expresiones complejas.

Métodos

Como norma general no se debe acceder a un método estático desde una instancia de una clase, debemos utilizar la clase en sí misma.

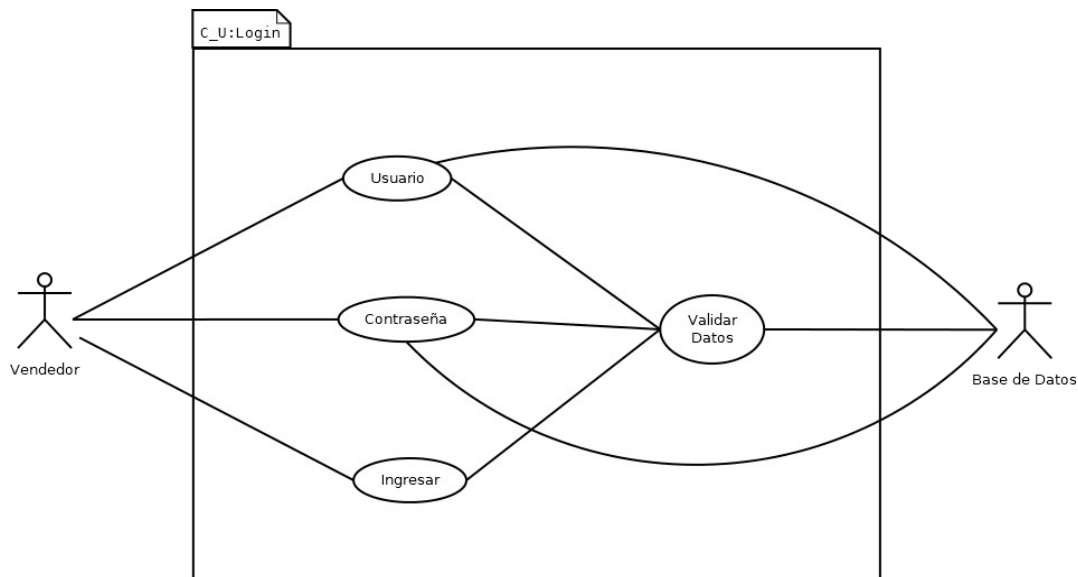
0.11 Documentación y Casos de Uso

Documentación y casos de uso

CASO DE USO

Id:	CU – 0		
Nombre:	Login		
Creado por:	L.S.Fernando	Actualizado por	L.S. Fernando
Fecha de creación:	3/12/21	Fecha de ultima revisión:	06/12/21
Actores:	Base de datos, Administrador		
Descripción:	Muestra pantalla para iniciar sesión		
Disparador:			
Predi-condiciones:			
Pos-condiciones:			
Flujo normal:	1.- Ingresar el nombre del usuario con el que se va ingresar al sistema. 2.- Ingresar la contraseña que corresponde al usuario con el que va ingresar 3.- Oprimir el botón “Ingresar” . 4.-En caso de no encontrar al usuario o la contraseña limpia los campos y muestra un mensaje de datos incorrectos. 5.- Ingresar al sistema.		
Flujos alternos:			
Includes:	N/A		
Frecuencia de uso:	Frecuente.		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Issues y Notes:	N/A		

Caso de uso 1: Login

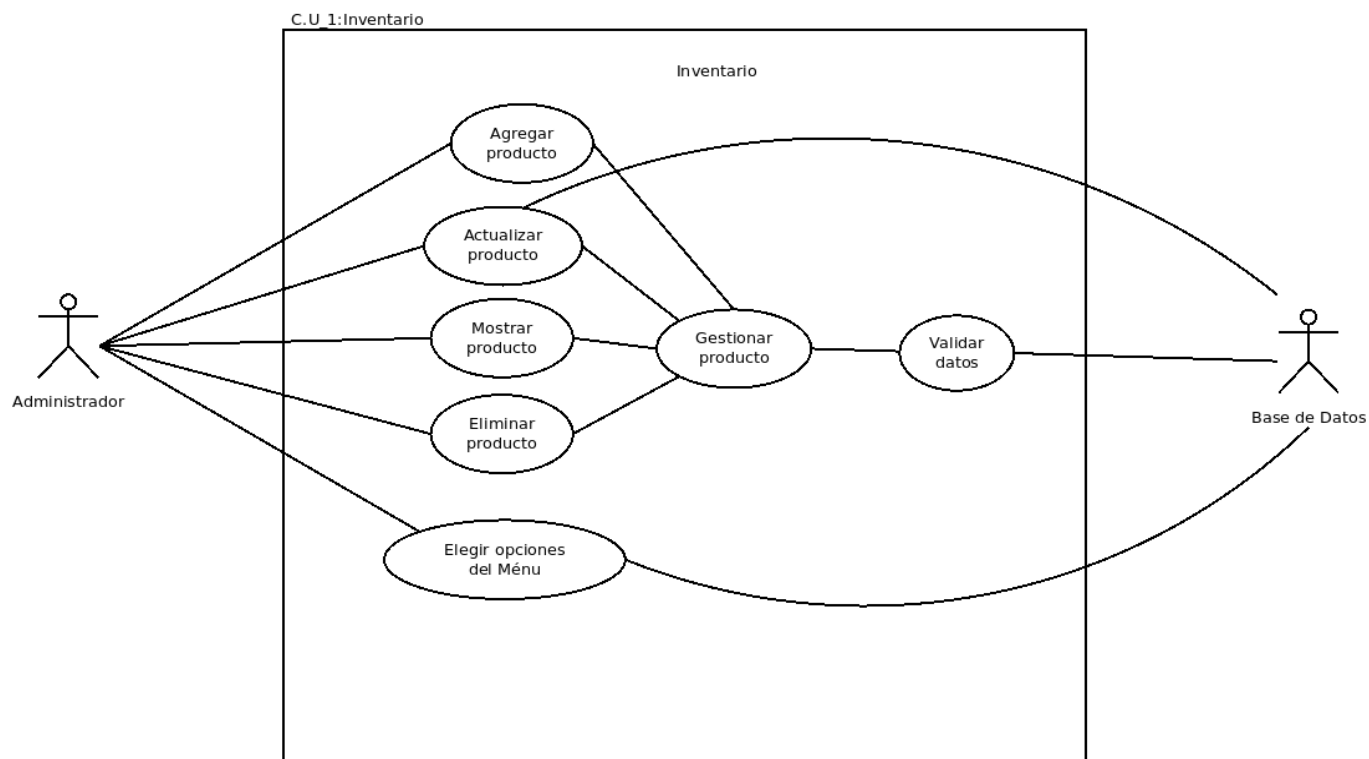


CASO DE USO

Id:	CU – 1		
Nombre:	Inventario		
Creado por:	L.S.F	Actualizado por	V.F.L.F
Fecha de creación:	01/12/21	Fecha de ultima revisión:	01/12/21
Actores:	Base de datos y Administrador.		
Descripción:	Se agregan los productos al inventario		
Disparador:	En el menú de la pantalla 2, presionar el botón Agregar producto		
Predi-condiciones:			
Pos-condiciones:	Se guardan los datos y se limpian los campos de la ventana 3		
Flujo normal:	<ol style="list-style-type: none"> 1.- Ingresar el código del producto 2.- Ingresar el nombre del producto <ol style="list-style-type: none"> 2.1 Si ingresa un numero le mostrará una ventana emergente con un mensaje “El dato debe ser ingresado con letras” 3.- Agregar una breve descripción del producto <ol style="list-style-type: none"> 3.1-Si ingresa un numero le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con letras” 4.-Agregar el precio de compra <ol style="list-style-type: none"> 4.1-Si ingresa una letra le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con números” 5.- Agregar el precio de venta <ol style="list-style-type: none"> 5.1-Si ingresa una letra le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con números” 6.- Ingresar la cantidad de productos que se agregaran 7.- Oprime al botón “Agregar” <ol style="list-style-type: none"> 7.1- Mostrará una ventana emergente con el mensaje “Datos ingresados correctamente”. 7.2- Oprimir el botón “ok” para cerrar la pantalla emergente 8.- Si oprimen el botón Actualizar se activan los campos para hacer las actualizaciones . 9.- si Oprimes el botón Mostrar Producto se despliega una tabla con todos los productos 		
Flujos alternos:	Para cerrar ésta pantalla le dan clic la pestaña Cerrar sesión		
Includes:	N/A		
Frecuencia de uso:	Frecuente.		

Requerimientos especiales:	N/A
Supuestos:	N/A
Issues y Notes:	N/A

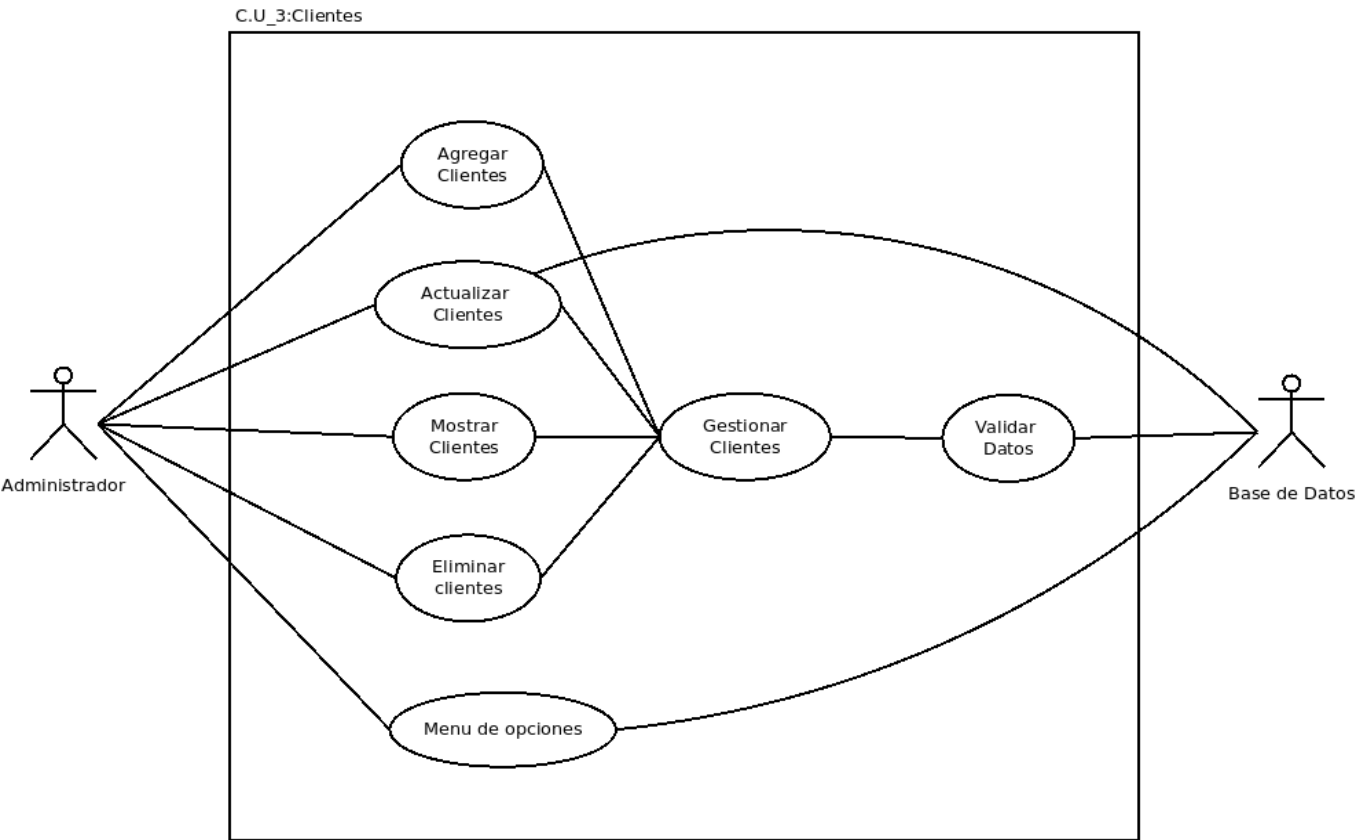
Caso de uso 2:Inventario



CASO DE USO			
Id:	CU – 3		
Nombre:	Clientes		
Creado por:	L.S.Fernando	Actualizado por	V.F.L.Fernando
Fecha de creación:	Fecha de ultima revisión:		

	2/12/21	3/12/21
Actores:	Base de datos, Administrador.	
Descripción:	Agregar, elimina y modifica nuevos clientes	
Disparador:	En el menú de la pantalla 3, oprimir el botón Agregar clientes	
Predi-condiciones:		
Pos-condiciones:	Se guardan los datos y se limpian los campos	
Flujo normal:	1.- Ingresar el nombre del cliente 2.- Ingresar su apellido 2.1 Si ingresa un número le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con letras” 3.- Ingresar el nombre de usuario 3.1-Si ingresa un número le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con letras” 4.- ingresar la contraseña 4.1-Si ingresa un número le mostrara una ventana emergente con un mensaje “El dato debe ser ingresado con letras” 5.- Oprimir el botón “ Agregar cliente ” 5.1- Si alguno de los campos no esta relleno se enviara un mensaje “Rellenar todos los campos” 5.1.1-Oprimir el botón OK para salir de la ventana emergente y regresa al paso 5. 5.2- Si los datos son correctos aparecerá el mensaje “se agregó correctamente” 5.2.1.- presionar OK para salir de la ventana se limpian los campos y regresa. 7.-si oprime el botón “ Modificar Cliente ”se muestra los datos de cliente para que se pueda modificar. 8.- se Oprime el botón” Eliminar cliente ” y se elimina el cliente seleccionado	
Flujos alternos:	Se cierra la ventana dando clic en pestaña “ Cerrar sesión ”	
Incluye:	N/A	
Frecuencia de uso:	Frecuente.	
Requerimientos especiales:	N/A	
Supuestos:	N/A	
Issues y Notes:	N/A	

Caso de uso 3 Clientes

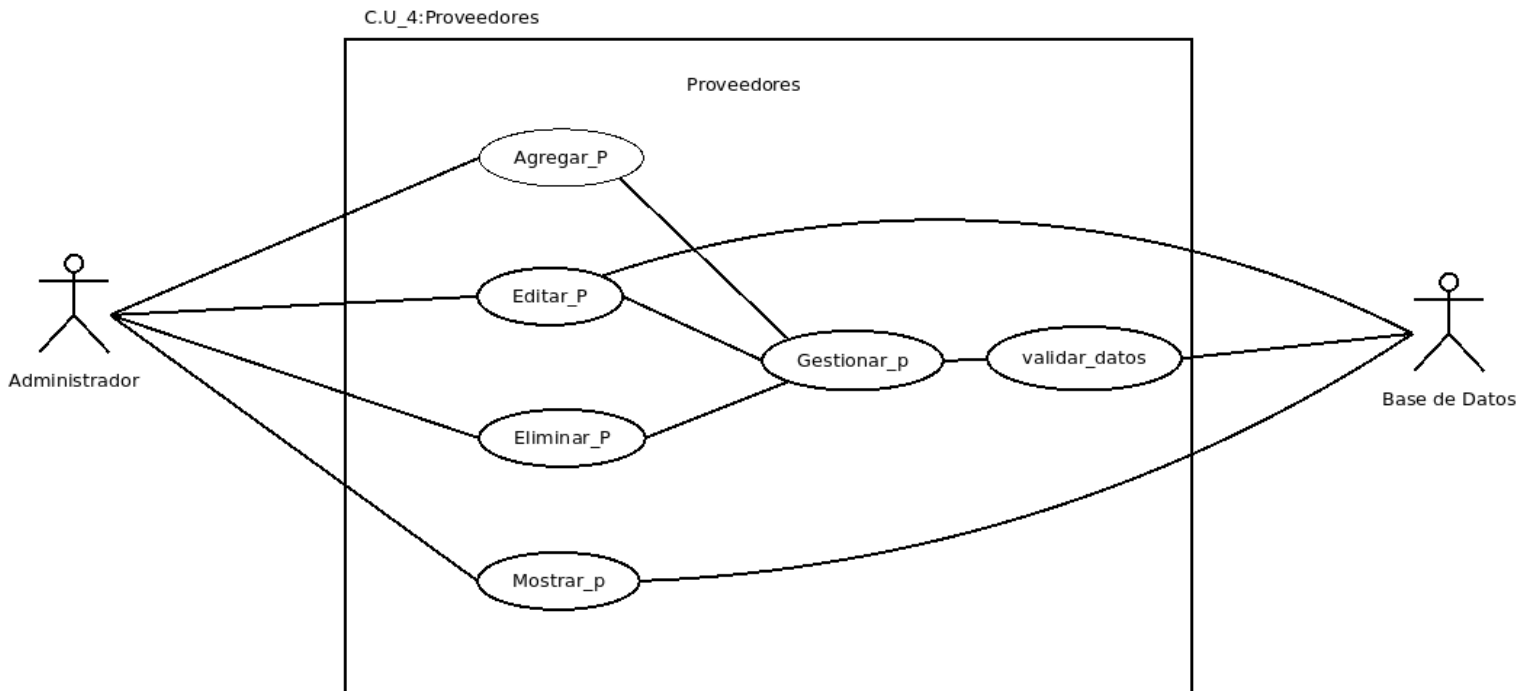


CASO DE USO			
Id:	CU – 4		
Nombre:	Proveedores		
Creado por:	V.F.L.Fernando	Actualizado por	M.M.Nancy Obed
Fecha de creación:	06/12/21	Fecha de ultima revisión:	
Actores:	Base de datos y administrador		
Descripción:	Se hacen los registros,modificaciones y borrados de proveedores		
Disparador:	Inicio de sesión		
Predi-condiciones:			
Pos-condiciones:			
Flujo normal:	1.- En el menú principal elegir la opción “ Archivo ” , en el submenu seleccionar “ Nueva venta ” 2.- Aparecerá una tabla. 2.-Para agregar un producto a la venta escribir el código en la caja de texto		

- 4.- En la tabla aparecerán los datos del producto al que corresponde ese código
- 5.-En la parte superior aparecerá el subtotal y posteriormente el total en caso de aplicar algún descuento
- 6.-Oprimir el botón aceptar
 - 6.1.-Guarda los datos de la venta
- 7.- El botón cancelar cancelará la venta y limpiara la tabla

Flujos alternos:	Regresa a la pantalla 2
Includes:	N/A
Frecuencia de uso:	Frecuente.
Requerimientos especiales:	N/A
Supuestos:	N/A
Issues y Notes:	N/A

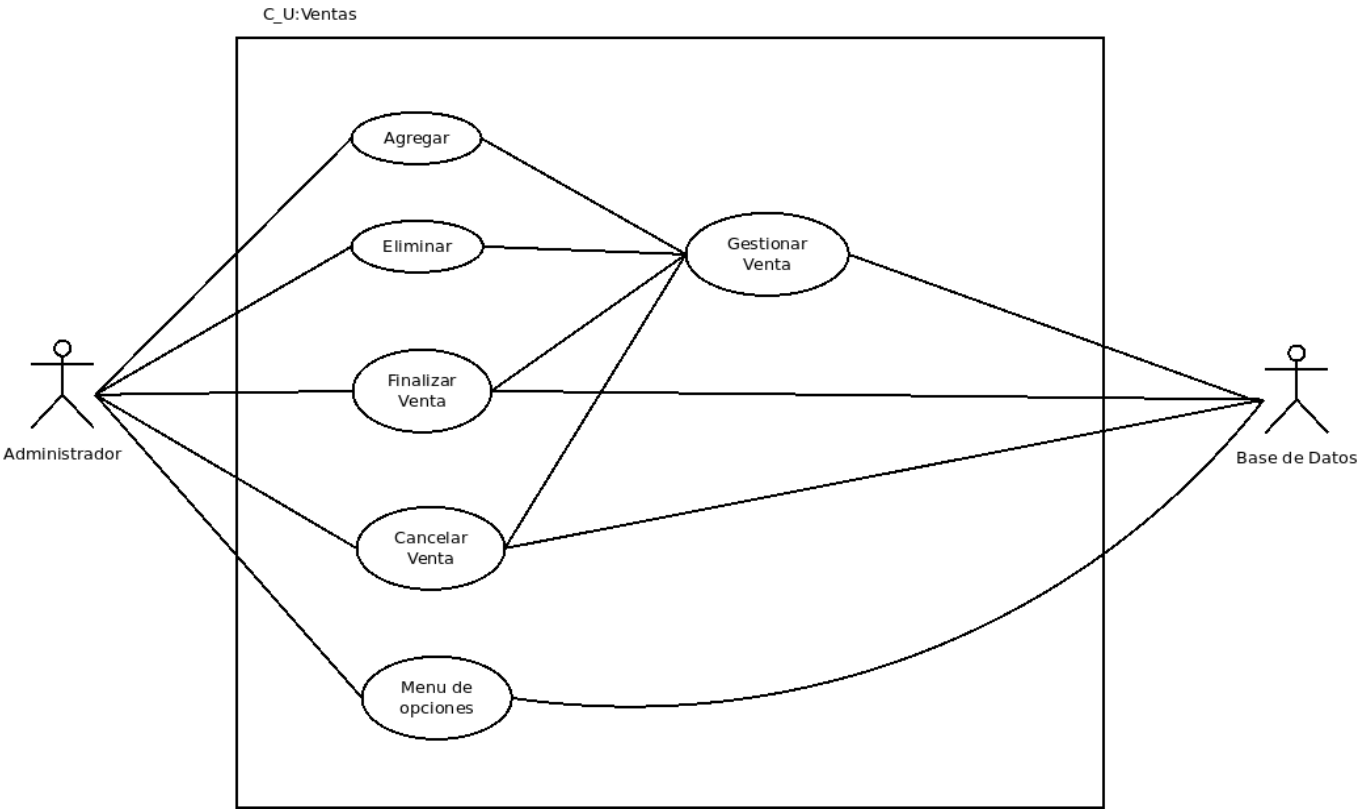
Caso de uso 4 Proveedores



CASO DE USO

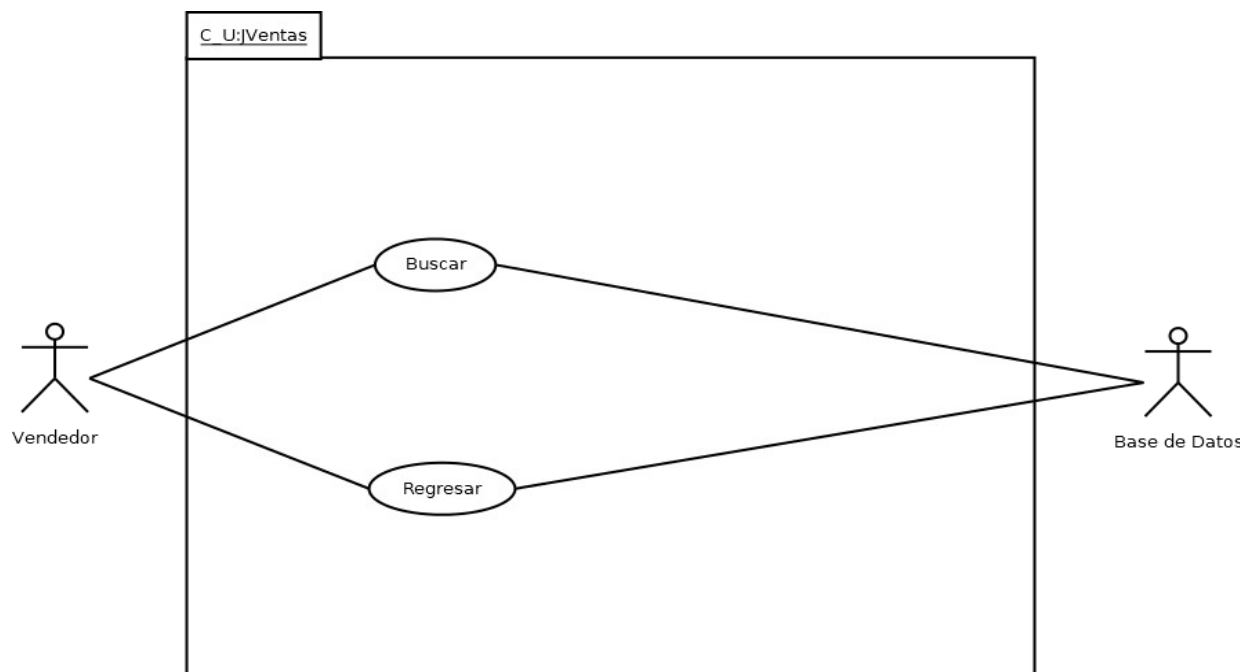
Id:	CU – 5		
Nombre:	Ventas		
Creado por:	Actualizado por		
	L.S.Fernado		L.S.Fernando
Fecha de creación:	Fecha de ultima revisión:		07/12/21
	06/12/21		
Actores:	Base de datos y vendedor		
Descripción:	Ejecuta venta de productos		
Disparador:	Inicio de sesión		
Predi-condiciones:			
Pos-condiciones:			
Flujo normal:	<div>1. Oprimes el Botón “Agregar “ para comenzar a agregar un producto a la venta.</div> <div>2.-Si ya terminas sus compras le das un clic al botón Finalizar venta</div> <div>3.- También si durante la venta el cliente ya no termina o ya no quiere las cosas le oprimes el botón Cancelar venta”</div> <div>4 .- para moverte en esta pantalla tiene tres pestañas</div> <div>4.1 si quieres irte a proveedores le das clic en la pestañas proveedor.</div>		
Flujos alternos:	Para cerrar esta pantalla le dan clic a la pestaña cerrar sesión		
Includes:	N/A		
Frecuencia de uso:	Frecuente.		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Issues y Notes:	N/A		

Caso de uso 5:Ventas



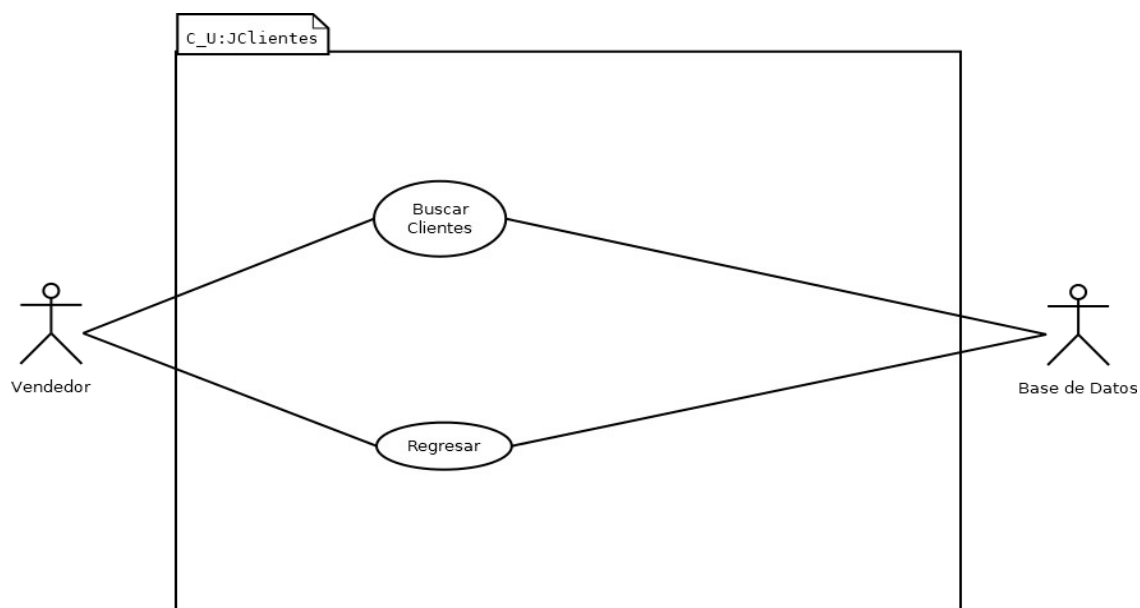
CASO DE USO		
Id:	CU – 6	
Nombre:	JVentas	
Creado por:	L.S.Fernado	Actualizado por M.M.Nancy Obed
Fecha de creación:	05/12/21	Fecha de ultima revisión:
Actores:	Base de datos y administrador	
Descripción:	Se hacen las consultas de los registros de ventas	
Disparador:	Ventas	
Predi-condiciones:		
Pos-condiciones:		
Flujo normal:	1.- En el menú principal elegir la opción “General” , en el submenu seleccionar “Ventas” 2.- Aparecerá una tabla. 3.-Para regresar pulsar el botón “Regresar”	
Flujos alternos:	Regresa a la pantalla Ventas	
Includes:	N/A	
Frecuencia de uso:	Frecuente.	
Requerimientos especiales:	N/A	
Supuestos:	N/A	
Issues y Notes:	N/A	

Caso de uso 6:JVentas



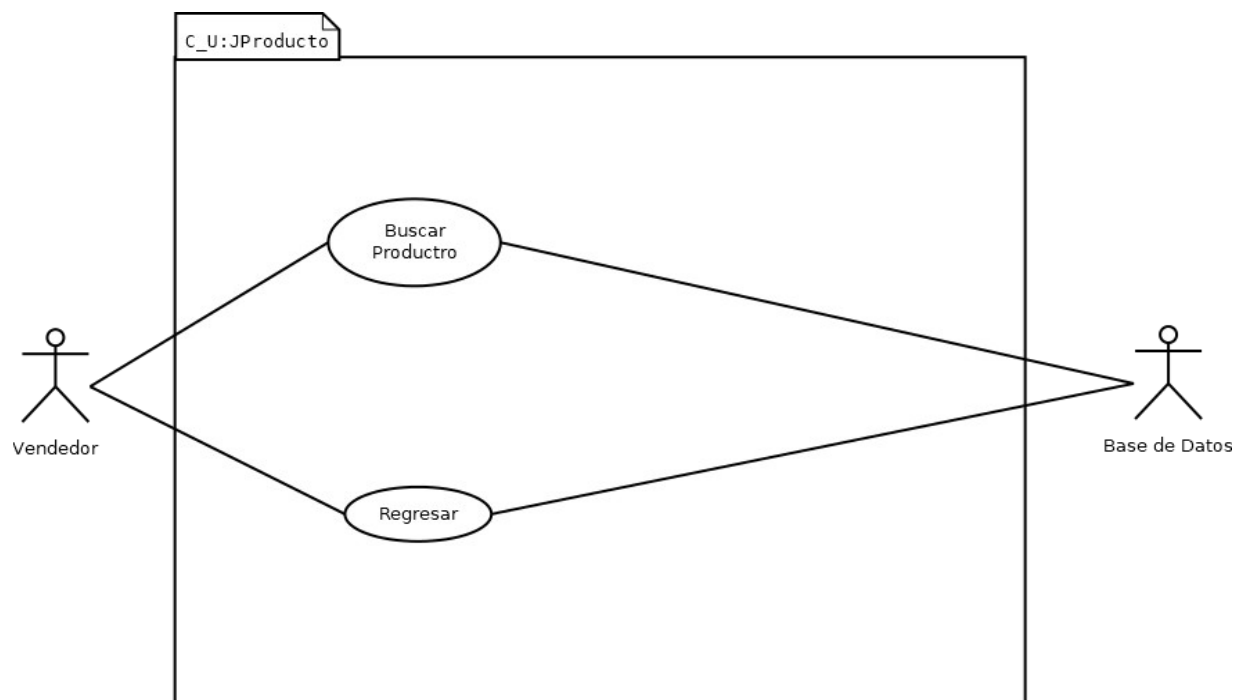
CASO DE USO		
Id:	CU – 7	
Nombre:	JClientes	
Creado por:	M.M.Nancy Obed	Actualizado por L.S.Fernado
Fecha de creación:	05/12/21	Fecha de ultima revisión:
Actores:	Base de datos y administrador	
Descripción:	Se hacen las consultas de los registros de Clientes	
Disparador:	Ventas	
Predi-condiciones:		
Pos-condiciones:		
Flujo normal:	1.- En el menú principal elegir la opción “General” , en el sub-menu seleccionar “Clientes” 2.- Aparecerá una tabla. 3.-Si desea realizar una búsqueda en especifico ingresar dicho dato en el espacio en blanco de la pantalla. 3.-Para regresar pulsar el botón “Regresar” .	
Flujos alternos:	Regresa a la pantalla Ventas	
Includes:	N/A	
Frecuencia de uso:	Frecuente.	
Requerimientos especiales:	N/A	
Supuestos:	N/A	
Issues y Notes:	N/A	

Caso de uso 6:JClientes



CASO DE USO			
Id:	CU – 7		
Nombre:	JProducto		
Creado por:	Actualizado por		
	V.F.L.Fernando	L.S.Fernado	
Fecha de creación:	Fecha de ultima revisión:		
	05/12/21		
Actores:	Base de datos y administrador		
Descripción:	Se hacen las consultas de los registros de Productos		
Disparador:	Ventas		
Predi-condiciones:			
Pos-condiciones:			
Flujo normal:	1.- En el menú principal elegir la opción “ General ” , en el sub-menu seleccionar “ Producto ” 2.- Aparecerá una tabla. 3.-Si desea realizar una búsqueda en especifico ingresar dicho dato en el espacio en blanco de la pantalla. 3.-Para regresar pulsar el botón “ Regresar ”.		
Flujos alternos:	Regresa a la pantalla Ventas		
Includes:	N/A		
Frecuencia de uso:	Frecuente.		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Issues y Notes:	N/A		

Caso de uso 6:JProducto



0.12 Personal Involucrado

Nombre	Roles
Nancy Obed Martínez Miguel	Master scrum
Luis Fernando Vásquez Fabián	Desarrollador
Fernando López Salvador	Auxiliar
Entrada	Formulario de perfil del cliente
M.T.C.A.Rolando-Pedro- Grabriel	Usuario Final
M.T.E.Everardo-de-Jesús- Pacheco Antonio	Programador
M.C.CLirio Ruiz Guerra	Administrador de la Base de Datos

Table 10: Personal Involucrado

Descripción Base datos

El sistema de ventas está orientada a la gestión de empresas comerciales o pequeños negocios, es un software practico que va permitir un mejor control del negocio. Se trata de un software para la gestión de clientes y productos. Donde puedes ingresar como administrador y realizar los que se necesite así sea ingresar, eliminar o modificar un producto o de ser un vendedor buscar los productos que el cliente desee y consultar precios, cantidades en existencia y realizar su venta. Así mismo generará un informe de la venta y su respectivo ticket. El sistema se hace cargo de registrar las ventas del producto o servicio de la organización, evalúa sus características o necesidades del negocio, llevando un detalle del registro de todas las transacciones.

La base de datos contiene la tabla inventario en la cual se almacena el id del producto el cual es un tipo de valor serial para identificar el producto, contiene la descripción en el cual se hace una pequeña descripción del producto a almacenar es de tipo varchar, también contiene el precio del producto el cual es de tipo flotante el el cual se almacenara el valor del producto, por ultimo tenemos la cantidad de producto con la que contamos en esta se almacena el total del producto en inventario y es de tipo entero.

Tabla inventario

	Columna	Tipo de dato	Descripción
PK	Id_prod	Serial	Número de identificación para producto
	Descripción	Varchar (20)	Descripción de producto
	precio	Float	Número de precio de producto
	Stock	Integer	Número de stock de producto

En la tabla clientes tenemos el id_clientes el cual es la primary key de tipo serial y sirve para identificar al cliente, también tenemos la variable nombre de tipo varchar que contiene el nombre del cliente, la variable dirección contiene la dirección del cliente y es de tipo varchar, contamos también con la variable numero la cual es de tipo varchar y contiene el numero de teléfono del cliente, la variable correo de tipo varchar con tiene la dirección de correo electrónico del cliente.

Tabla Clientes

	Columna	Tipo de dato	Descripción
PK	Id_clientes	Serial	Número de identificación para cliente
	nombre	Varchar (20)	Nombre completo del cliente
	dirección	Varchar (20)	Dirección completa del cliente
	teléfono	Varchar (12)	Teléfono de contacto
	correo	Varchar (20)	Correo electrónico de contacto
	contraseña	Varchar (20)	Contraseña del cliente

La tabla proveedores contiene el id_prov el cual almacena el id del cliente siento esta la primary key de la tabla, contamos con variable nombre de tipo varchar la cual almacena el nombre de los proveedores, la variable dirección de tipo varchar contiene la dirección de los proveedores, la variable teléfono contiene el numero de teléfono de los proveedores y por ultimo la variable correo de tipo varchar en el cual almacenamos el correo de los proveedores.

Tabla proveedores

	Columna	Tipo de dato	Descripción
PK	Id_prov	Serial	Número de identificación para proveedor
	Nombre	Varchar (20)	Nombre completo del proveedor
	dirección	Varchar (20)	Dirección completa del proveedor
	teléfono	Varchar (12)	Teléfono de contacto
	Correo	Varchar (20)	Correo electrónico de contacto

En la tabla ventas contamos con la variable id_venta la cual es serial y es la primary key de la tabla sirve para identificar la venta, contamos con el id_prod el cual es una llave foránea de la tabla inventario sirve para identificar el producto, tenemos el id_clientes el cual es una llave foránea de la tabla clientes la cual sirve para identificar al cliente, también el id_prov el cual es una llave foránea y sirve para identificar al proveedor, tenemos la cantidad el cual es un tipo de variable integer y sirve para saber la cantidad de producto a vender, la variable precio sirve para saber la cantidad total a pagar y por ultimo la variable pago la cual almacena la cantidad ingresada a pagar .

	Columna	Tipo de dato	Descripción
PK	Id_venta	Serial	Código de identificación para ventas
FK	Id_prod	Integer	Número de identificación para producto
FK	Id_clientes	Integer	Número de identificación para cliente
FK	Id_prov	Integer	Número de identificación para proveedor

Diccionario de datos

Tabla usuarios

	Columna	Tipo de dato	Descripción
	usuario	Varchar (20)	Nombre de usuario
	contraseña	Varchar (20)	Contraseña de usuario
	tipo	Varchar (20)	Descripción de tipo de usuario

Tabla inventario

	Columna	Tipo de dato	Descripción
PK	Id_prod	Serial	Número de identificación para producto
	Descripción	Varchar (20)	Descripción de producto
	precio	Float	Número de precio de producto
	Stock	Integer	Número de stock de producto

Tabla Clientes

	Columna	Tipo de dato	Descripción
PK	Id_clientes	Serial	Número de identificación para cliente
	nombre	Varchar (20)	Nombre completo del cliente
	dirección	Varchar (20)	Dirección completa del cliente
	teléfono	Varchar (12)	Teléfono de contacto
	correo	Varchar (20)	Correo electrónico de contacto
	contraseña	Varchar (20)	Contraseña del cliente

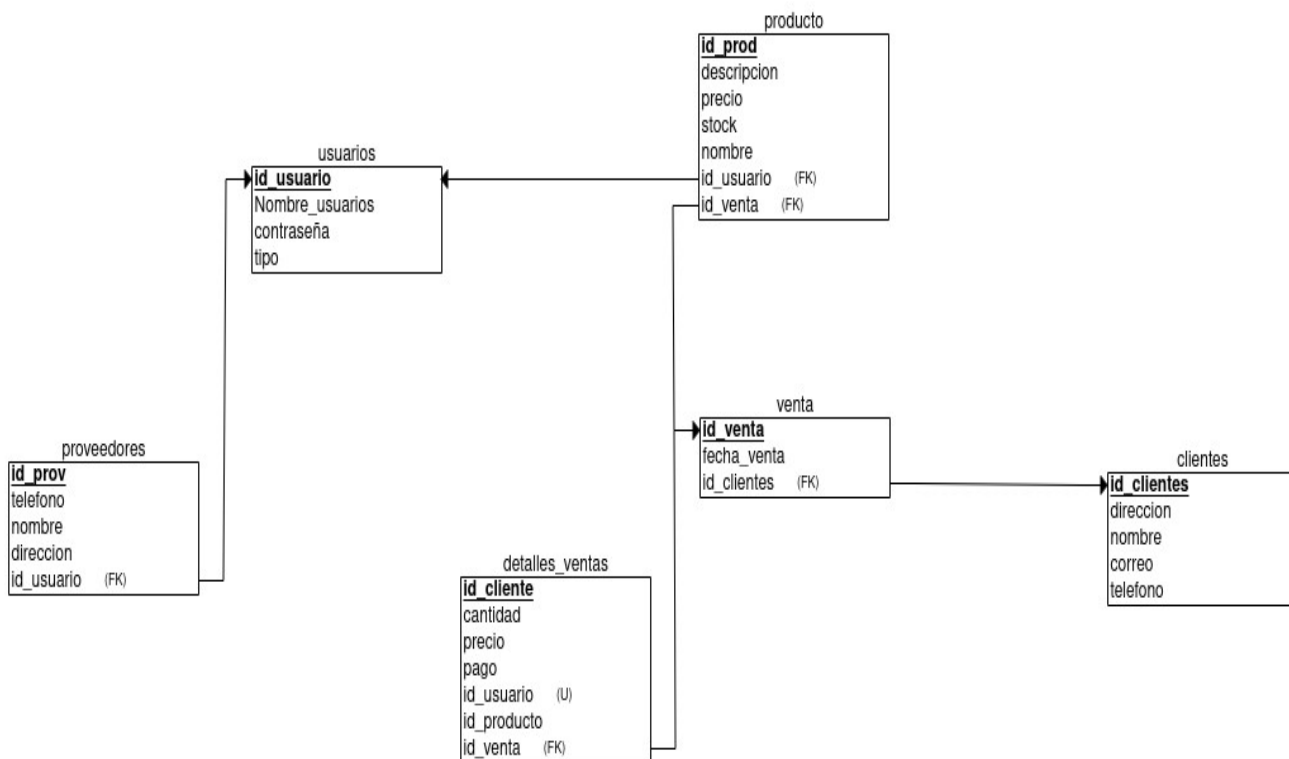
Tabla proveedores

	Columna	Tipo de dato	Descripción
PK	Id_prov	Serial	Número de identificación para proveedor
	Nombre	Varchar (20)	Nombre completo del proveedor
	dirección	Varchar (20)	Dirección completa del proveedor
	teléfono	Varchar (12)	Teléfono de contacto
	Correo	Varchar (20)	Correo electrónico de contacto

Tabla ventas

	Columna	Tipo de dato	Descripción
PK	Id_venta	Serial	Código de identificación para ventas
FK	Id_prod	Integer	Número de identificación para producto
FK	Id_clientes	Integer	Número de identificación para cliente
FK	Id_prov	Integer	Número de identificación para proveedor

Diagrama relacional BD



0.13 Script

- Revocando privilegios del rol 'public' REVOKE ALL ON DATABASE ventas FROM PUBLIC
- Rol solo lectura, en este rol solo esta permitido las consultas que se realizaran en el Programa
- permitiendo solo el acceso a la información de las tablas mas no la modificacion de estas mismas.
- Este rol esta diseñado para los vendedores de una tienda los cuales solo necesitan realizar consultas
- para conocer los datos de los clientes o de algunos productos, por lo cual los permisos de estos mismos son
- minimos para le manejo de la base de datos.

```
CREATE ROLE Vendedor
GRANT CONNECT ON DATABASE venta TO Vendedor
GRANT USAGE ON SCHEMA ventas TO Vendedor
GRANT SELECT ON ALL TABLES IN SCHEMA ventas TO Vendedor
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT SELECT ON TABLES TO Vendedor
```

- Rol lectura-escritura, en este rol se permite la modificación de las tablas y de su contenido asi como la manipulación de la información
- que contiene cada tabla, permitiendo eliminar, modificar, insertar el contenido en las tablas o también eliminar las tablas.
- Este rol esta diseñado para los o el administrador de la tienda el cual por el contrario del vendedor -necesita realizar varias acciones en
- la base de datos como modificar la cantidad de productos en existencia el modificar datos de los -clientes si es necesario o eliminar algunas cosas que sean necesarias.

```
CREATE ROLE Administrador
GRANT CONNECT ON DATABASE venta TO Administrador
GRANT USAGE, CREATE ON SCHEMA ventas TO Administrador
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA ventas TO Administrador
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO Administrador
GRANT USAGE ON ALL SEQUENCES IN SCHEMA Ventas TO Administrador
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT USAGE ON SEQUENCES TO Administrador
```

- Se crean los usuarios con sus respectivas contraseñas

```
CREATE USER Vendedores WITH PASSWORD 'Ceti123'
CREATE USER Administradores WITH PASSWORD 'UNSI123'
```

- Se Concede el permiso correspondiente a cada uno de los usuarios

```
GRANT Vendedor TO vendedores
GRANT Administrador TO Administradores:REVOKE CREATE ON SCHEMA public FROM PUBLIC
```

- Revocando privilegios del rol 'public' REVOKE ALL ON DATABASE ventas FROM PUBLIC
- Rol solo lectura, en este rol solo esta permitido las consultas que se realizaran en el Programa
- permitiendo solo el acceso a la información de las tablas mas no la modificacion de estas mismas.
- Este rol esta diseñado para los vendedores de una tienda los cuales solo necesitan realizar consultas

–para conocer los datos de los clientes o de algunos productos, por lo cual los permisos de estos mismos son

–minimos para le manejo de la base de datos.

```
CREATE ROLE Vendedor
```

```
GRANT CONNECT ON DATABASE venta TO Vendedor
```

```
GRANT USAGE ON SCHEMA ventas TO Vendedor
```

```
GRANT SELECT ON ALL TABLES IN SCHEMA ventas TO Vendedor
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT SELECT ON TABLES TO Vendedor
```

– Rol lectura-escritura, en este rol se permite la modificación de las tablas y de su contenido asi como la manipulación de la información

–que contiene cada tabla, permitiendo eliminar, modificar, insertar el contenido en las tablas o también eliminar las tablas.

–Este rol esta diseñado para los o el administrador de la tienda el cual por el contrario del vendedor –necesita realizar varias acciones en

–la base de datos como modificar la cantidad de productos en existencia el modificar datos de los –clientes si es necesario o eliminar algunas cosas que sean necesarias.

```
CREATE ROLE Administrador
```

```
GRANT CONNECT ON DATABASE venta TO Administrador
```

```
GRANT USAGE, CREATE ON SCHEMA ventas TO Administrador
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA ventas TO Administrador
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO Administrador
```

```
GRANT USAGE ON ALL SEQUENCES IN SCHEMA Ventas TO Administrador
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA ventas GRANT USAGE ON SEQUENCES TO Administrador
```

– Se crean los usuarios con sus respectivas contraseñas

```
CREATE USER Vendedores WITH PASSWORD 'Ceti123'
```

```
CREATE USER Administradores WITH PASSWORD 'UNSiS123'
```

–Se Concede el permiso correspondiente a cada uno de los usuarios

```
GRANT Vendedor TO vendedores
```

```
GRANT Administrador TO Administradores:
```

0.14 Diagrama de Secuencia

Diagramas de Secuencia

1.-Login

@startuml

note right: nancy

actor Administrador

Activate login

Administrador -> login: 1.-Ingresa sus datos

database baseDeDatos #pink

login -> login: 2.-Ingresa su Usuario

login -> login: 3.-Ingresa su Contraseña

login -> baseDeDatos: 4.- Validar los datos

baseDeDatos --> Administrador: 5.- El administrador entra al sistema

@enduml

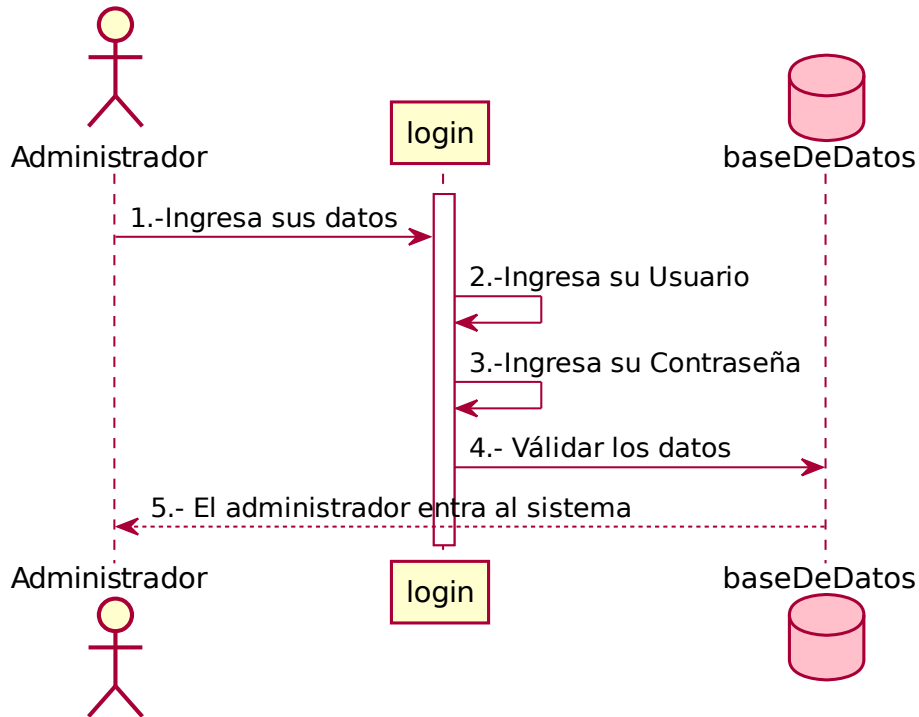


Figura 1: Diagrama de secuencia de login

2.- Inventario

@startuml

actor Administrador

Administrador->Botón:1.-Buscar producto

Activate Botón

Botón->Datos:2.-Nombre del producto

deactivate Botón

database baseDeDatos #pink

Activate Datos

Datos->baseDeDatos :3.-Verifica la búsqueda

deactivate Datos

Activate baseDeDatos

Administrador<--baseDeDatos:4.-Producto encontrado

deactivate baseDeDatos

Administrador->Botón:5.-Agregar producto

Botón->Datos:6.-Datos del producto

Datos->baseDeDatos :7.-Verifica datos del producto

Administrador<--baseDeDatos:8.-Producto Agregado exitosamente

Administrador->Botón:9.- Modificar producto

Botón->Datos:10.- Renombrar datos del producto

Datos->Datos:11.-se checan que los datos sean correctos

Datos->baseDeDatos :12.-Guarda datos del producto

Administrador<--baseDeDatos:13.-Producto Modificado exitosamente

Administrador->Botón:14.- Eliminar producto

Botón->Datos:15.- Escoger producto a eliminar

Datos->baseDeDatos:16.- se Confirma que se quiere eliminar

Administrador<--baseDeDatos:7.-Producto Eliminado exitosamente

@enduml

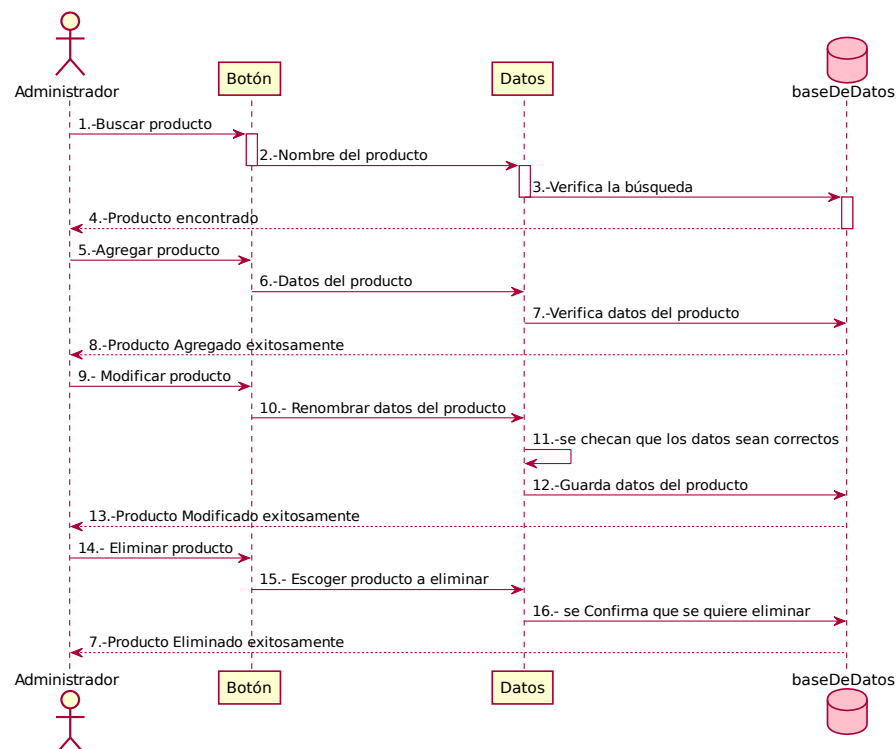


Figura 2: Inventario-secuencia

3.-Registrar Usuario

@startuml

actor Usuarios

Activate Datos

database baseDeDatos

Usuarios->Datos: 1.-Nuevo Usuario

Datos->Datos:2.-Nombre Usuario

Datos->Datos:3.-Nombre real

Datos->Datos:4.-Contraseña

Datos->Datos:5.- Rol

Datos->baseDeDatos:6.-verificar datos

activate baseDeDatos

baseDeDatos-->Usuarios:7.- Se registro su usuario correctamente

deactivate baseDeDatos

@enduml

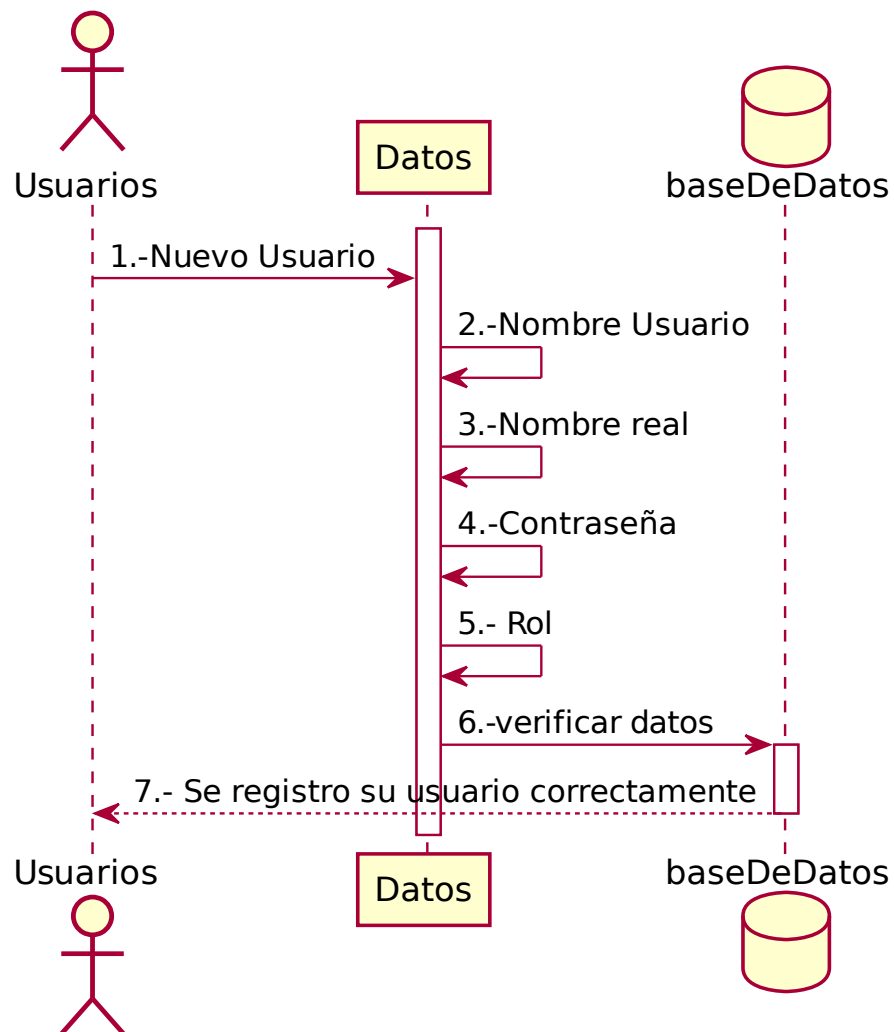


Figura 3: Secuencia-Usuarios

4.-Registrar Proveedores

@startuml

actor Administrador

Activate Datos

database baseDeDatos

Administrador->>Datos: 1.-Nuevo proveedor

Datos->>Datos:2.-Nombre proveedor

Datos->>Datos:3.-Código de proveedor

Datos->>Datos:4.-Teléfono

Datos->>Datos:5.- Dirección

Datos->>baseDeDatos:6.-verificar datos

activate baseDeDatos

baseDeDatos-->>Administrador:7.- Se registro un nuevo proveedor correctamente

deactivate baseDeDatos

@enduml

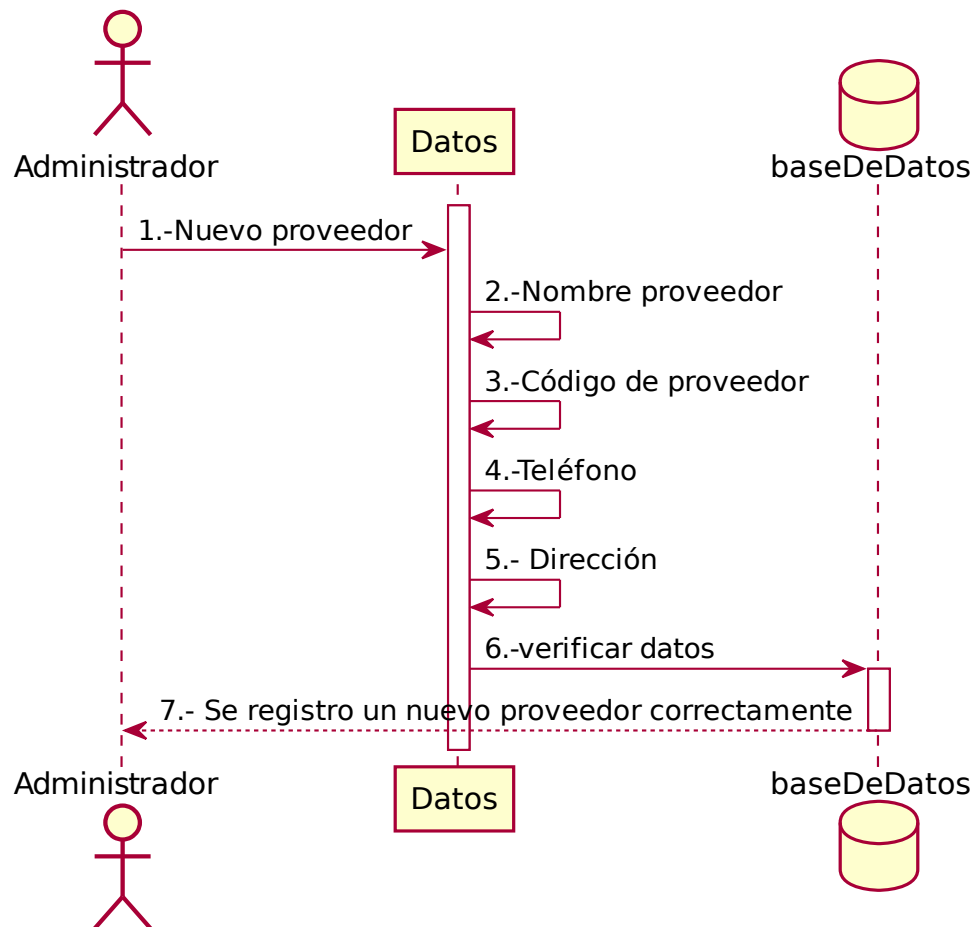


Figura 4: Secuencia-Proveedor

5.-Registro de Clientes

@startuml

actor Administrador

Activate Datos

database baseDeDatos #pink

Administrador->>Datos: 1.-Nuevo Cliente

Datos->>Datos:2.-Nombre Cliente

Datos->>Datos:3.-Código de Cliente

Datos->>Datos:4.-Teléfono

Datos->>Datos:5.- Dirección

Datos->>baseDeDatos:6.-verificar datos

activate baseDeDatos

baseDeDatos-->>Administrador:7.- Se registro un nuevo Cliente correctamente

deactivate baseDeDatos

@enduml

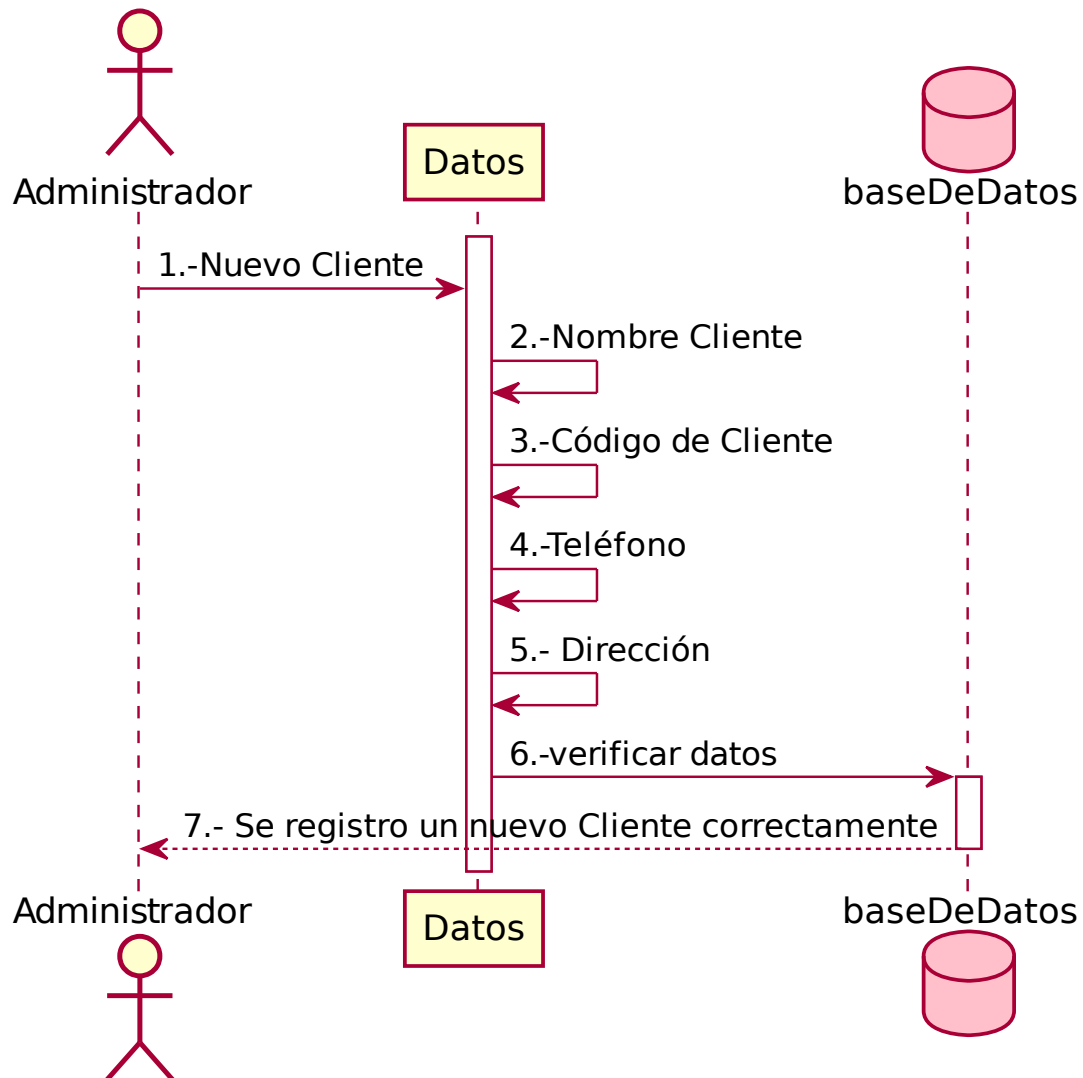


Figura 5: Secuencia-Clientes

6.-Información de la Tienda

```
@startuml
actor Administrador
Activate Datos
database baseDeDatos #pink
Administrador->>Datos: 1.-Edita la informacion
Datos->>Datos:2.-Nombre empresa
Datos->>Datos:3.-Código de empresa
Datos->>Datos:4.-Teléfono
Datos->>Datos:5.- Dirección
Datos->>baseDeDatos:6.-verificar datos
activate baseDeDatos
baseDeDatos-->público:7.- Se muestra la información de la empresa
deactivate baseDeDatos
@enduml
```

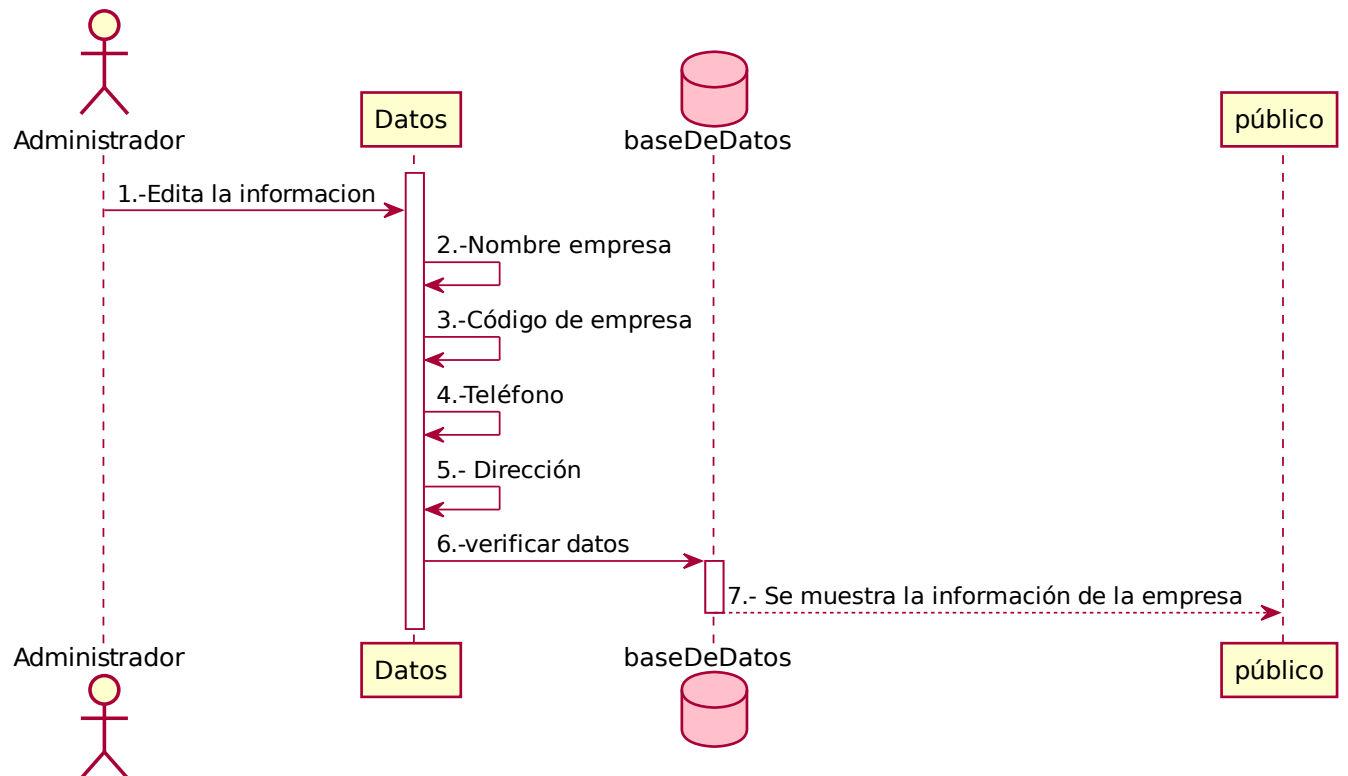


Figura 6: Secuencia-Iempresa

7.-Consulta Ventas

@startuml

actor Vendedor

database baseDeDatos #pink

Vendedor->baseDeDatos :1.-Búscas las ventas realizadas

Activate baseDeDatos

baseDeDatos → baseDeDatos:2.-Verifica por fechas

deactivate baseDeDatos

baseDeDatos-->3.-Vendedor:Devuelve el reporte de las ventas

@enduml

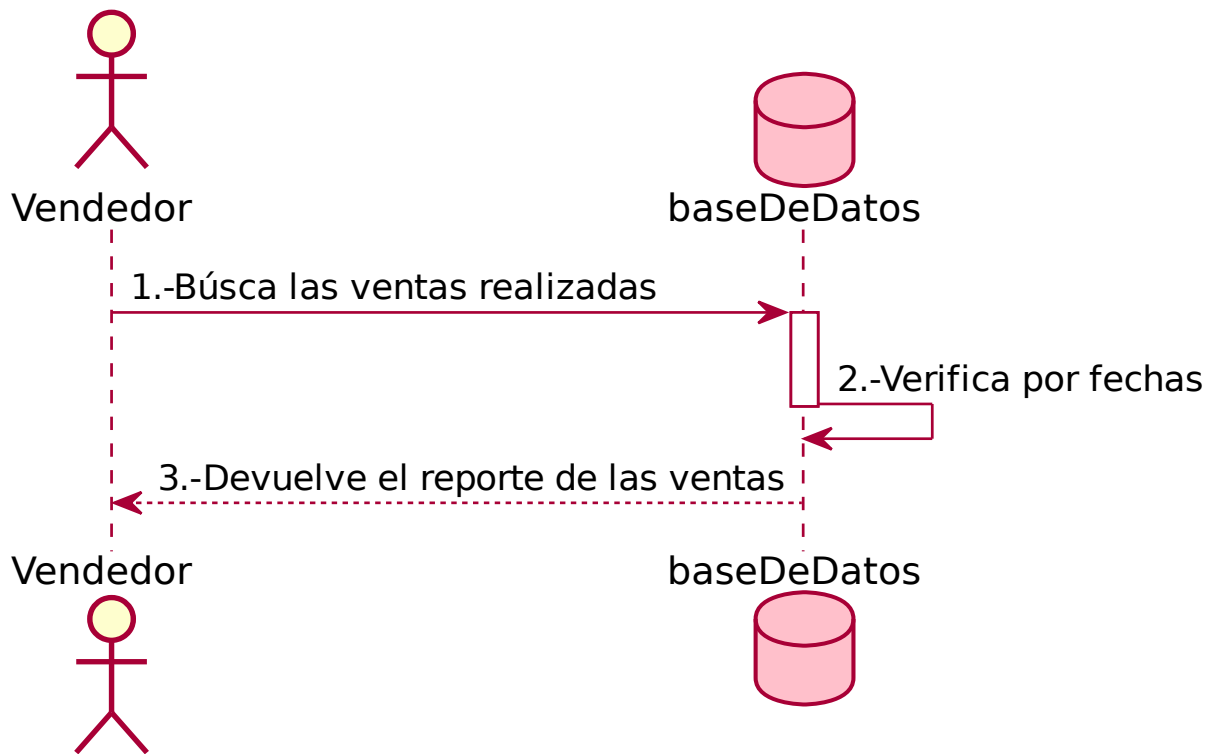


Figura 7: Secuencia-ConsultaVentas

0.15 Diagrama de Despliegue

Diagrama de Despliegue

```
@startuml
node Iniciodesesión as i
node Principal as p
node Ventas as v
node Clientes as c
node Proveedores as pv
node Usuarios as u
node Producto as pd
node JDBC as j
node BaseDeDatos as bd
i-->p
p-->v
p-->c
p-->pd
p-->pv
p-->u
v-->j
c-->j
pv-->j
u-->j
pd-->j
j-->bd
@enduml
```

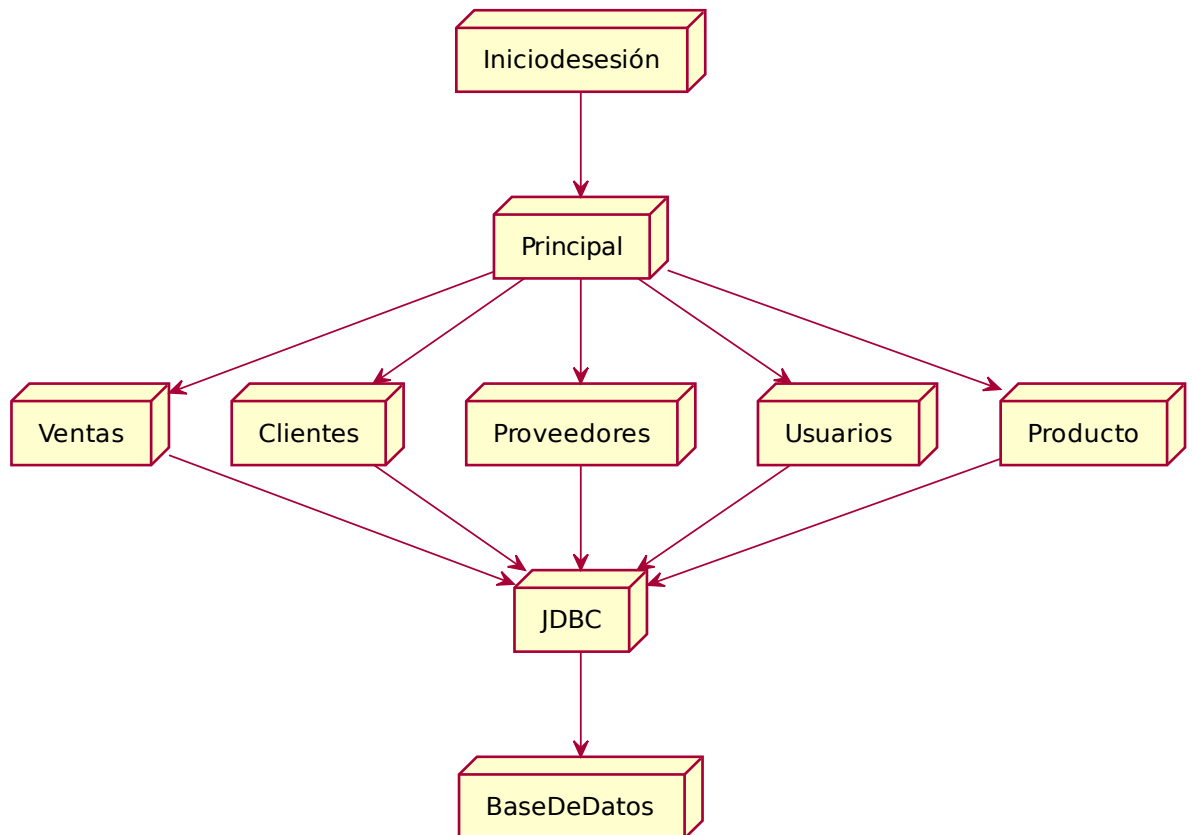


Figura 1: Diagrama de Despliegue

0.16 Diagrama de Componentes

Diagrama de Componentes

```

@startuml
package Diseño <<Interface>> {
    component Principal
    component Ventas
    component Inventario
    component Clientes
    component Información
    component Principal
}
package Operaciones <<componentst>> {
    component altaProductos
    component bajaProductos
    component altaUsuarios
    component bajaUsuarios
    component altaClientes
    component bajaclientes
    component altaProveedores
    component bajacProveedores
}
package Lógico{
    component GestiónProductos
    component GestiónLogin
    entity ingreso
    component GestiónVentas
}

database postgres #pink{
    component SistemaNikky
}

GestiónLogin-->Principal
GestiónLogin-->SistemaNikky
Operaciones-->SistemaNikky
    
```

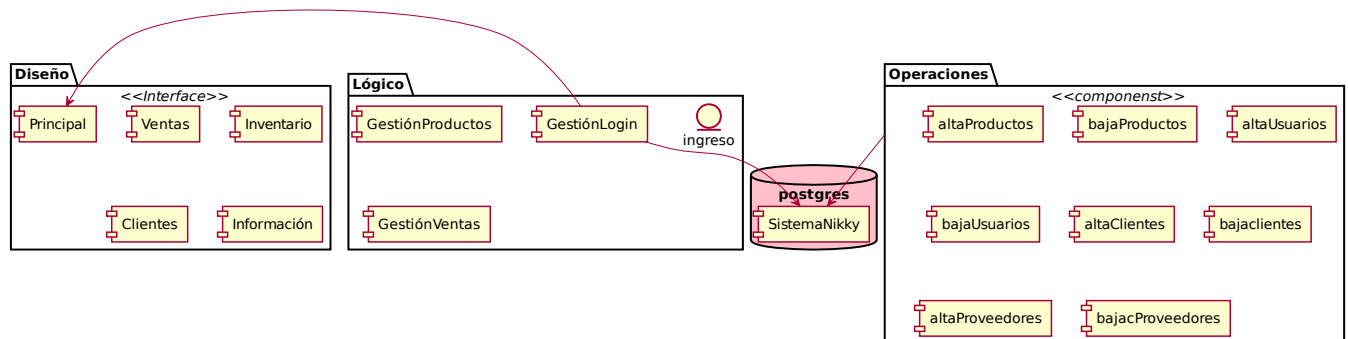


Figura 1: Diagrama de Componentes

Diagrama de Flujo de Datos

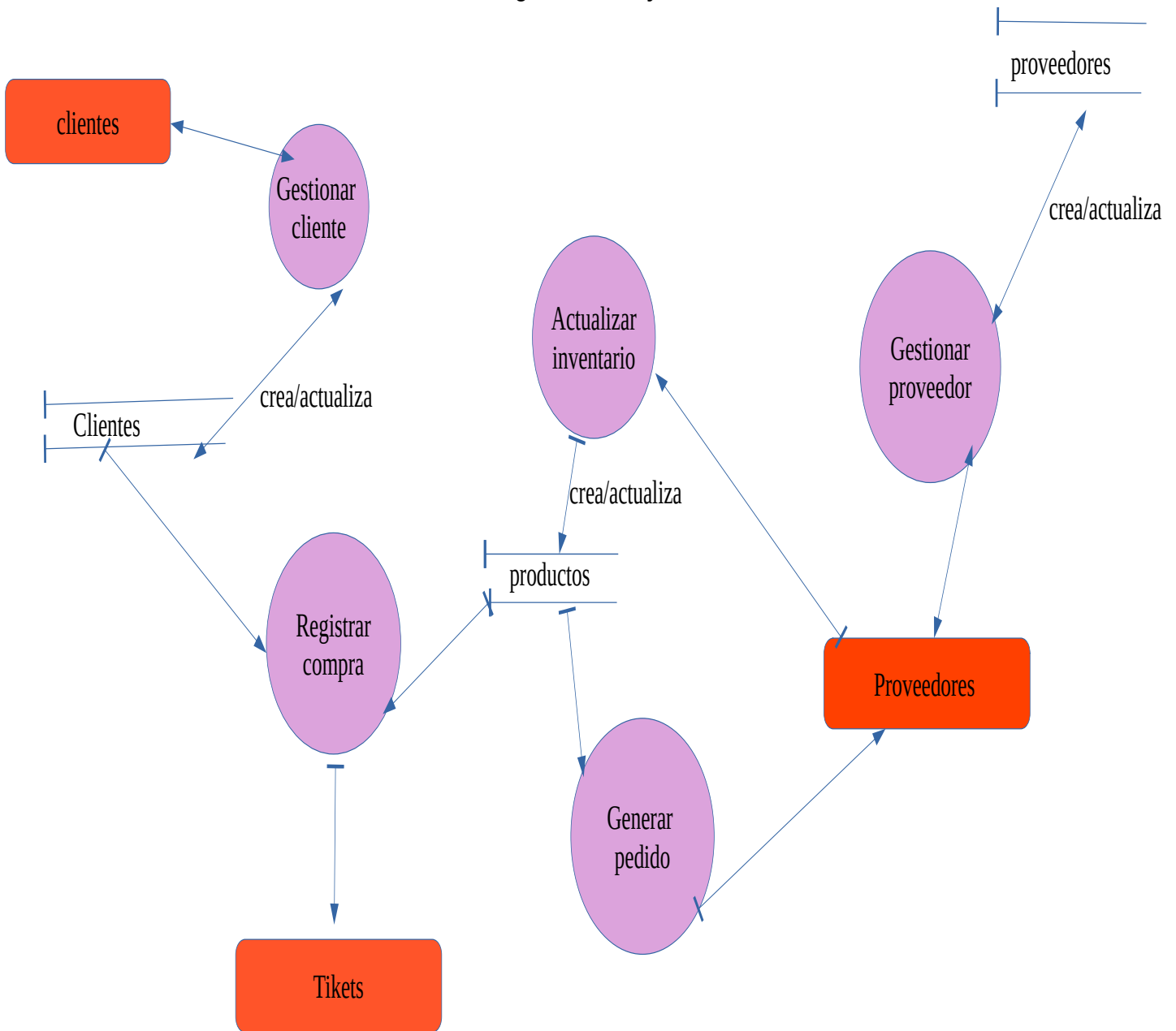
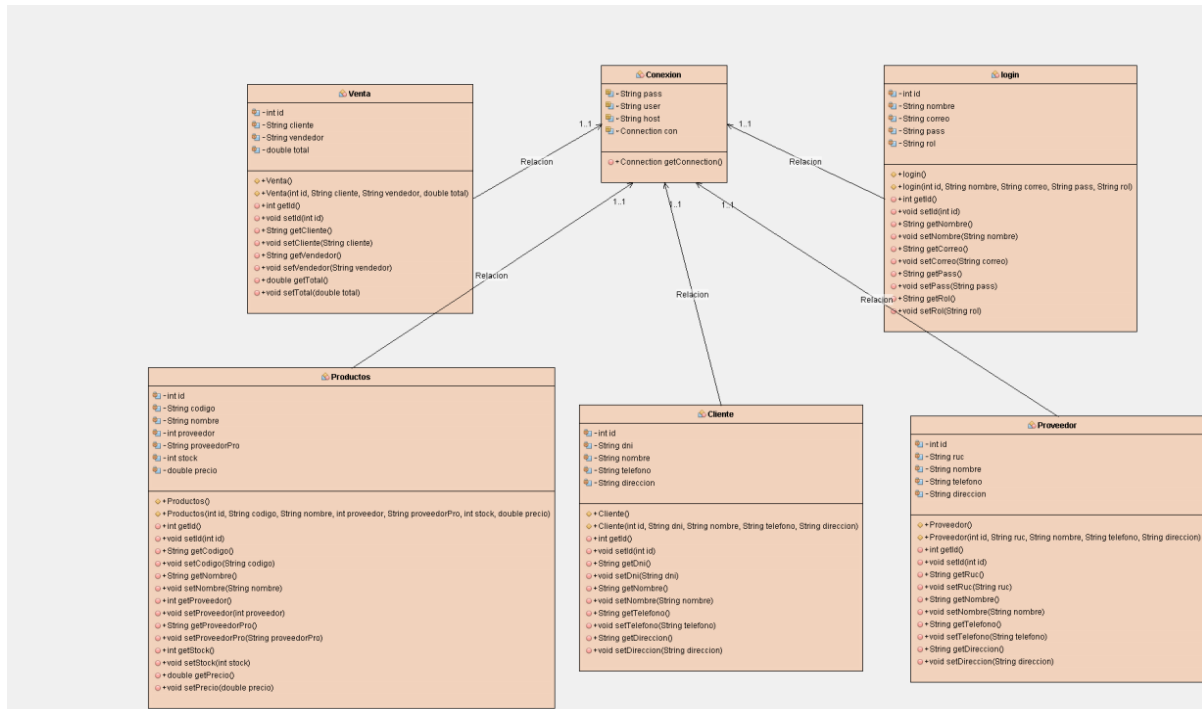
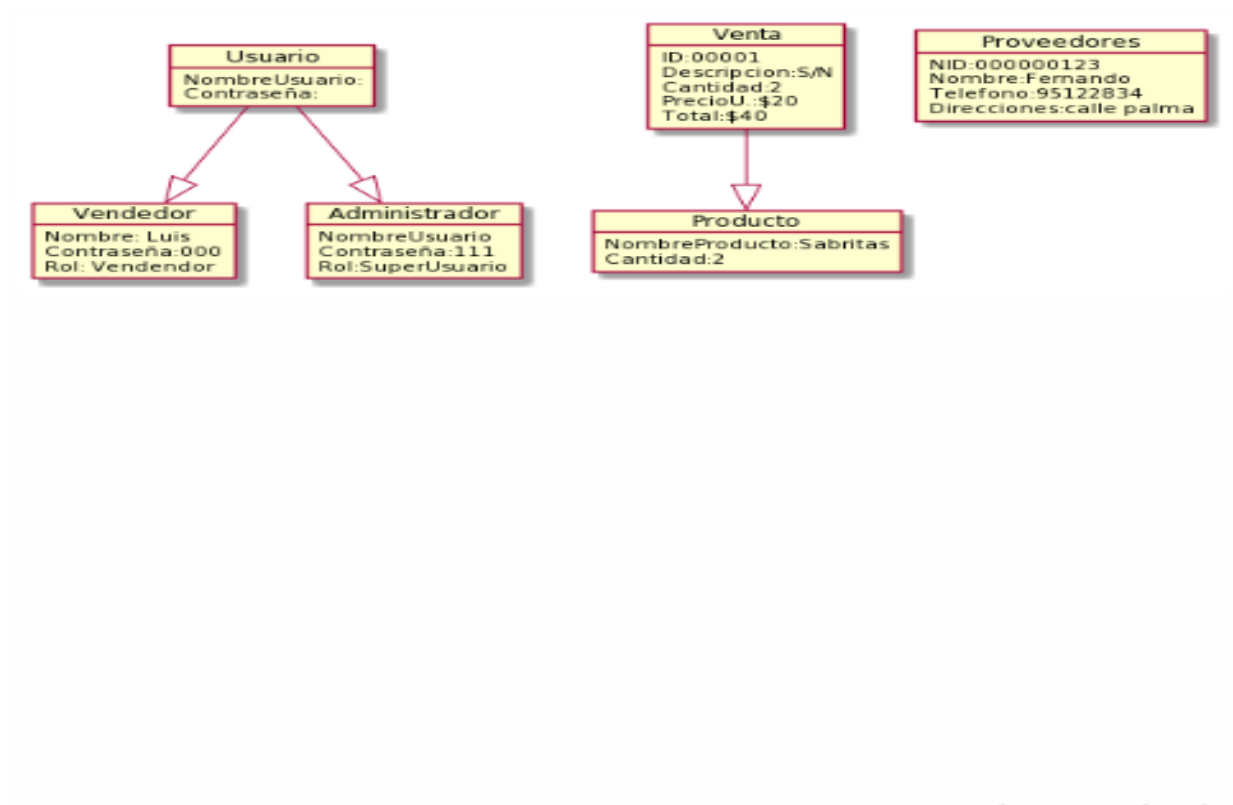


Diagrama de clases



La siguiente imagen hace referencia al diagrama de clases

Diagrama de objetos



La siguiente imagen hace referencia al diagrama de objetos

Codigo UML Diagrama de objetos

```
@startuml
object Usuario
Usuario : NombreUsuario:
Usuario : Contraseña:
object Vendedor
Vendedor : Nombre: Luis
Vendedor : Contraseña:000
Vendedor : Rol: Vendedor
object Administrador
Administrador : NombreUsuario
Administrador : Contraseña:111
Administrador : Rol:SuperUsuario

object Venta
Venta : ID:00001
Venta : Descripcion:S/N
Venta : Cantidad:2
Venta : PrecioU.:$20
Venta : Total:$40

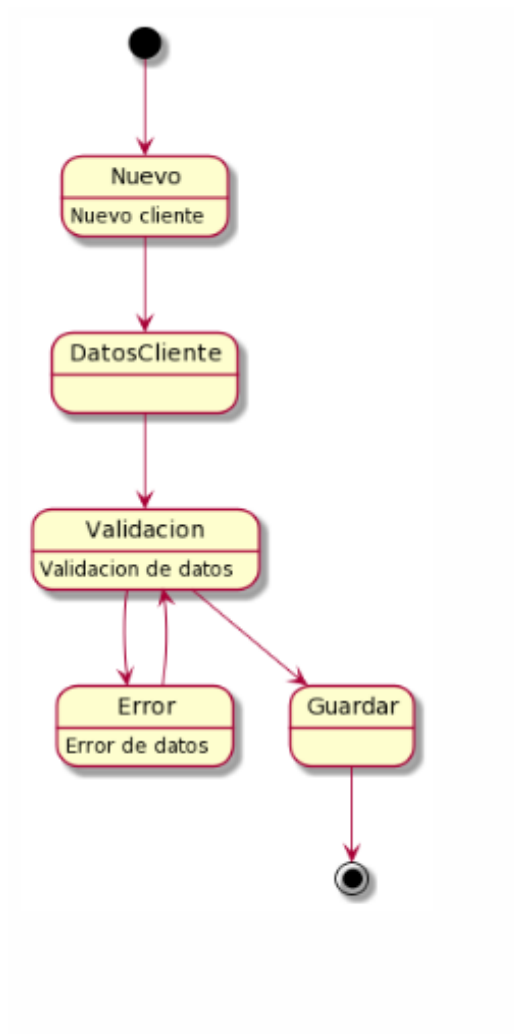
object Producto
Producto : NombreProducto:Sabritas
Producto : Cantidad:2

object Proveedores
Proveedores : NID:000000123
Proveedores : Nombre:Fernando
Proveedores : Telefono:95122834
Proveedores : Direcciones:calle palma

Venta--|>Producto
Usuario--|> Vendedor
Usuario --|> Administrador
@enduml
```

La siguiente imagen hace referencia al código UML diagrama de objetos

Diagrama de estados



La siguiente imagen hace referencia a diagrama Registro clientes

Diagrama de estados

```
@startuml
[*] --> Nuevo
Nuevo : Nuevo cliente
Nuevo--> DatosCliente

DatosCliente--> Validacion
Validacion: Validacion de datos
Validacion-->Error
Error: Error de datos
Error-->Validacion
Validacion-->Guardar
Guardar --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Registro cliente

Diagrama de estados



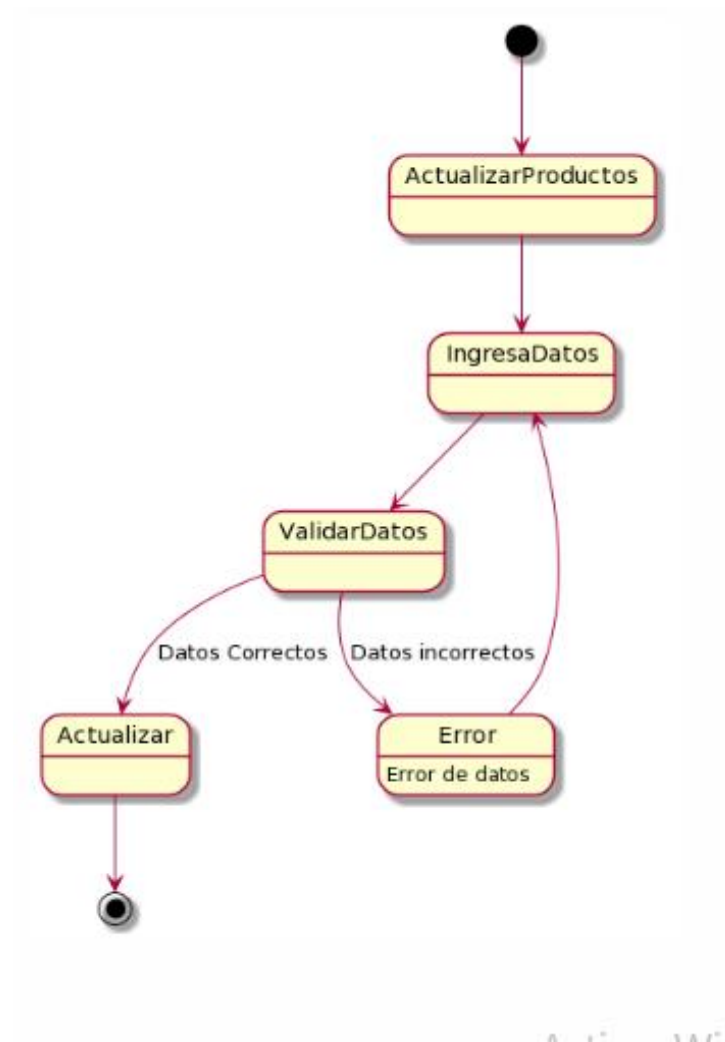
La siguiente imagen hace referencia a diagrama Registro Proveedores

Diagrama de estados

```
@startuml
[*] --> Nuevo
Nuevo : Nuevo Proveedor
Nuevo--> DatosProveedor
DatosProveedor--> Validacion
Validacion: Validacion de datos
Validacion-->Error
Error: Error de datos
Error-->Validacion
Validacion-->Guardar
Guardar --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Registro Proveedores

Diagrama de estados



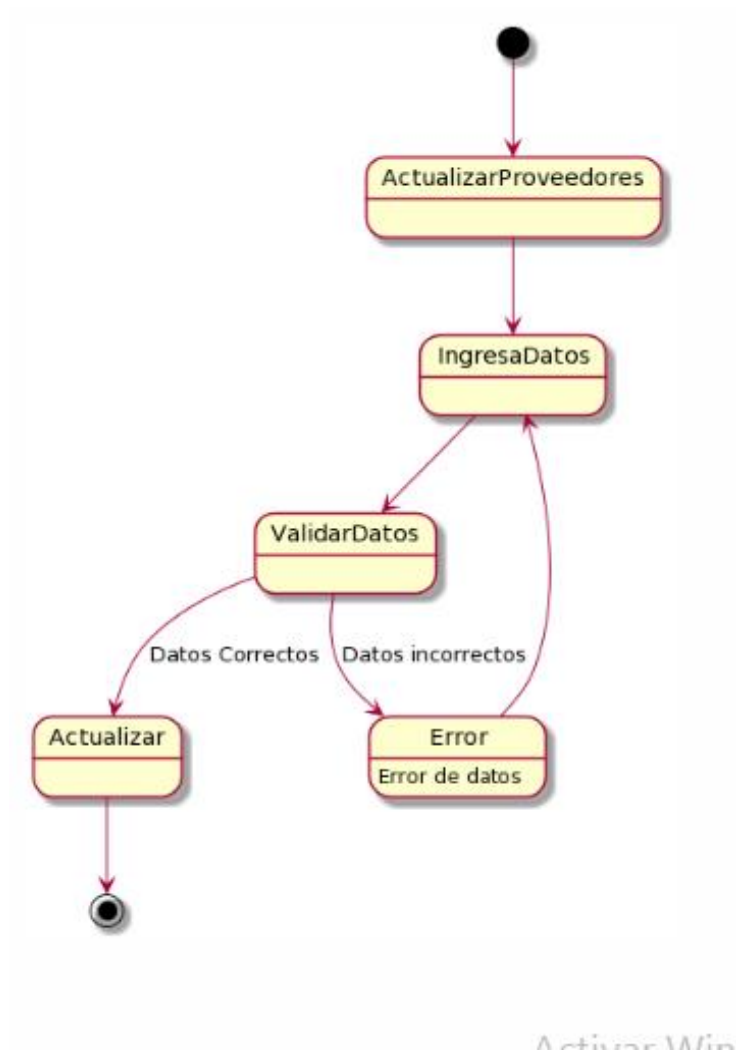
La siguiente imagen hace referencia a diagrama Actualizar Productos

Diagrama de estados

```
@startuml
[*] --> ActualizarProductos
ActualizarProductos--> IngresaDatos
IngresaDatos--> ValidarDatos
ValidarDatos--> Actualizar: Datos Correctos
ValidarDatos--> Error: Datos incorrectos
Error: Error de datos
Error--> IngresaDatos
Actualizar --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Actualizar Productos

Diagrama de estados



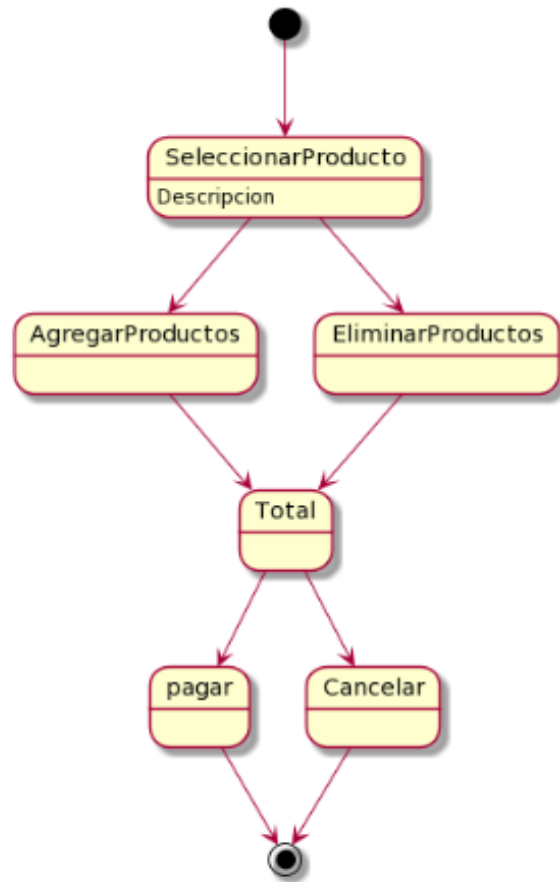
La siguiente imagen hace referencia a diagrama Actualizar Proveedores

Diagrama de estados

```
@startuml
[*] --> ActualizarProveedores
ActualizarProveedores--> IngresaDatos
IngresaDatos--> ValidarDatos
ValidarDatos--> Actualizar: Datos Correctos
ValidarDatos--> Error: Datos incorrectos
Error: Error de datos
Error--> IngresaDatos
Actualizar --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Actualiza Proveedores

Diagrama de estados



Activar

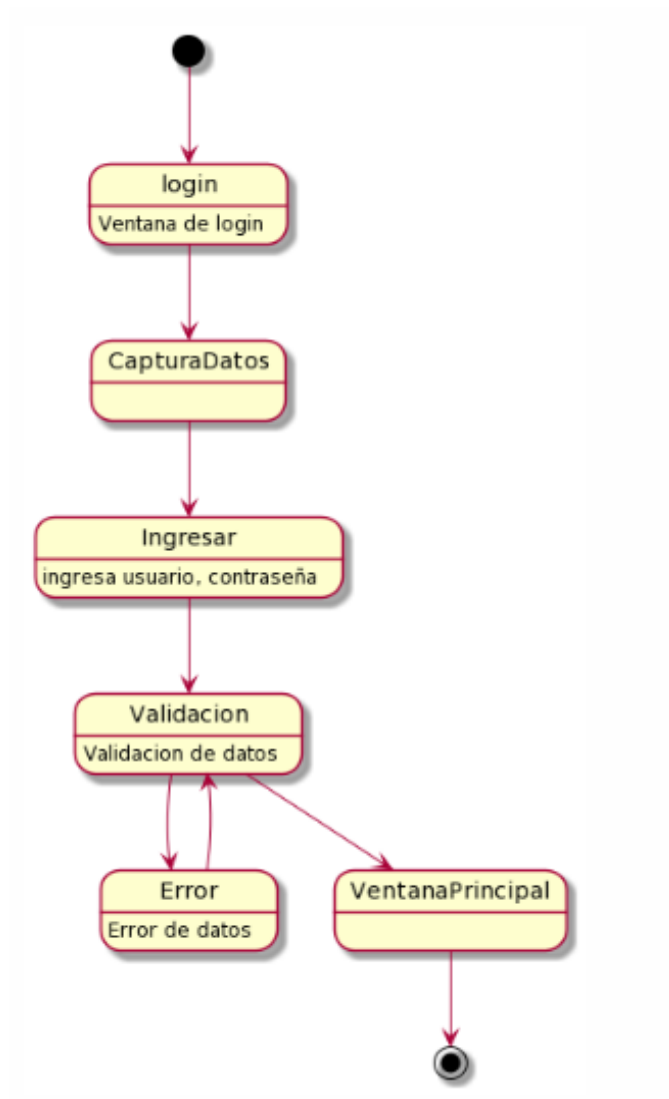
La siguiente imagen hace referencia a diagrama Ventas

Diagrama de estados

```
@startuml
[*] --> SeleccionarProducto
SeleccionarProducto: Descripcion
SeleccionarProducto--> AgregarProductos
SeleccionarProducto--> EliminarProductos
AgregarProductos--> Total
EliminarProductos-->Total
Total -->pagar
Total-->Cancelar
pagar --> [*]
Cancelar --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Ventas

Diagrama de estados



La siguiente imagen hace referencia a diagrama de Login

Diagrama de estados

```
@startuml
[*] --> login
login : Ventana de login
login--> CapturaDatos
CapturaDatos-->Ingresar
Ingresar : ingresa usuario, contraseña
Ingresar--> Validacion
Validacion: Validacion de datos
Validacion-->Error
Error: Error de datos
Error-->Validacion
Validacion-->VentanaPrincipal
VentanaPrincipal --> [*]
@enduml
```

La siguiente imagen hace referencia al código UML Login