# Resultado de imagen para logo espol

DISEÑO DE SOFTWARE

PROYECTO FINAL

PARALELO #1

GRUPO #2

Integrantes:

* Concha Regatto Axell
* Villamar Saltos Luis
* Coello Kondratova Alan

SISTEMA DE INVENTARIO, COMPRA Y VENTA

TecnoImport S.A.

**Descripción de la Propuesta**

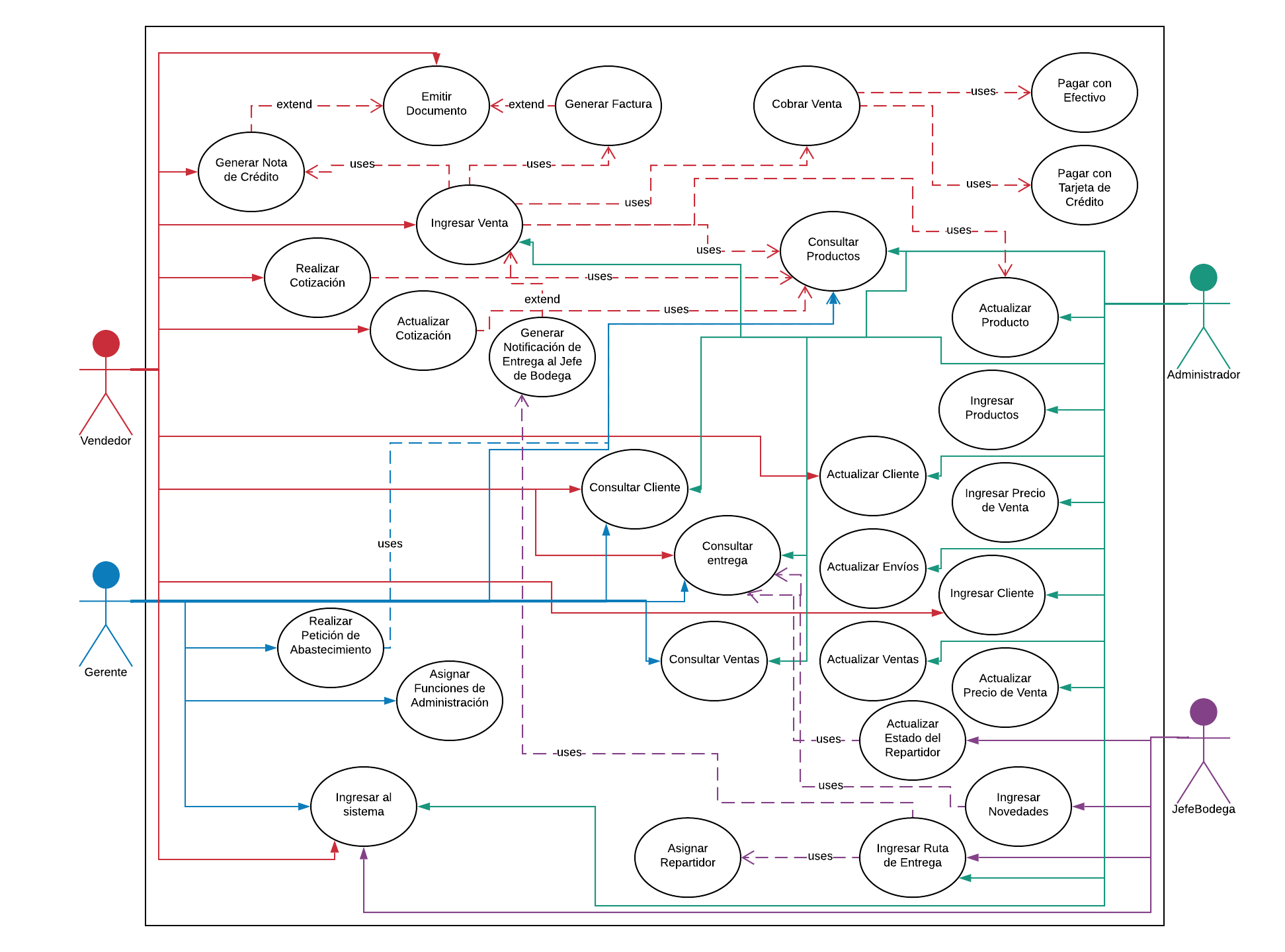
Se propone un sistema que permite el acceso a los datos de compras, ventas e inventario de la empresa TecnoImport S.A.

Al sistema se accederá por medio de un Login general.

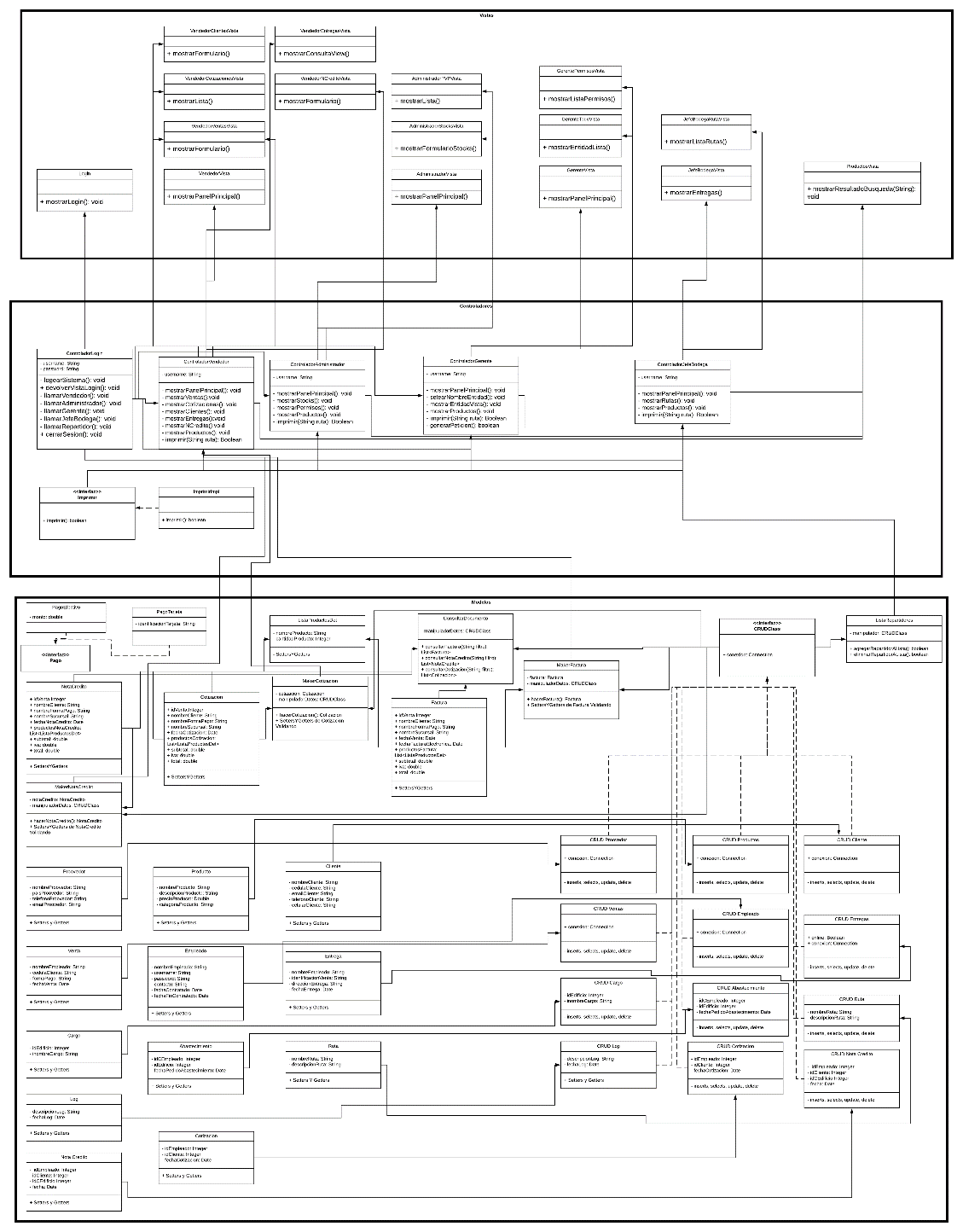
Además de esto, el sistema permite a los distintos empleados de la empresa acceder a distintos aspectos del mismo según su nivel de acceso.

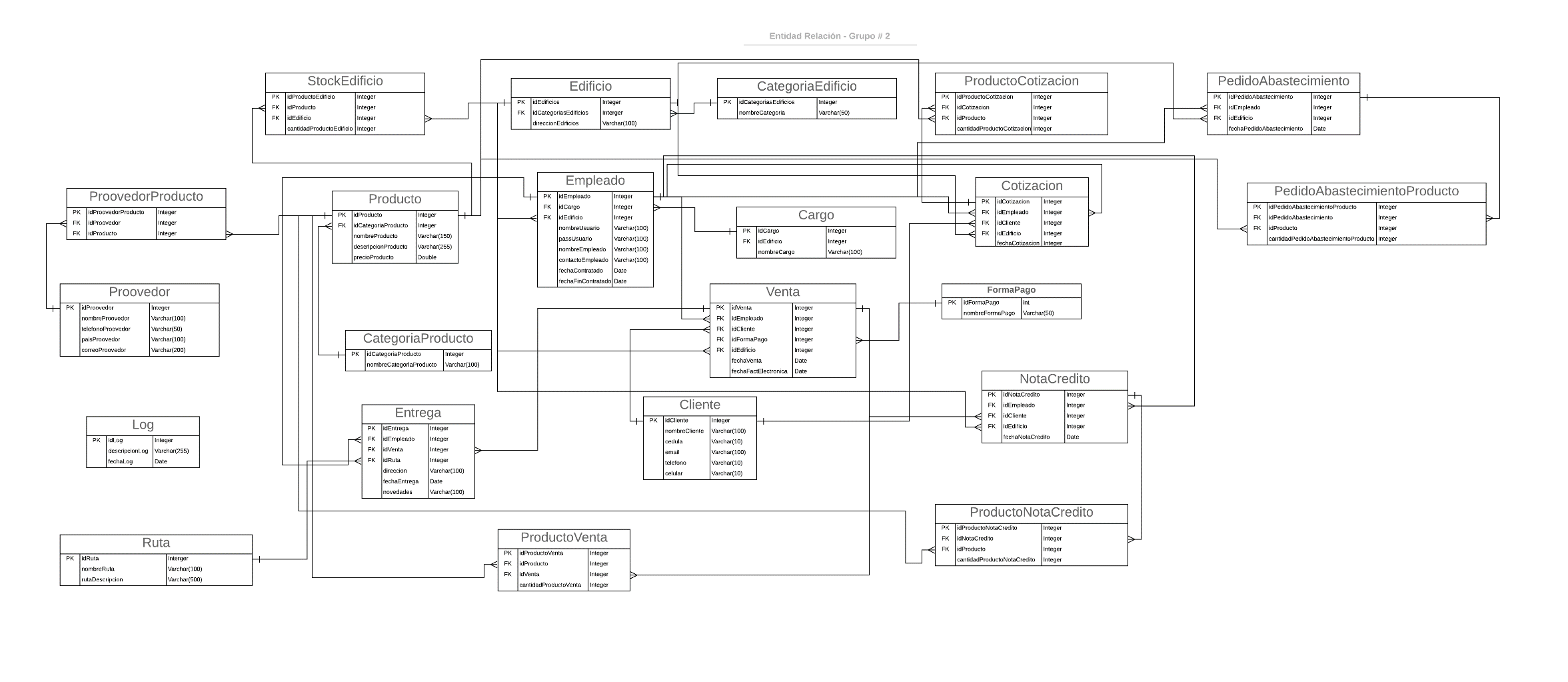
Se podrá realizar búsquedas de productos, realizar cotizaciones de los mismos, cancelar ventas mediante efectivo o tarjeta de crédito, emitir facturas y notas de crédito electrónicas, actualizar stocks y precios de venta, realizar peticiones de abastecimiento de uno o varios artículos.

**Diagrama de Casos de Usos**



**Diagrama de Clases**



**Diagrama normalizado del modelo físico de la base de datos**

**Documentación Caso de Uso**

|  |  |
| --- | --- |
| **Caso de Uso** |  |
| **ID** | **CDU-A01** |
| **Titulo** | Ingresar venta |
| **Autor** | Alan Coello |
| **Actor principal** | Vendedor |
| **Actor secundario** | - |
| **Usa** | Consultar producto, actualizar producto, cobrar venta |
| **Extiende** | - |
| **Descripción** | Al realizar una venta el vendedor la ingresa al sistema. |
| **Precondición** | El producto que se desea vender se encuentra disponible |
| **Secuencia normal** | 1. El vendedor accede al sistema 2. El vendedor ingresa los datos del producto que se desea vender 3. En caso de existir, el producto es mostrado y la venta puede ser realizada 4. Se actualiza la información del producto (cantidad disponible) |
| **Secuencia alternativa** | 2.1 En caso de no disponer del producto, el vendedor recibe este mensaje y procede a informar al cliente esto.  2.2 El cliente puede elegir otro producto  2.3 El vendedor ingresa esta venta al sistema |
| **Postcondición** | La venta es realizada |

|  |  |
| --- | --- |
| **Caso de Uso** |  |
| **ID** | **CDU-A02** |
| **Titulo** | Realizar cotización |
| **Autor** | Alan Coello |
| **Actor principal** | Vendedor |
| **Actor secundario** | - |
| **Usa** | Consultar productos |
| **Extiende** | - |
| **Descripción** | A través de la consulta del producto especifico el vendedor estima el precio del producto |
| **Precondición** | El producto existe o está registrado |
| **Secuencia normal** | 1. El vendedor inicia sesión 2. El vendedor ingresa los datos del producto en la sección de búsqueda 3. Si el producto existe, toda la información del producto es mostrada en pantalla y el vendedor puede realizar la cotización. |
| **Secuencia alternativa** | 2.1 Al buscar el producto este no se encuentra registrado  2.2 Se le da la opción al vendedor de buscar otro producto o cancelar la búsqueda. |
| **Postcondición** | La información del producto es mostrada y el vendedor realiza su cotización correcta |

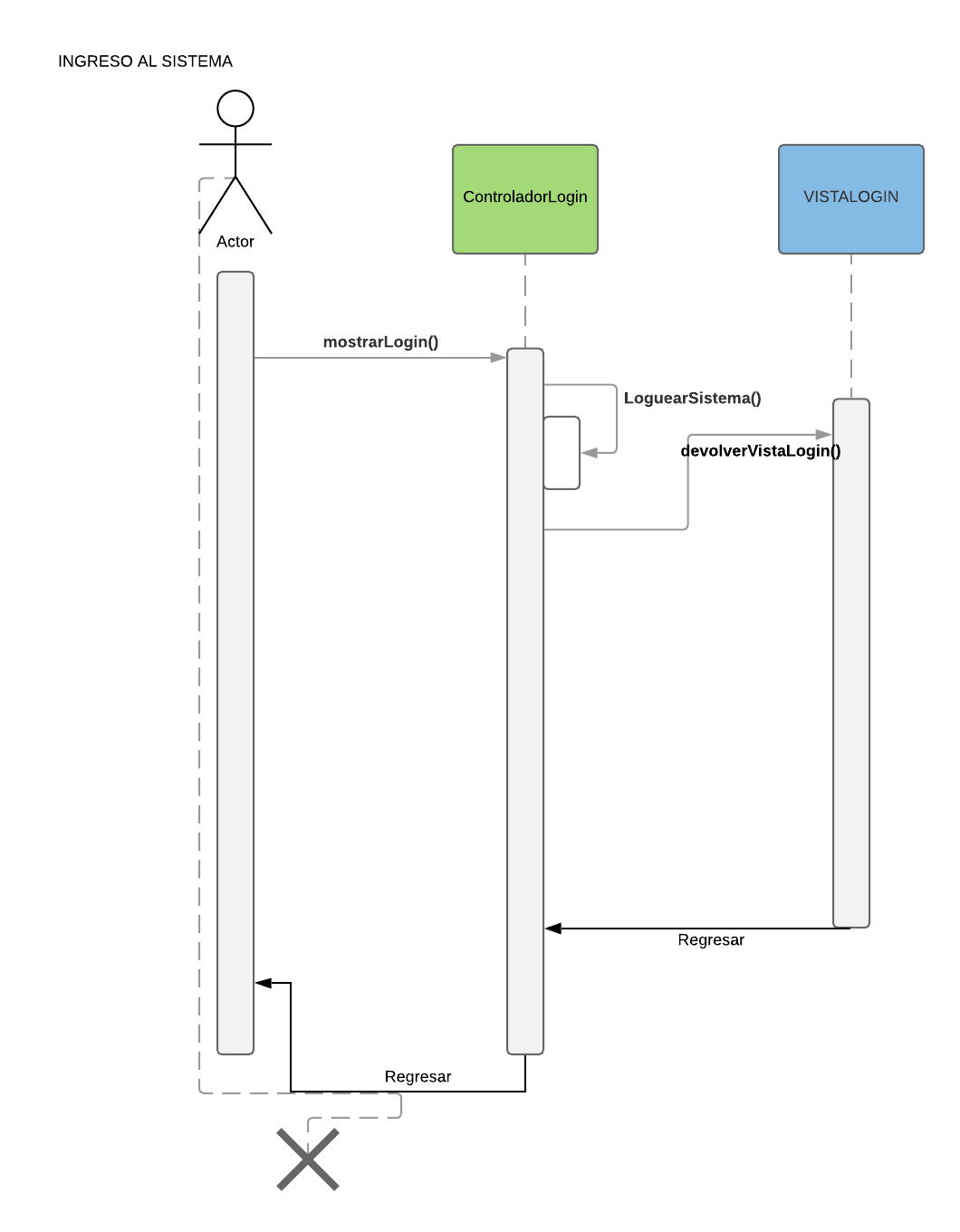
|  |  |
| --- | --- |
| **Caso de Uso** |  |
| **ID** | **CDU-A03** |
| **Titulo** | Emitir documento |
| **Autor** | Alan Coello |
| **Actor principal** | Vendedor |
| **Actor secundario** | - |
| **Usa** | - |
| **Extiende** | - |
| **Descripción** | El vendedor emite un documento, ya sea nota de crédito, factura o algún documento futuro de manera electrónica al cliente |
| **Precondición** | Se realizo una venta y se generó el documento respectivo |
| **Secuencia normal** | 1. El vendedor accede al sistema 2. Realiza una venta 3. El cliente decide recibir factura o el documento pertinente 4. El vendedor genera este documento 5. El vendedor ingresa la dirección a la cual se emitirá el documento 6. En caso de existir esta dirección, el documento es emitido |
| **Secuencia alternativa** | 2.1 En caso de ser incorrecta la dirección ingresada, el cliente indica al vendedor la dirección correcta  2.2 El vendedor ingresa la dirección corregida  2.3 El documento es emitido a la dirección deseada |
|  |  |
| **Postcondición** | El documento fue recibido por el cliente |

|  |  |
| --- | --- |
| **Caso de Uso** |  |
| **ID** | **CDU-A04** |
| **Titulo** | Realizar petición de abastecimiento |
| **Autor** | Alan Coello |
| **Actor principal** | Gerente |
| **Actor secundario** | Jefe de bodega |
| **Usa** | Consultar producto |
| **Extiende** | - |
| **Descripción** | El gerente consulta un producto y decide que se necesita reabastecer el stock local |
| **Precondición** | El producto esta registrado, y existe en la bodega |
| **Secuencia normal** | 1. El gerente ingresa al sistema 2. Realiza una consulta de producto 3. Selecciona el producto que necesita reabastecimiento 4. Realiza la petición de abastecimiento con la cantidad de unidades necesarias |
| **Secuencia alternativa** | 2.1 En caso de no haber disponibilidad de unidades en la bodega se cancela la solicitud de abastecimiento |
| **Postcondición** | La solicitud es enviada a la bodega |

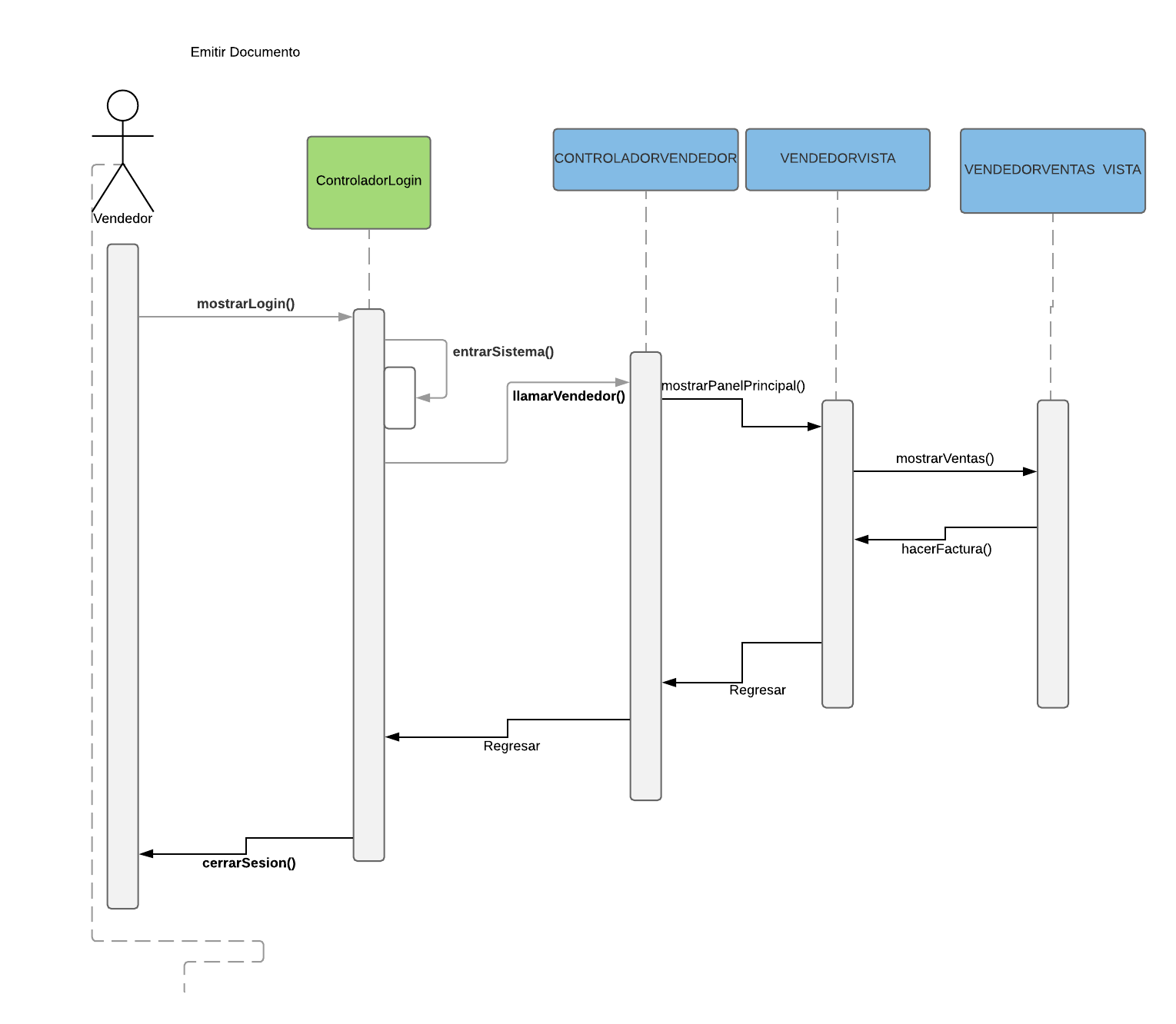
|  |  |
| --- | --- |
| **Caso de Uso** |  |
| **ID** | **CDU-A05** |
| **Titulo** | Ingresar al sistema |
| **Autor** | Alan Coello |
| **Actor principal** | Vendedor, Gerente, Jefe de bodega, administrador |
| **Actor secundario** | - |
| **Usa** | - |
| **Extiende** | - |
| **Descripción** | Se validan los datos ingresados para iniciar sesión en el sistema |
| **Precondición** | El usuario esta registrado en el sistema |
| **Secuencia normal** | 1. El actor accede al dispositivo desde el cual ingresara al sistema 2. Ingresa su nombre de usuario 3. Ingresa su contraseña 4. El sistema valida si los datos ingresados son correctos 5. El sistema valida el rol que el usuario posee y desbloquea las funciones disponibles para este rol 6. Se realiza el ingreso al sistema |
| **Secuencia alternativa** | 2.1 Los datos no son correctos  2.2 El sistema requiere que se ingresen datos que sean correctos  2.3 En caso de no tenerse los datos necesarios, el usuario puede referirse a un sistema de recuperación de datos |
| **Postcondición** | El inicio de sesión en el sistema es exitoso y las funcionalidades son mostradas |

**Diagrama de Secuencias**

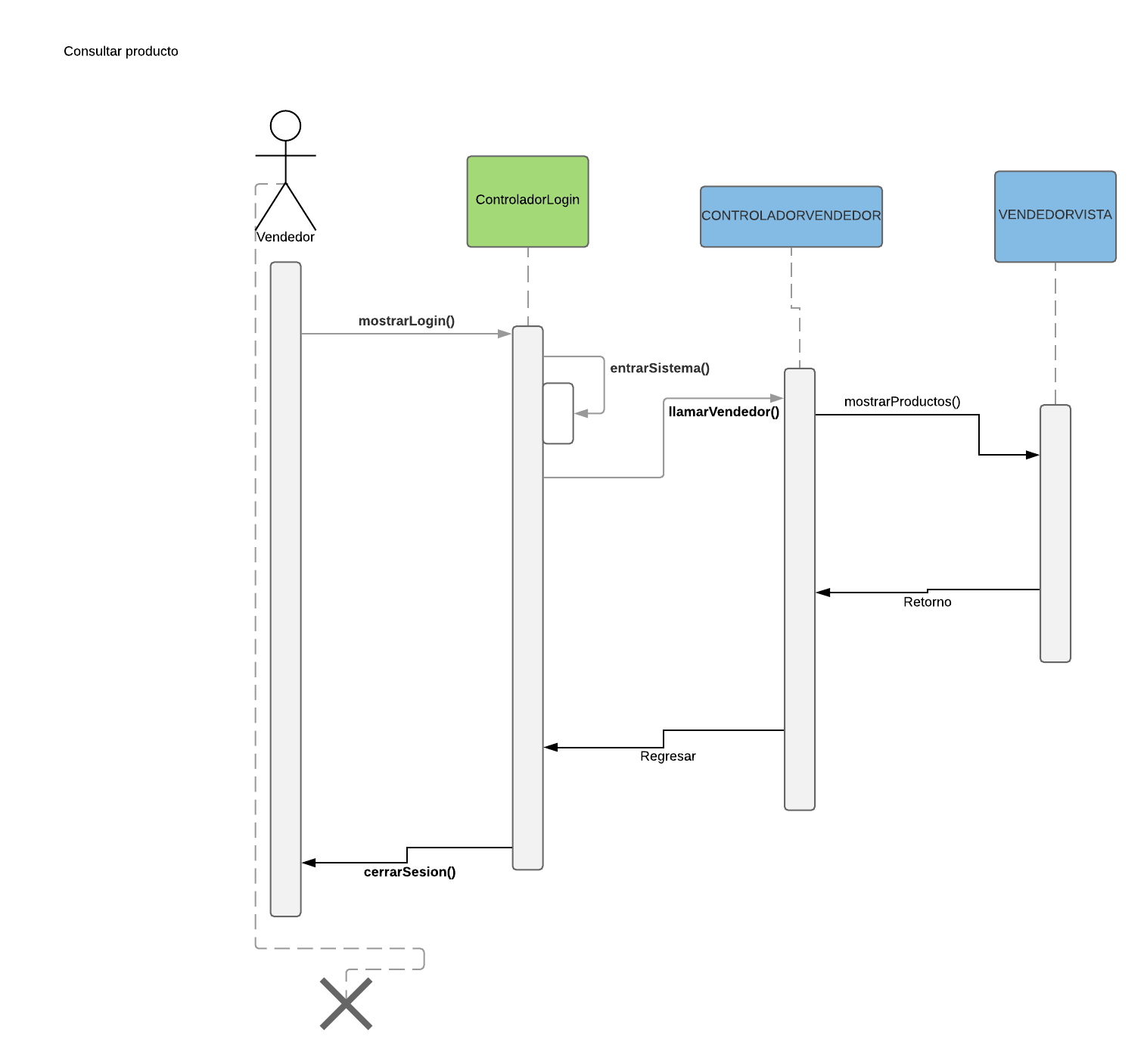
Ingreso al Sistema

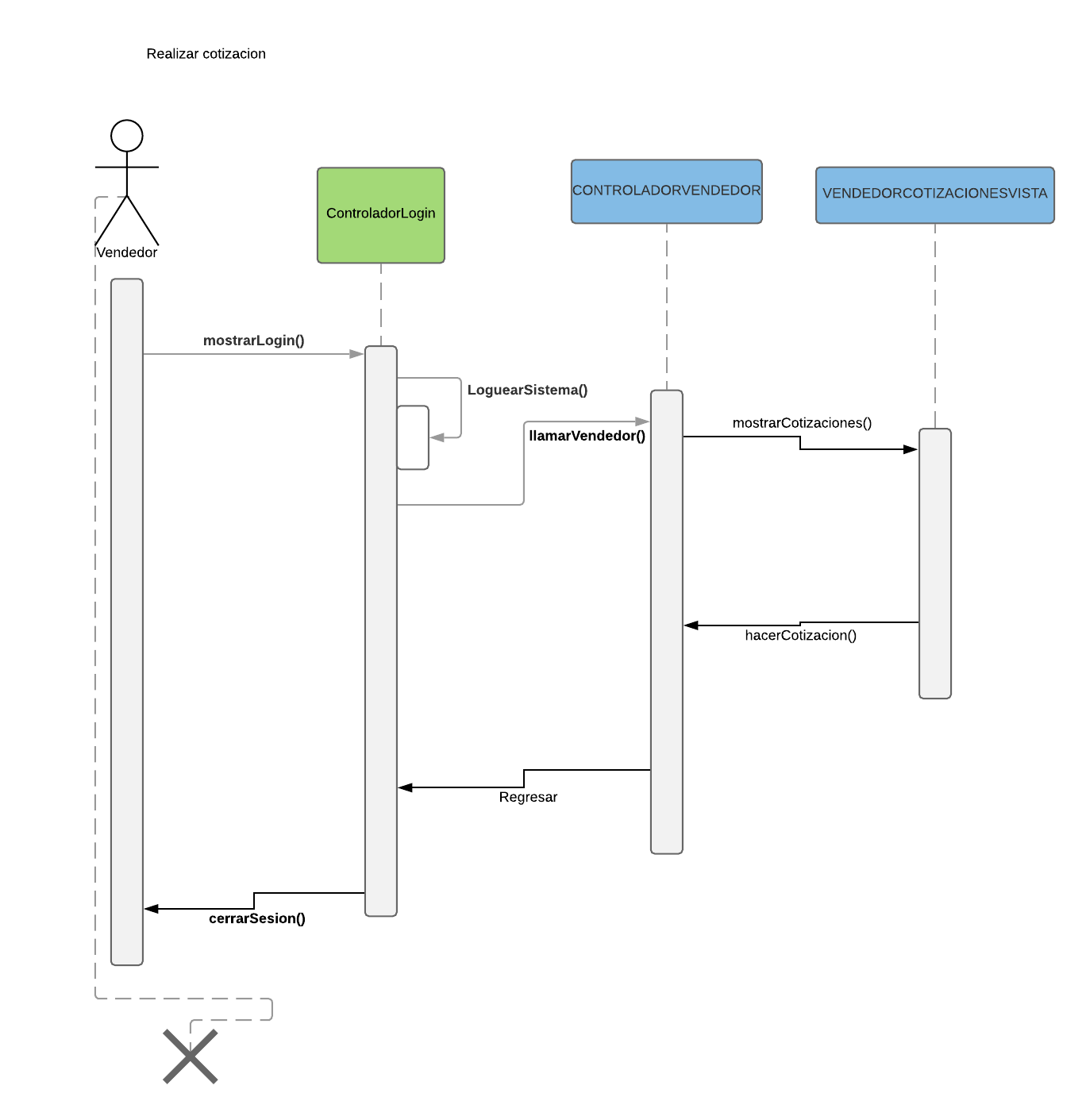


Emitir Documento

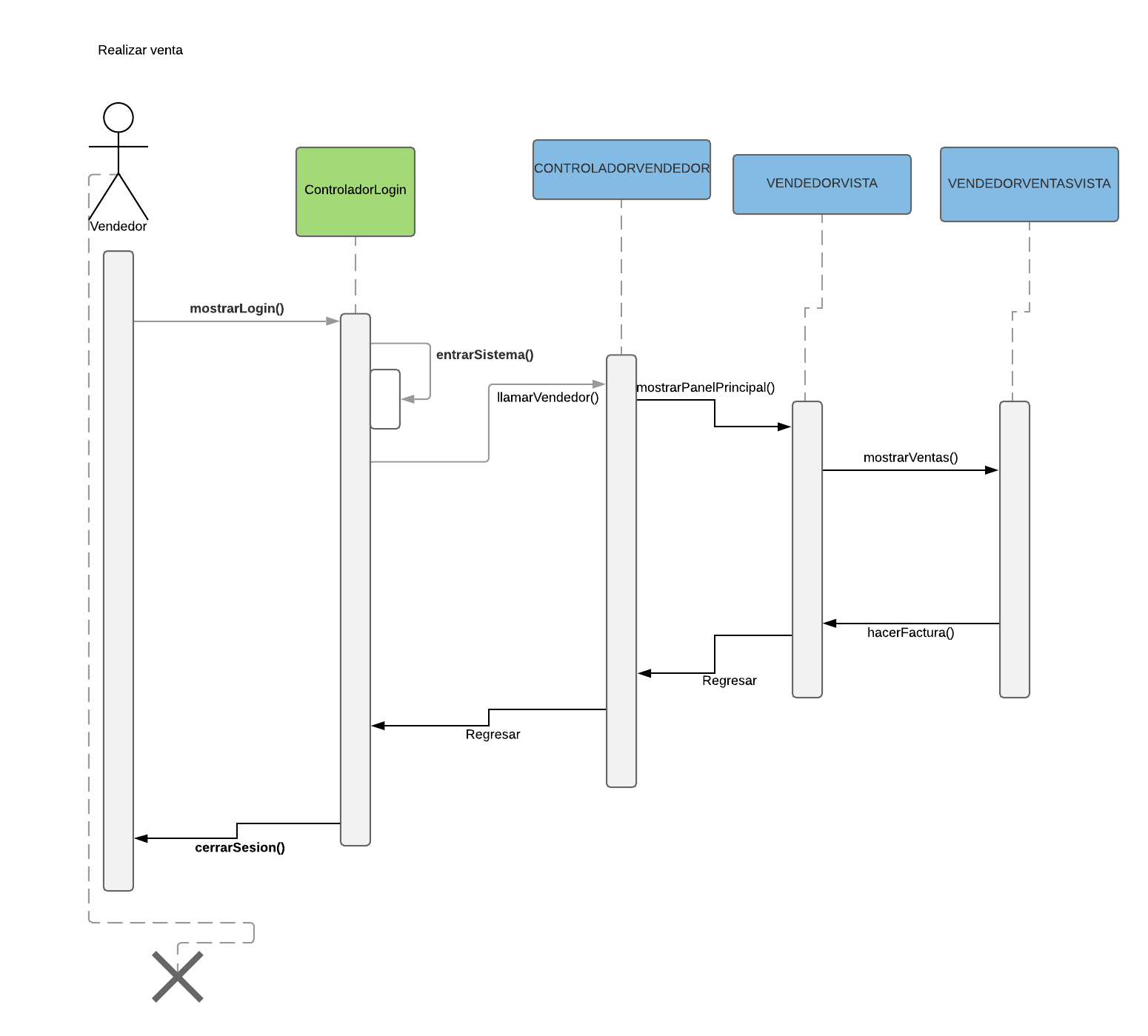


Consultar Producto



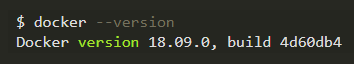
Realizar Cotización

Realizar Venta



**Comandos Dockers**

Lo primero que debemos hacer es instalar Docker de la pagina oficial, luego verificamos la instalación con el siguiente comando:

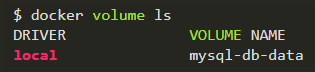


Luego de tener descargado Docker deberíamos descargarnos una imagen (contenedor de MySQL), pero realizar esto nos causaba un problema que la información de dicho contenedor no persistía luego de terminar el proceso de Docker o simplemente reiniciar el computador, por lo tanto, para solucionar dicho inconveniente primero debemos montar un volumen. Para crear un volumen en Docker aplicamos el siguiente comando:



Donde en este ejemplo “mysql-db-data” es el nombre que le proporcionamos al nuestro volumen.

Podemos verificar la creación de nuestro primer volumen usando el comando ls de la siguiente manera:

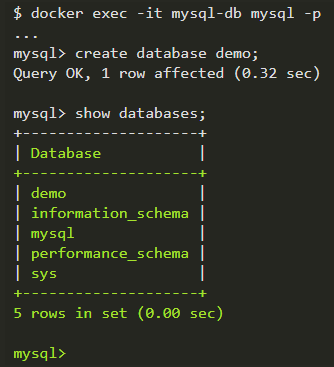


Una vez verificado la creación del volumen, ya podemos persistir de la información solucionando el primer problema presentado al usar Docker, entonces seguimos levantando la imagen o contenedor de mysql en el volumen antes creado con el siguiente comando:



* -d: Deatached Mode es la forma en que indicamos que corra en background.
* -p : puerto, el contenedor corre en el puerto 3306 pero hacemos un bind para que lo escuchemos en Host el puerto 33061.
* -name : para no tener que hacer referencia al hash le asignamos un nombre.
* -e : environment le asignamos la contraseña.
* --mount es el volumen que acabamos de crear

Y por último para ingresar a nuestro contenedor de forma dinámica empleamos el siguiente comando:



Luego para facilitarnos el manejo de tablas podemos conectarnos por MySQL Workbench a través de los siguientes datos:

* **Host: 127.0.0.1**
* **User: root**
* **Password: secret**
* **Puerto: 33061**

**Capturas e información de Correcciones**

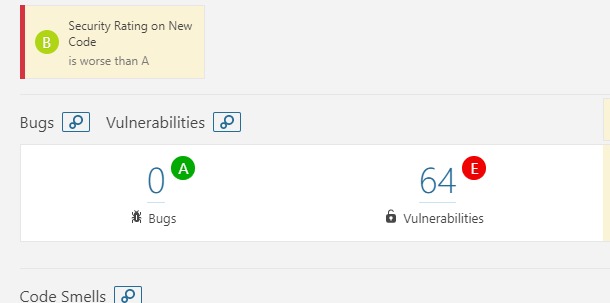
* Bug Corregido, no cerrar Prepared Statement
* Bugs encontrados al inicio de la depuración:



* Bug Cargo Dao Solucionado:



* Sin Bugs:



* Eliminado 30 vulnerabilidades, haciendo final a event handlers

