Leonardo David Gomez Pesantes

Vielka Verónica Villavicencio Vallejo

Dennys Eduardo López Damián



Profesor

Ing. David Jurado Mosquera

Paralelo

2

Diseño de software

Proyecto

Tabla de contenido

[Descripción del proyecto 2](#_Toc17325792)

[Diagrama de Caso de Uso 3](#_Toc17325793)

[Diagrama de clases 9](#_Toc17325794)

[Diagramas de Secuencias 10](#_Toc17325795)

[Diagrama Normalizado de la Base de Datos 14](#_Toc17325796)

[Reseña del refactoring 15](#_Toc17325797)

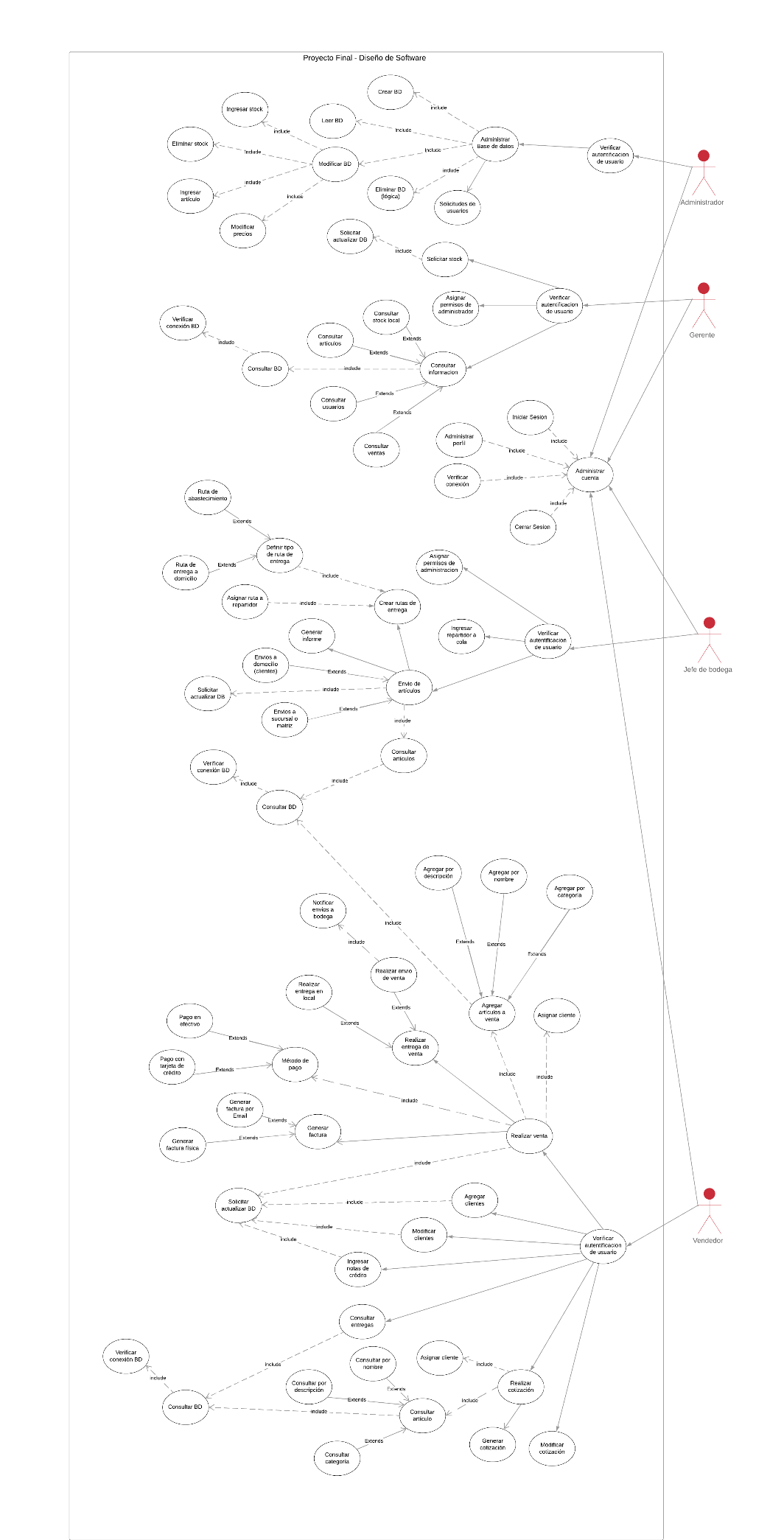
[Pruebas Unitarias 17](#_Toc17325798)

[Dockers 22](#_Toc17325799)

# **Descripción del proyecto**

Este proyecto consiste en diseñar un sistema de la empresa TecnoImport que le permita almacenar en una base de datos información de sus productos, empleados y clientes, que esto le ayude a manejar de mejor forma los activos de su empresa.  
Para realizar este sistema necesitamos utilizar el uso de arquitecturas como MVC que son indispensable para poder saber que necesita realmente el cliente en su sistema. La utilización de una base de datos es primordial para almacenar los datos de la empresa.

# **Diagrama de Caso de Uso**

****

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **PDS-A01** |
| **Titulo** | Modificar BD |
| **Autor** | Leonardo Gómez |
| **Actor principal** | Administrador |
| **Actor secundario** |  |
| **Usa** | Modificar Stock, Ingresar Stock, Modificar precios |
| **Extiende** |  |
| **Descripción** | El administrador puede realizar diferentes funciones que permitan modificar la BD, entre las cuales tenemos  Eliminar stock  Ingresar stock  Ingresar artículo  Modificar precios  Con la finalidad de tener una Base de Datos confiable referente a los productos que existen disponibles en la empresa |
| **Precondición** | El administrador está autenticado y posee una cuenta en el sistema |
| **Secuencia normal** | 1. El administrador selecciona la opción de modificar la Base de Datos. 2. El sistema despliega las diferentes funciones que existen (mencionadas en la descripción) para lo que el administrador selecciona función que él vaya a ejecutar. 3. El administrador se encarga de manipular la información de la Base de Datos de acuerdo a lo que se le haya solicitado. 4. El administrador verifica los cambios y los guarda en la Base de Datos. |
| **Secuencia alternativa** | 2.1.- La conexión con la Base de Datos tuvo algún error, en dicho caso, la solicitud hacia la Base no pudo ser establecida, por lo que se le notificará al usuario de aquello. |
| **Postcondición** | La Base de Datos reflejará los respectivos cambios del stock de acuerdo a lo que el administrador haya realizado. |

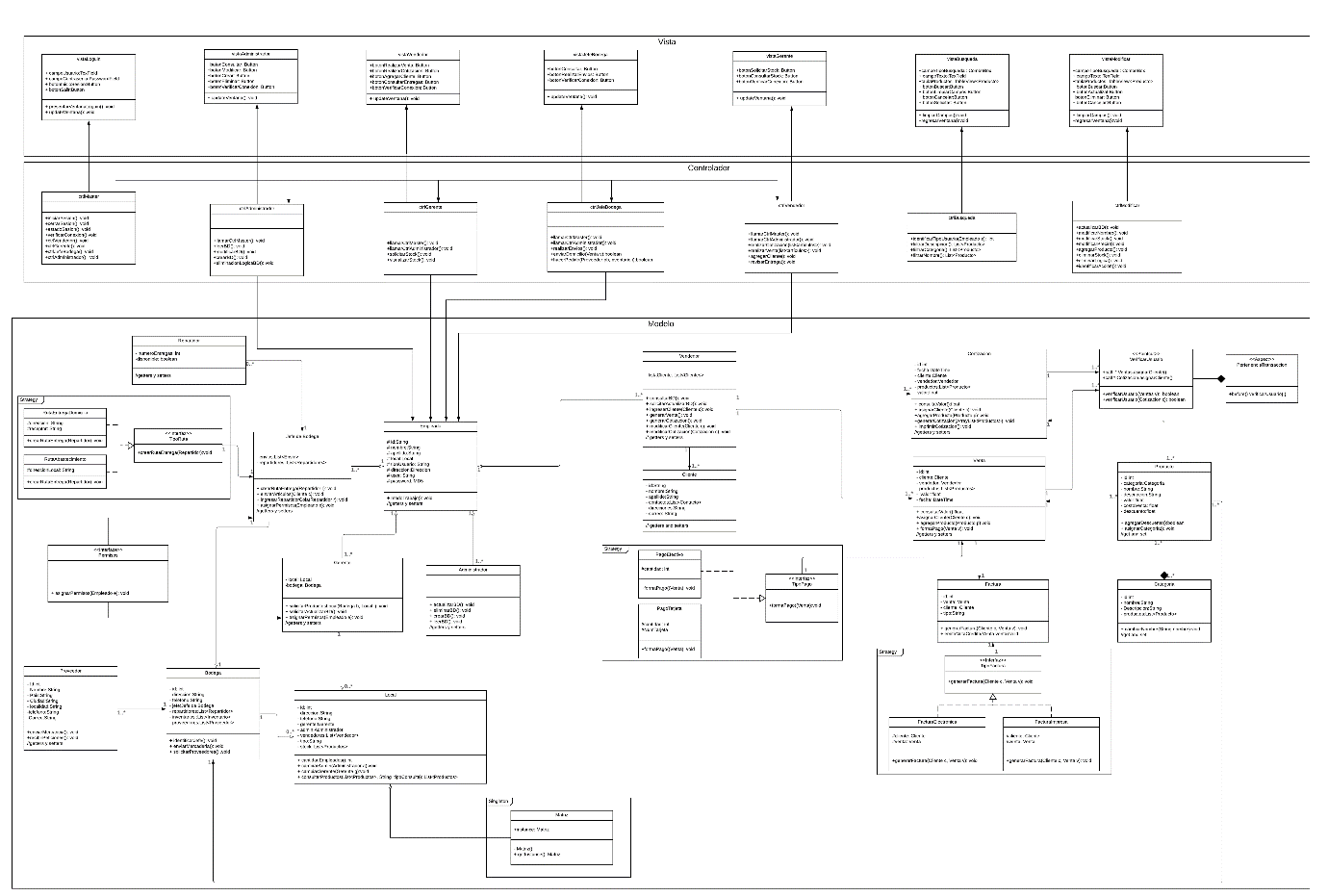
|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **PDS-A02** |
| **Titulo** | Envíos a sucursal o matriz |
| **Autor** | Leonardo Gómez |
| **Actor principal** | Jefe de bodega |
| **Actor secundario** |  |
| **Usa** |  |
| **Extiende** | Envío de artículos |
| **Descripción** | En este caso, se realizó el envío a una local propio de la empresa, ya sea este una sucursal o matriz, permitiendo que se pueda realizar la distribución del stock en los diferentes locales que requieran artículos. |
| **Precondición** |  |
| **Secuencia normal** | 1. El gerente realiza una solicitud de stock. 2. El jefe de bodega realiza la consulta del stock existente y notifica al gerente la cantidad de stock disponible. 3. El jefe de bodega asigna a un repartidor que se encuentre disponible en la cola de repartidores para que se encargue de realizar la entrega del stock solicitado 4. El jefe de bodega entrega todos los papeles necesarios para que el repartidor se los entregue al gerente de la sucursal o matroz 5. Se realiza el envío a la sucursal o matriz. 6. Se genera un informe con los detalles del stock que fue enviado a la sucursal que realizó la solicitud. |
| **Secuencia alternativa** | 2.1.- En caso que no exista el stock solicitado, el jefe de bodega le notifica al gerente para que este decida una cantidad de stock diferente. |
| **Postcondición** | El stock se encontrará en camino a las sucursales o matriz que hayan realizado los pedidos, pudiendo así ser reabastecidos; el gerente se encarga de firmar los papeles que el jefe de bodega haya entregado al repartidor, para tener la constancia que la entrega se realizó con éxito. |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **PDS-A03** |
| **Titulo** | Solicitar stock |
| **Autor** | Leonardo Gómez |
| **Actor principal** | Gerente |
| **Actor secundario** |  |
| **Usa** |  |
| **Extiende** |  |
| **Descripción** | El gerente puede realizar solicitud de stock debido a que ya se acabó el existente o está próximo a acabarse los artículos en su sucursal |
| **Precondición** | El gerente está autenticado y posee una cuenta en el sistema |
| **Secuencia normal** | 1. El gerente realiza una solicitud al jefe de bodega para que envié determinados artículos. 2. El jefe de bodega recibe la solicitud, consulta la base de datos de la bodega y notifica al gerente de la cantidad de stock que se encuentra en la bodega 3. El gerente notifica al jefe de bodega que desea confirmar el pedido del stock. 4. El jefe de bodega se encarga de realizar el respectivo procedimiento para realizar el envío del stock |
| **Secuencia alternativa** | 2.1.- No hay el stock solicitado por el gerente, por lo que modifica su solicitud de acuerdo a la cantidad de stock que exista en bodega. |
| **Postcondición** | El local entrará en espera para que el nuevo stock llegue, situación que como mucho tomará un día en completarse |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **PDS-A04** |
| **Titulo** | Realizar cotización |
| **Autor** | Leonardo Gómez |
| **Actor principal** | Vendedor |
| **Actor secundario** |  |
| **Usa** | Asignar cliente, Consultar artículo |
| **Extiende** |  |
| **Descripción** | El vendedor puede realizar cotizaciones, ingresando en la cotización los artículos que el cliente desea, con la finalidad de poder obtener información económica los elementos que está considerando en comprar. |
| **Precondición** | El vendedor está autenticado y posee una cuenta en el sistema.  El cliente a quien se realizará la cotización fue ingresado en el sistema. |
| **Secuencia normal** | 1. El vendedor indica al sistema a nombre de quien se realizará la cotización. 2. El cliente realiza la selección de artículos que él está interesado en adquirir. 3. El vendedor se encarga de realizar una consulta con la base de datos para obtener los precios de este grupo de artículos. 4. El sistema recibe cada uno de los productos que el vendedor ingresa, calculando el valor total de los productos que este ingrese. 5. El vendedor entrega un documento con la información que requiere la cotización. |
| **Secuencia alternativa** | 2.1.- En caso que no exista conexión con la Base de Datos, las solicitudes de consulta no pueden ser llevadas a cabo por parte de los vendedores. |
| **Postcondición** | Se tendrá una cotización en la cual se reflejará información sobre los artículos que el cliente haya seleccionado. |

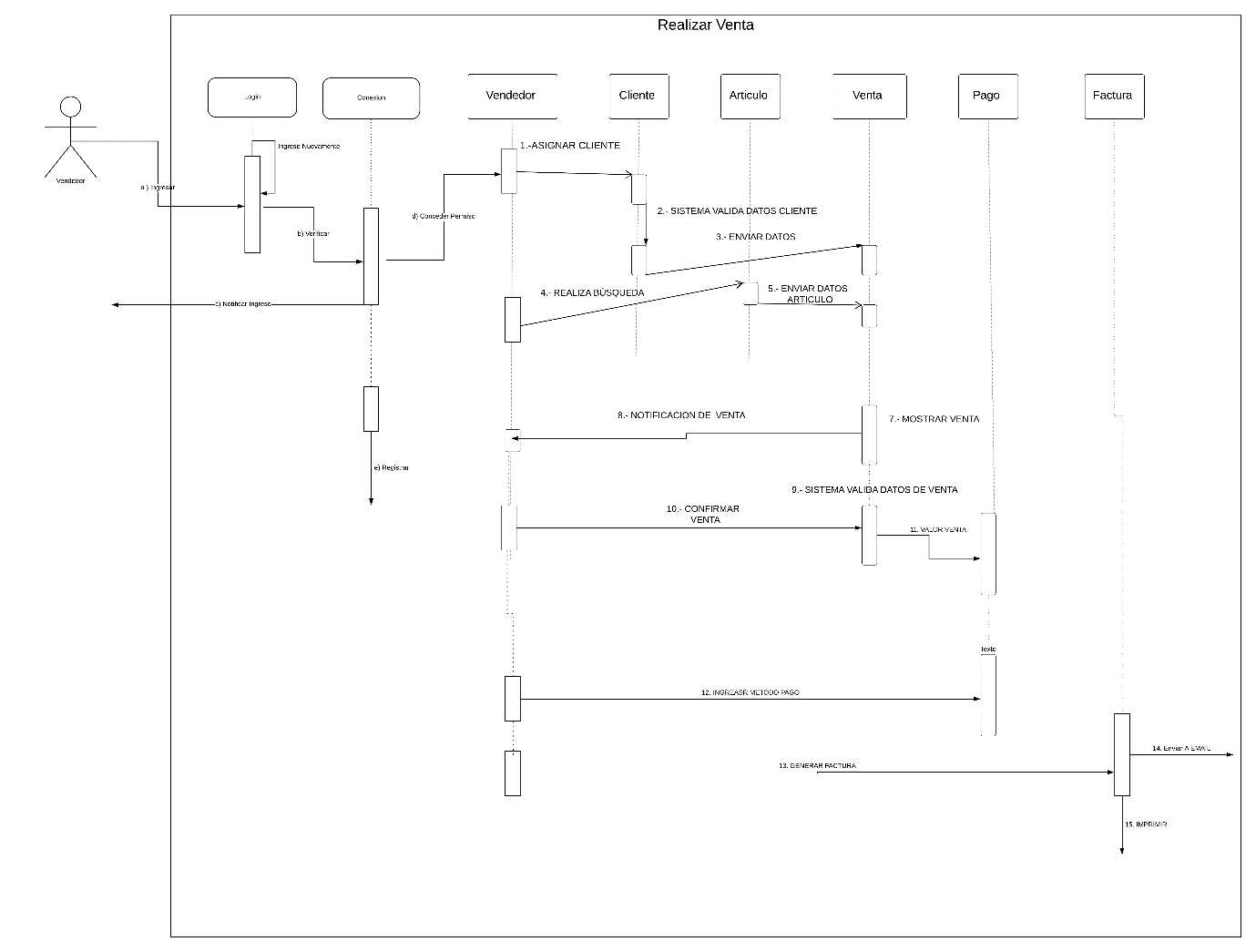
|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **PDS-A05** |
| **Titulo** | Realizar venta |
| **Autor** | Leonardo Gómez |
| **Actor principal** | Vendedor |
| **Actor secundario** |  |
| **Usa** | Asignar cliente, agregar artículos, Actualizar BD, método de pago |
| **Extiende** |  |
| **Descripción** | El vendedor puede realizar una venta, para lo que necesita el cliente a quien realizará la venta y los artículos que este último irá a comprar |
| **Precondición** | El vendedor está autenticado y posee una cuenta en el sistema  El cliente a quien se realizará la venta fue ingresado en el sistema. |
| **Secuencia normal** | 1. El cliente le entrega al vendedor todos los artículos que desea comprar. 2. El vendedor selecciona los diferentes artículos y realiza una notificación al sistema sobre el artículo y la cantidad del artículo que será retirad, realizando una solicitud de actualización al administrador de la Base de Datos. 3. El administrador recibe la notificación y se encarga de realizar la actualización de la Base de Datos. 4. El vendedor tiene permitido retirar los artículos solicitados. 5. Dicha venta genera una factura para el cliente. 6. Se concluye la venta. |
| **Secuencia alternativa** | 2.1.- No hay conexión con la DB, por lo que la actualización no puede ser llevada a cabo de manera inmediata.  2.2.- En caso que no exista stock disponible en el local, el vendedor solicita a la bodega información sobre los productos y si estos existen en bodega |
| **Postcondición** | La Base de datos es actualizada con la disminución de los productos vendidos, además que el cliente se encontrará registrado en el sistema; con todos estos datos, el vendedor podrá generar una factura para que el cliente pueda utilizarla como referencia de la realización de la compra. |

# **Diagrama de clases**

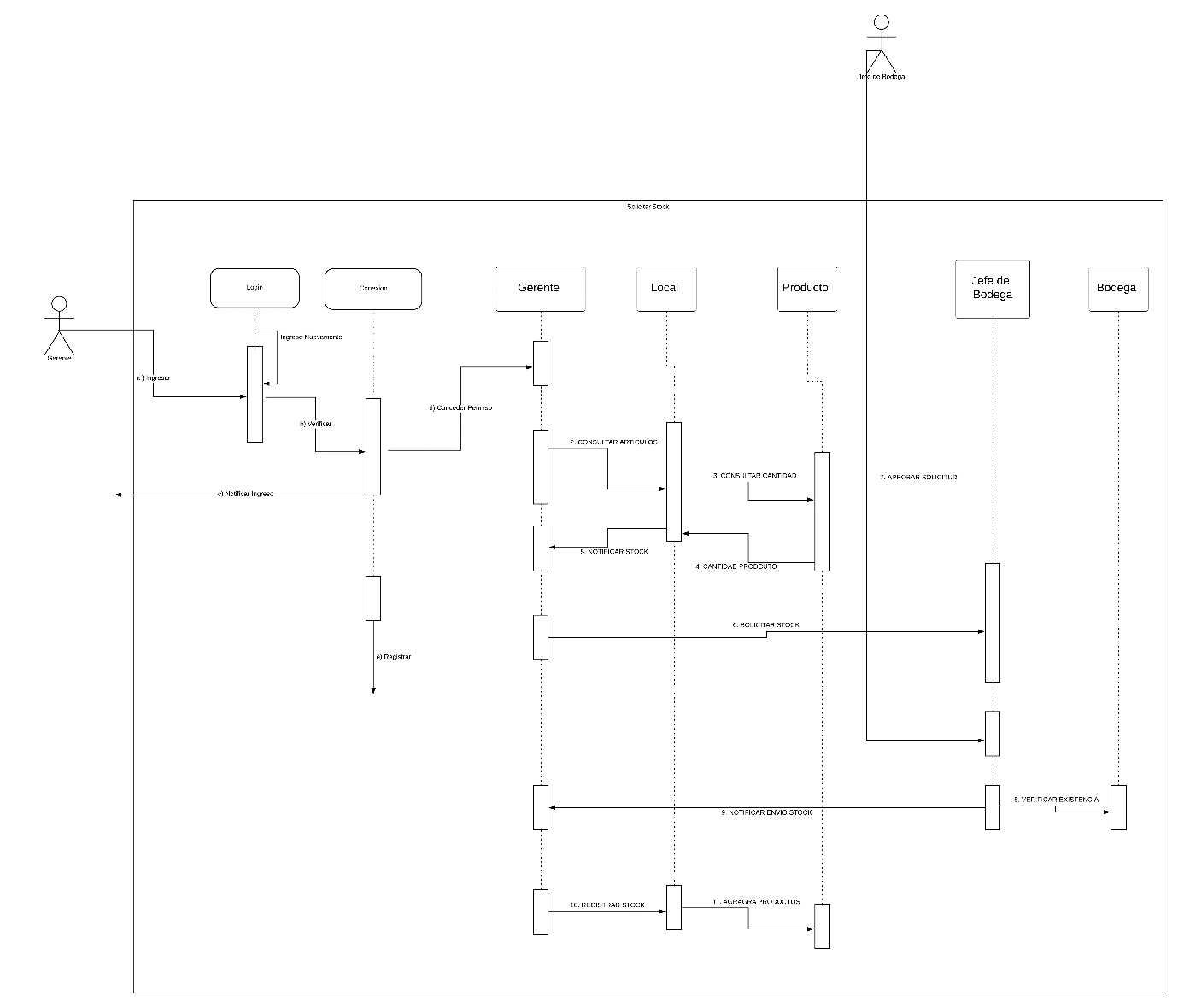
****

# **Diagramas de Secuencias**

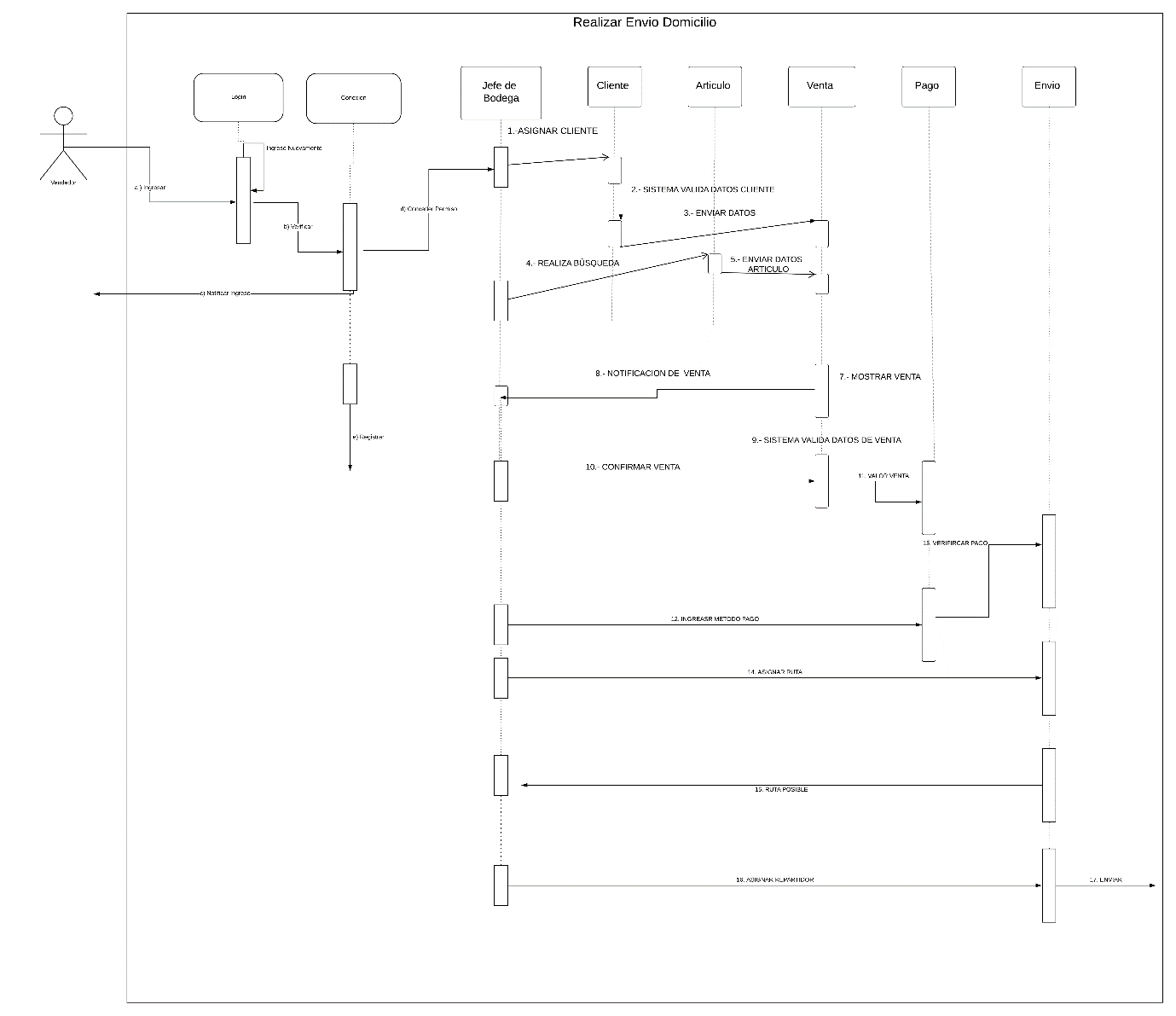
Secuencia de ventas



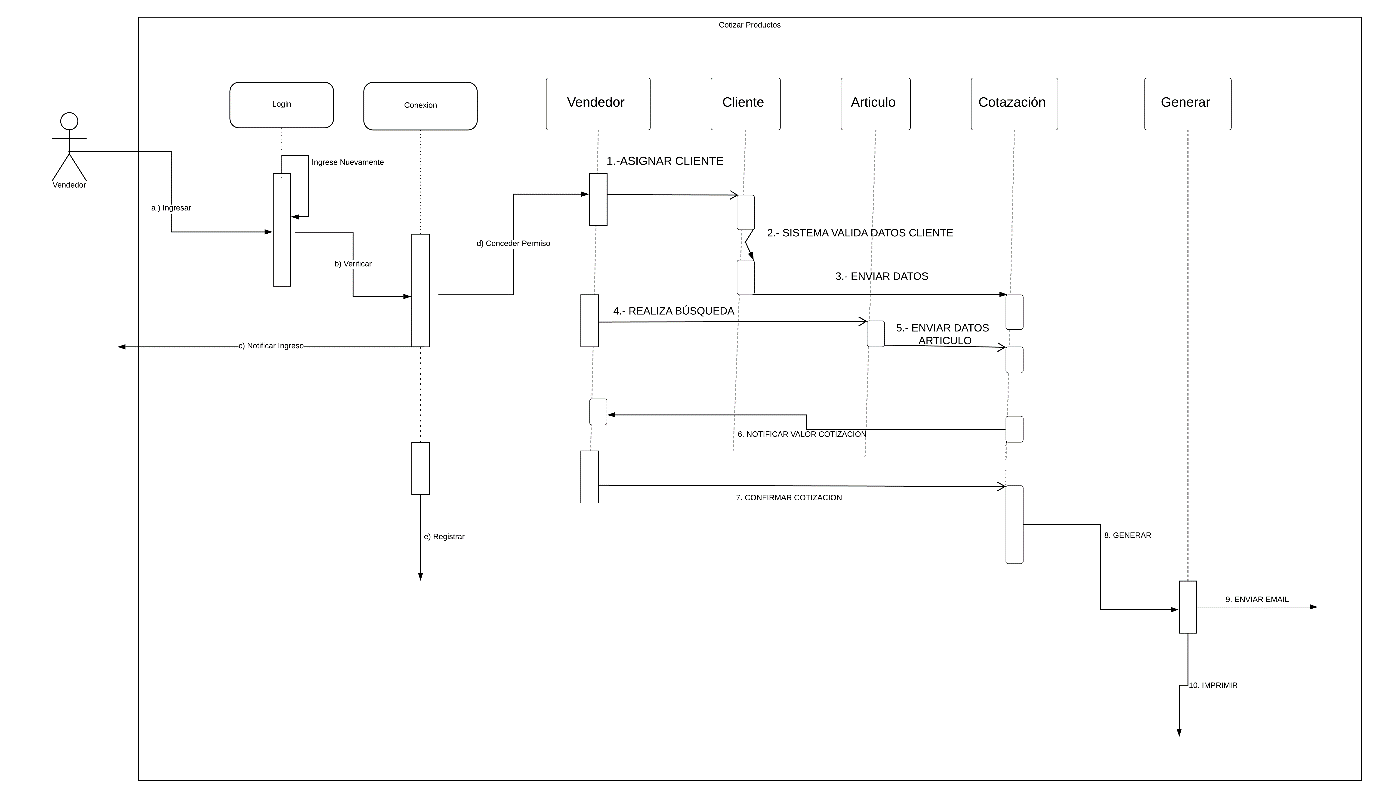
Secuencia de Solicitud de Stock



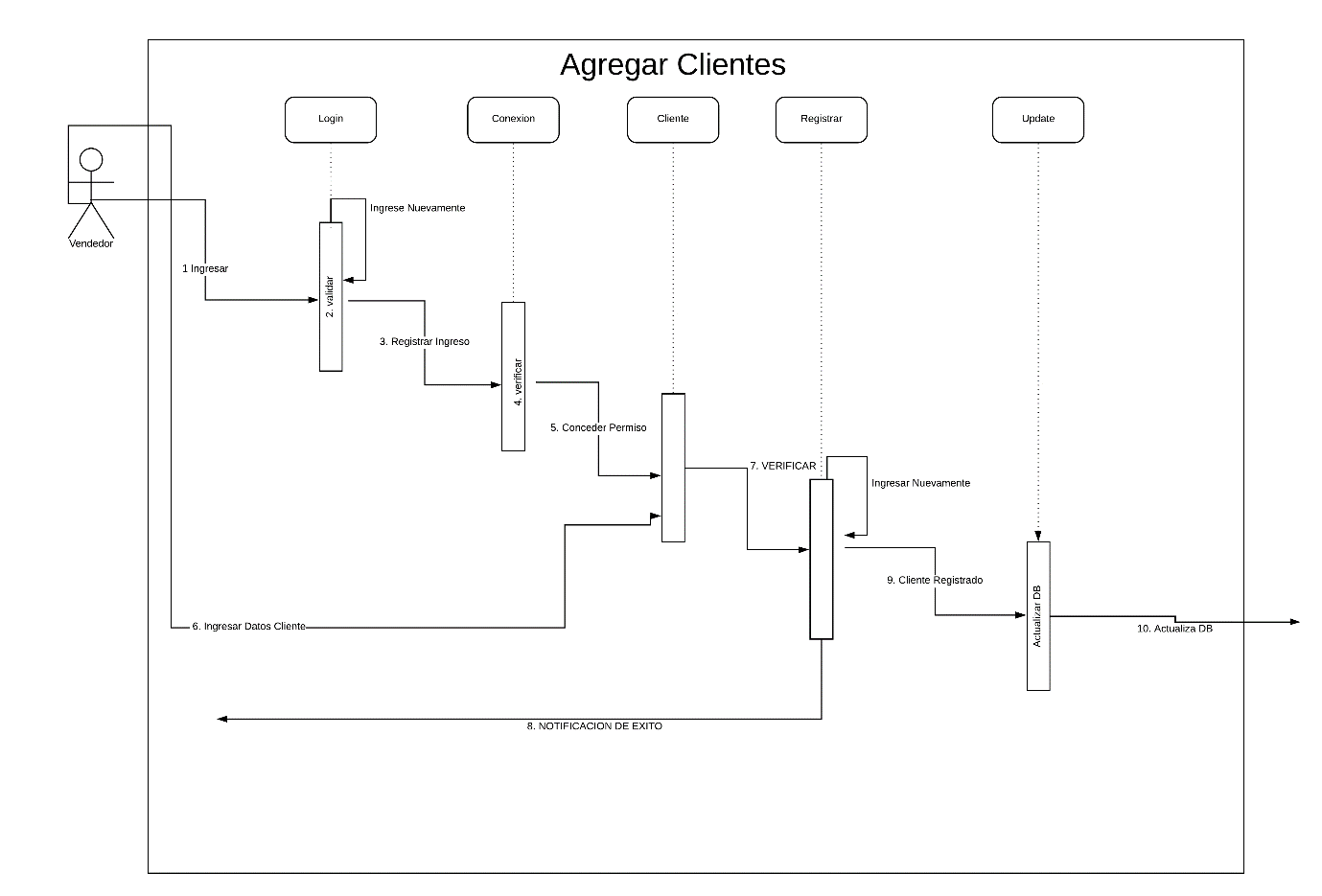
Secuencia de Envió a Domicilio



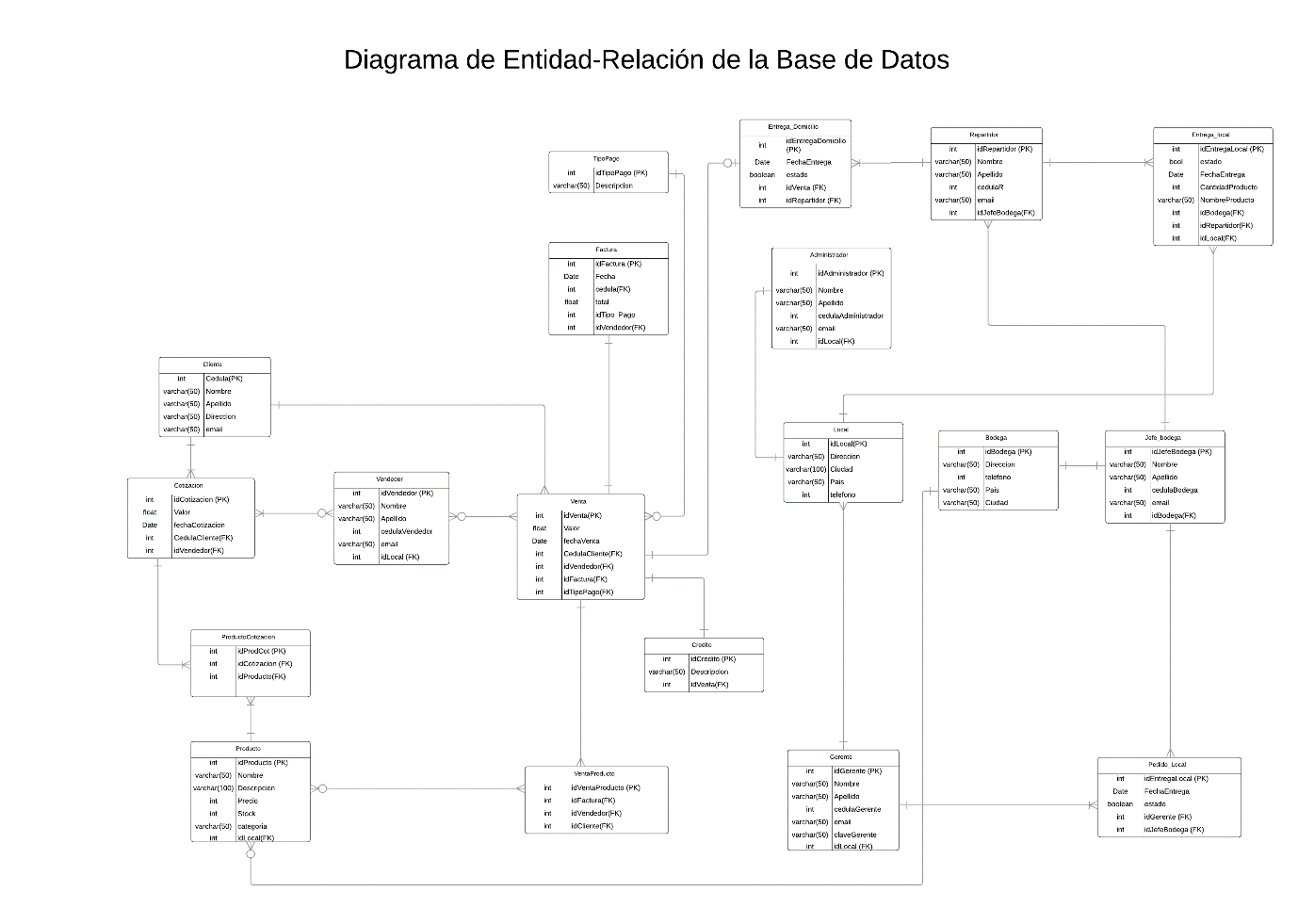
Secuencia de Cotizar Productos



Secuencia de Agregar Cliente

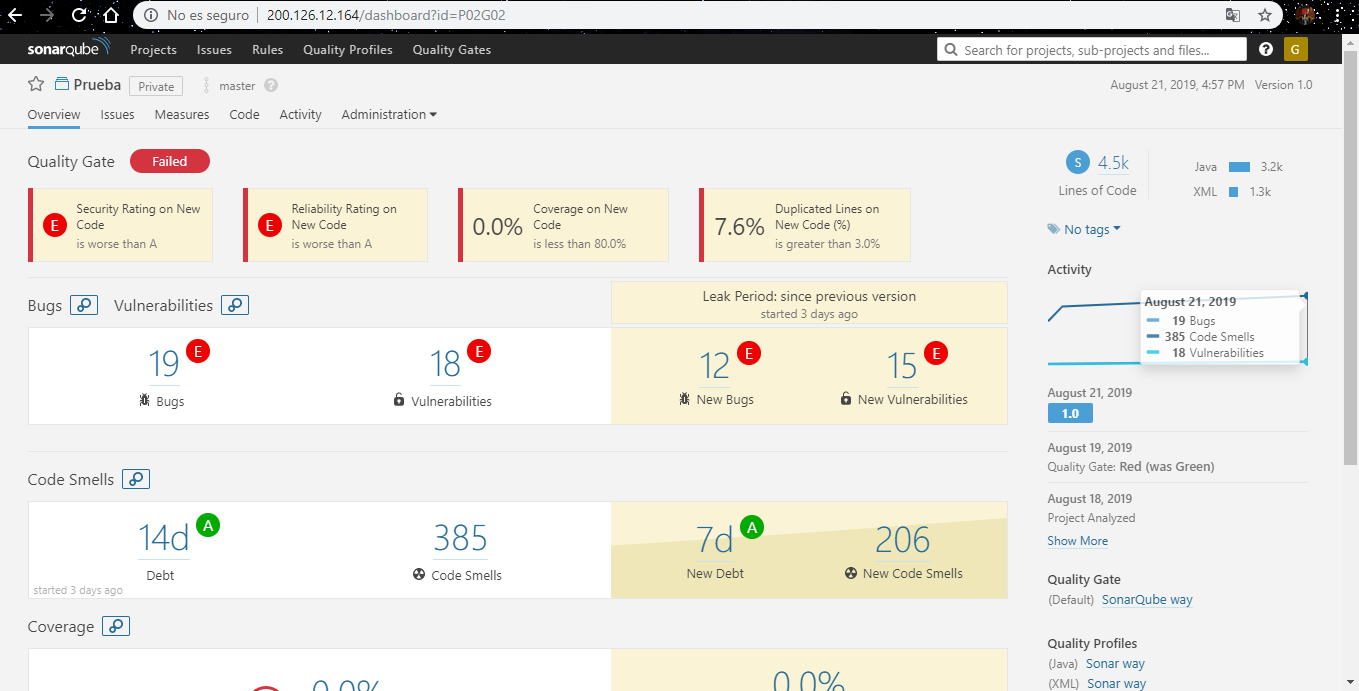


# **Diagrama Normalizado de la Base de Datos**

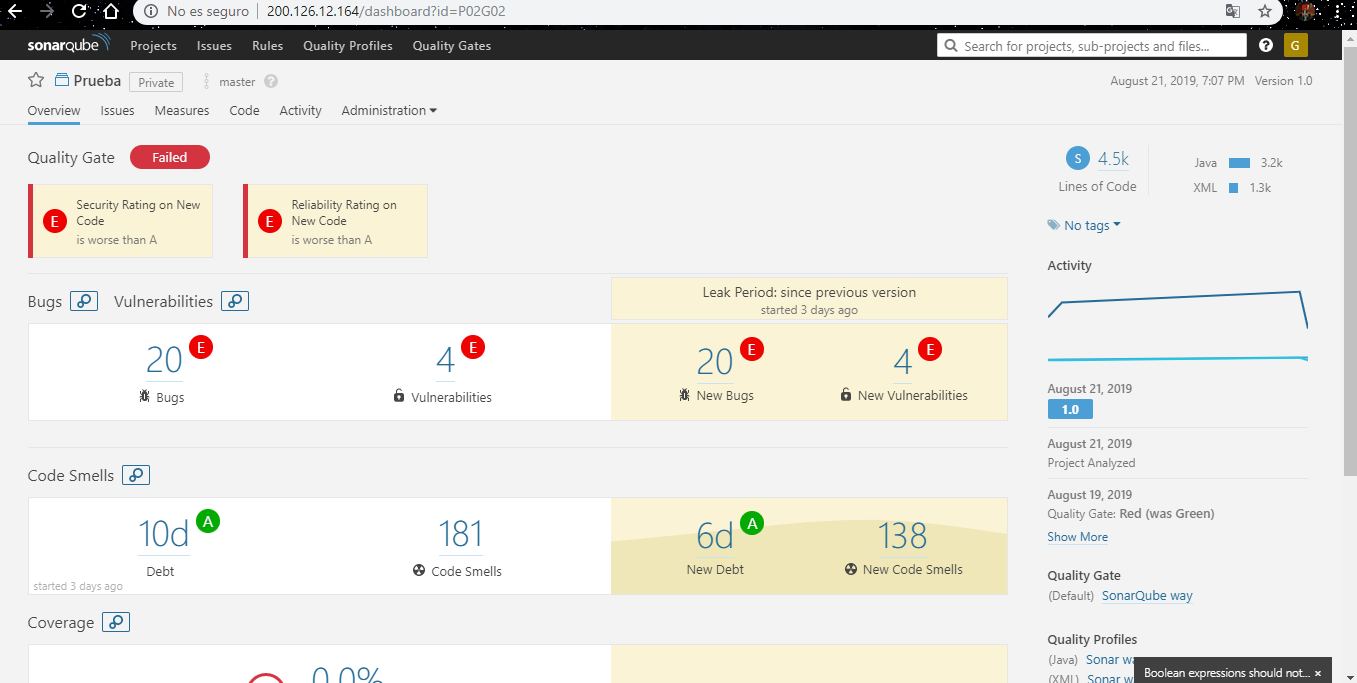
****

# **Reseña del refactoring**

**Antes**

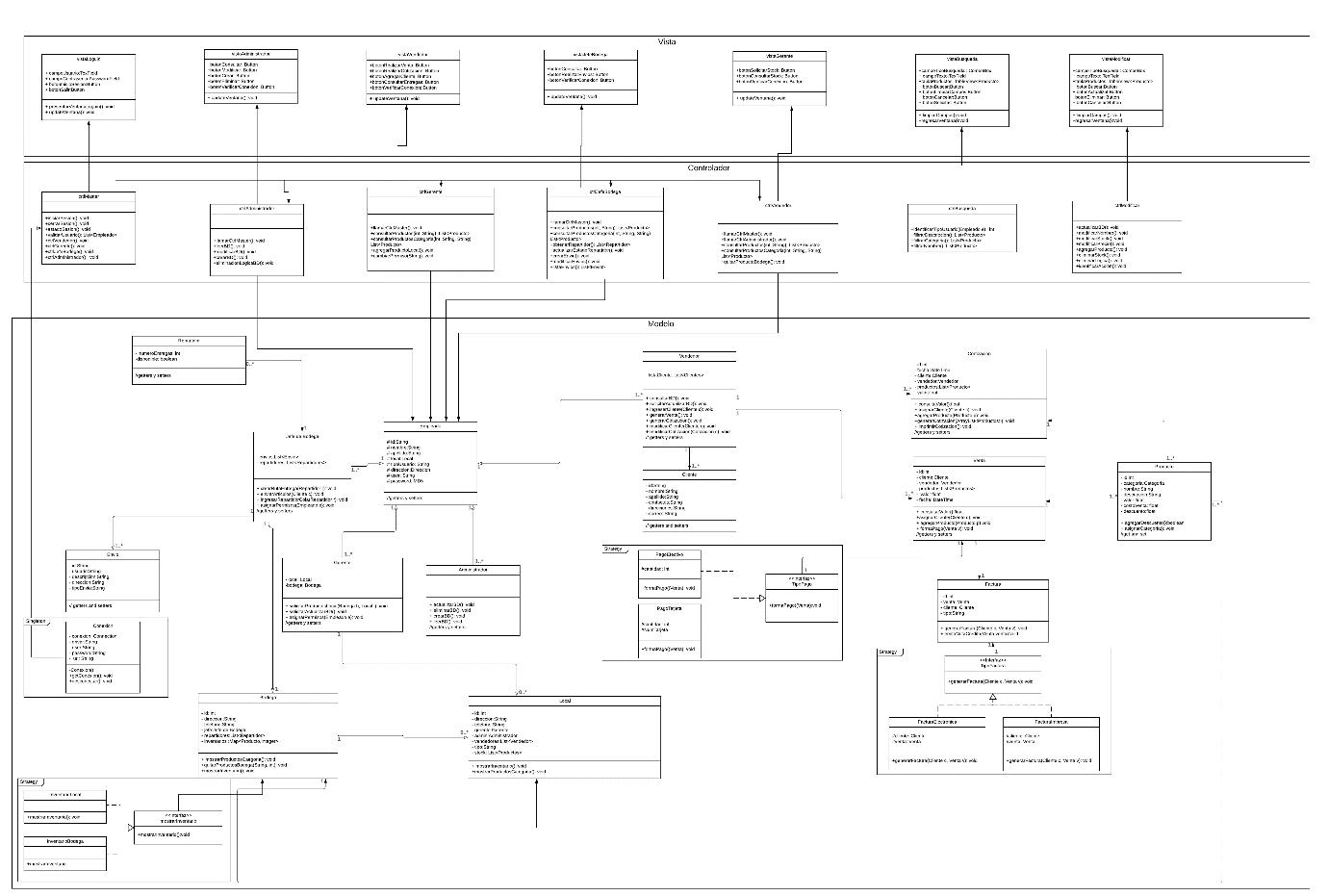


**Después**



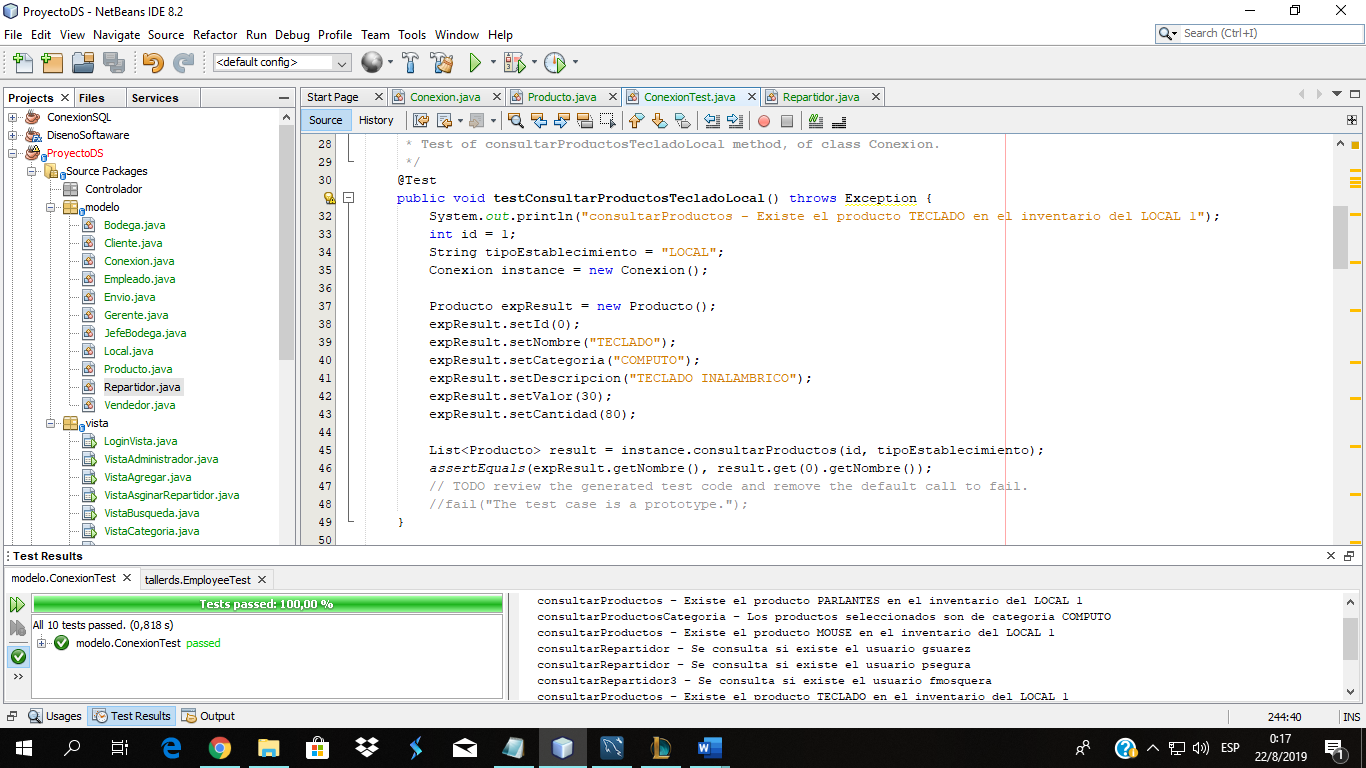
Durante el proceso del refactoring que se llevó a cabo en nuestro código, se encontraron Smell Code que en su mayoría correspondía a Código muerto, clases o métodos vagos y sobre todo, clases especulativas, las cuales fueron en su mayoría para métodos que posteriormente se pensaban implementar. Además de determinados problemas de declaraciones de variables, campos temporales, el uso de impresión por consola de ciertos mensajes o valores, etc. El procedimiento del refactoring que se llevó a cabo consistió en eliminar todo el código que no intervenía en el funcionamiento del programa hasta el punto que se tiene avanzado, principalmente eliminando clases que no interactuaban con el programa, se eliminaron también varios mensajes por consola, además de la inutilización de varias variables que no cumplían ninguna función.

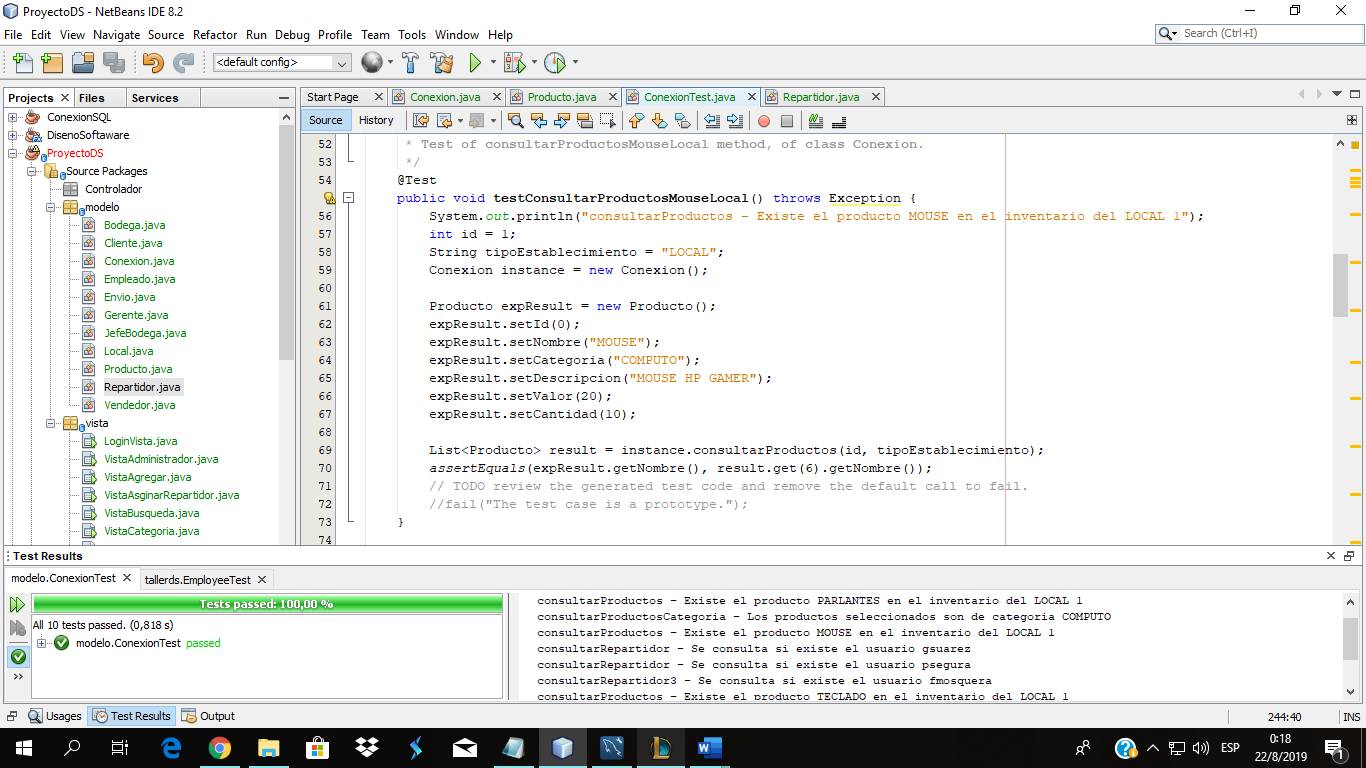
Este procedimiento ayudó a disminuir en gran medida los malos olores, sin embargo, al haber utilizado para el paquete de Vista la librería JFrame, la gran mayoría de los Smell Code que se presentan luego de la refactorización, corresponden a elementos internos que no se pueden modificar.

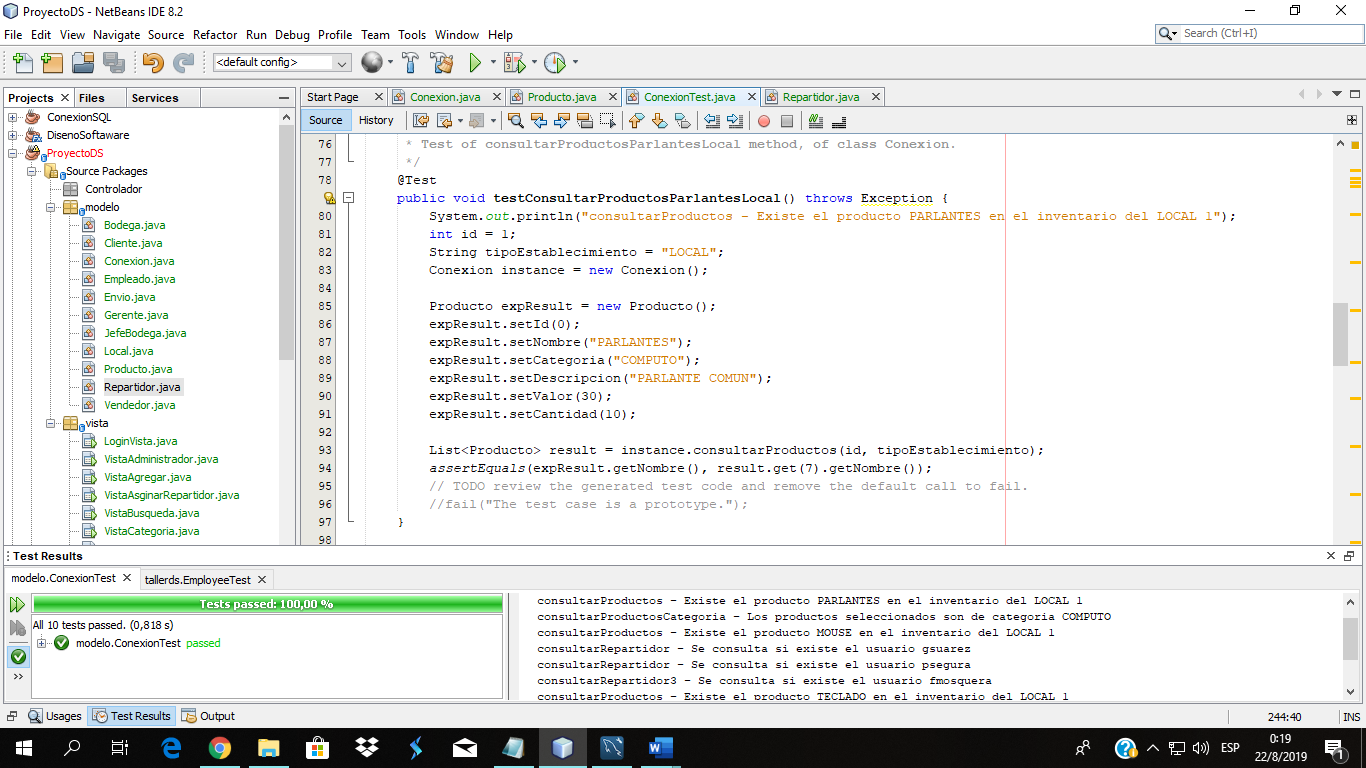


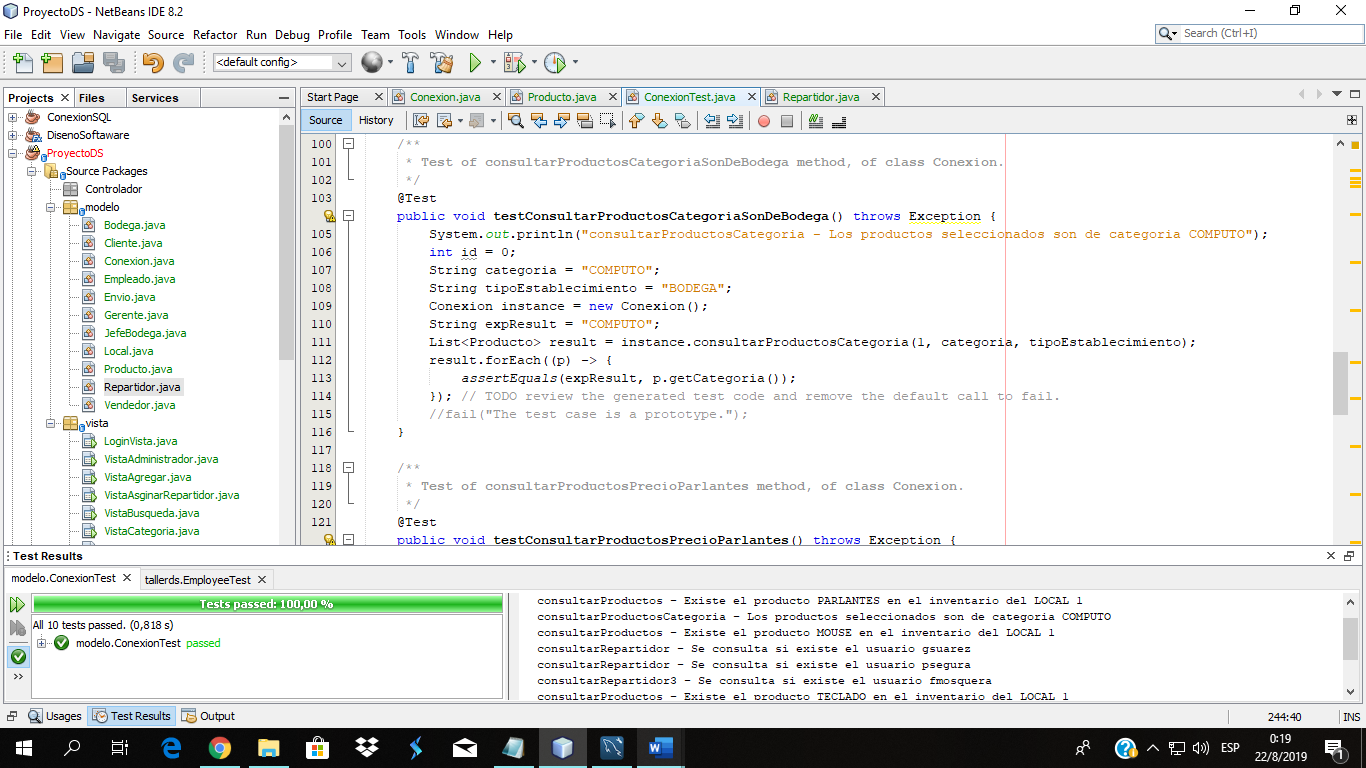
Finalmente se obtuvo un MVC con la respectiva refactorización de las clases que se tuvieron que eliminar, dejando clases útiles o que interactuaban con el programa.

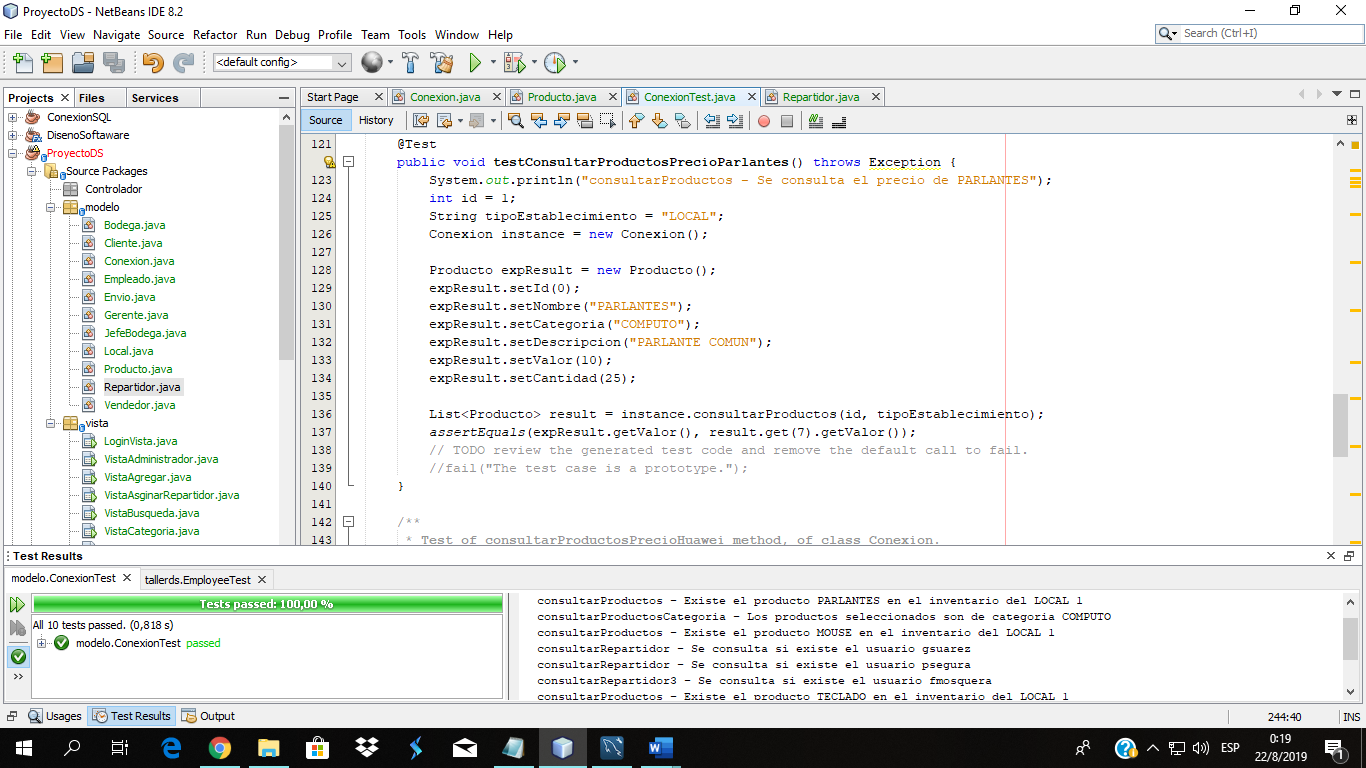
# Pruebas Unitarias

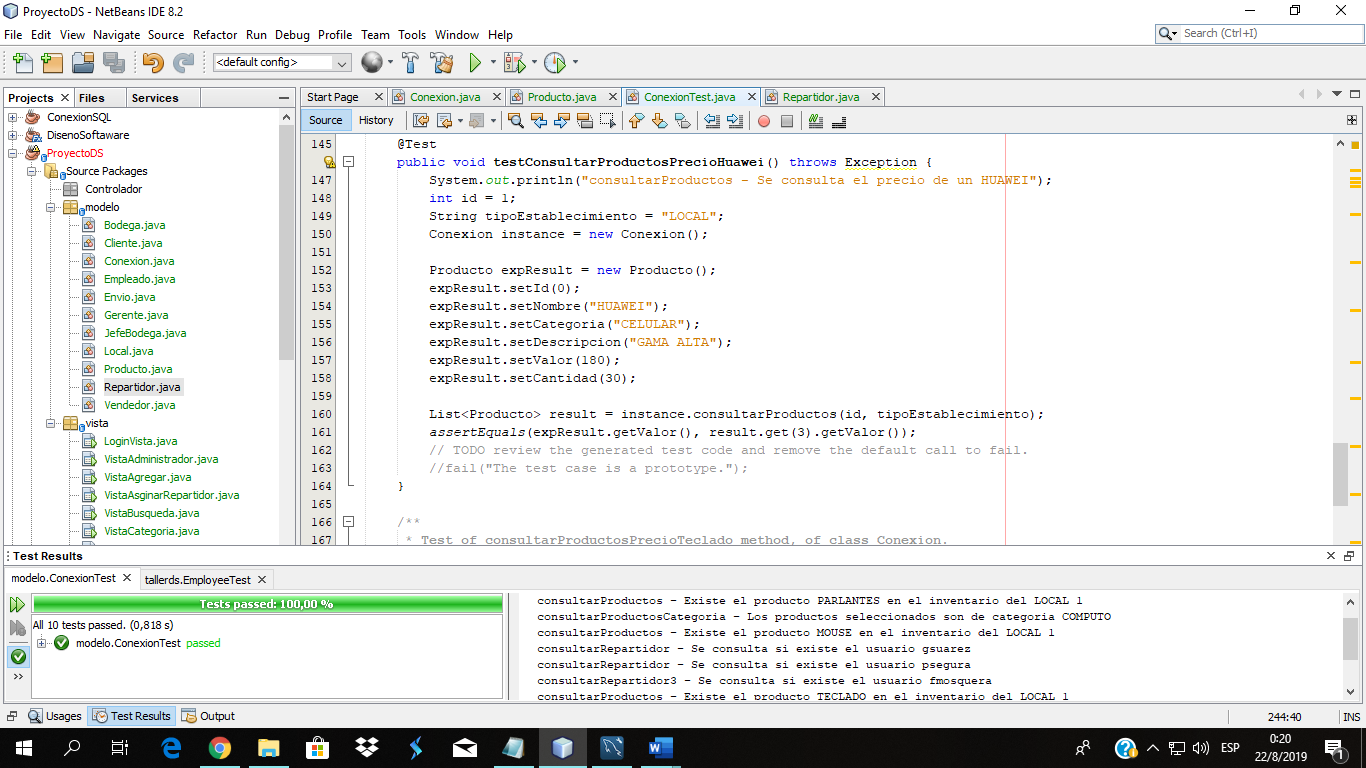


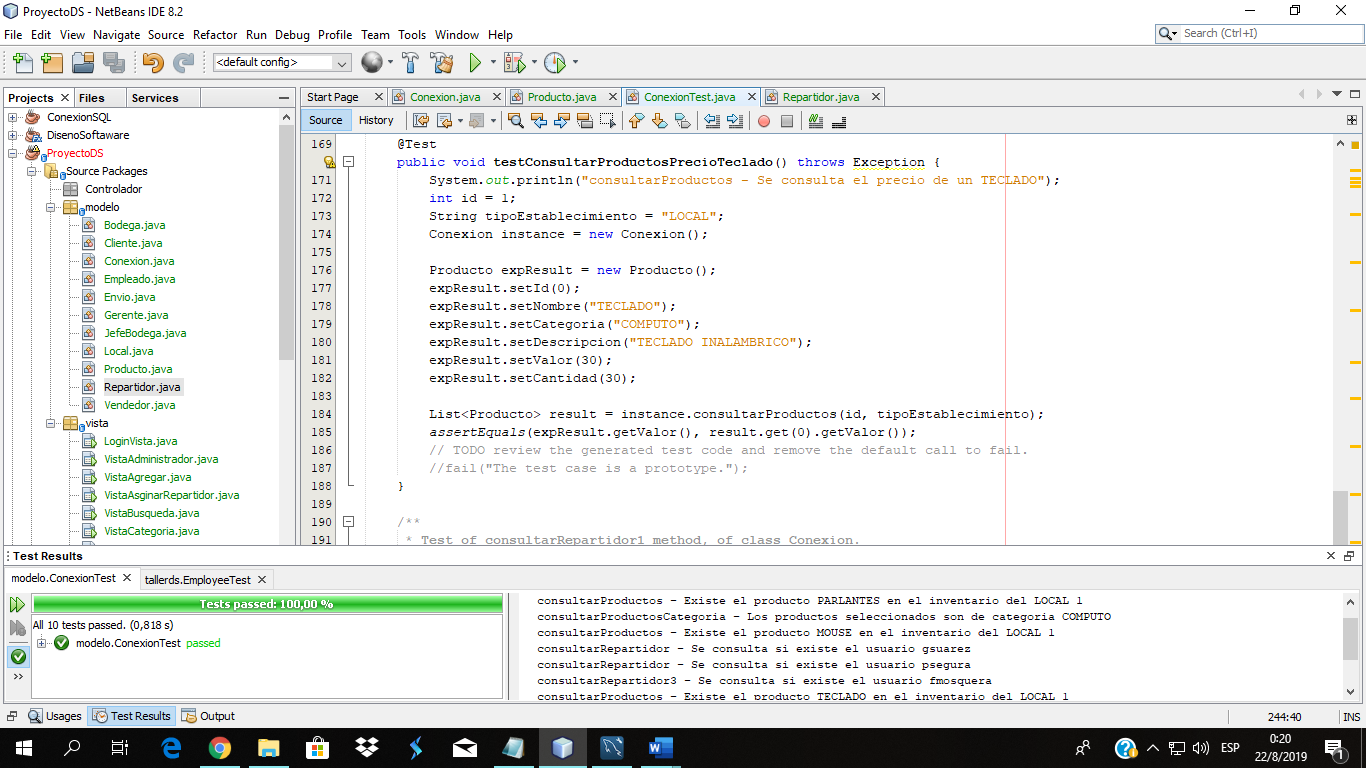


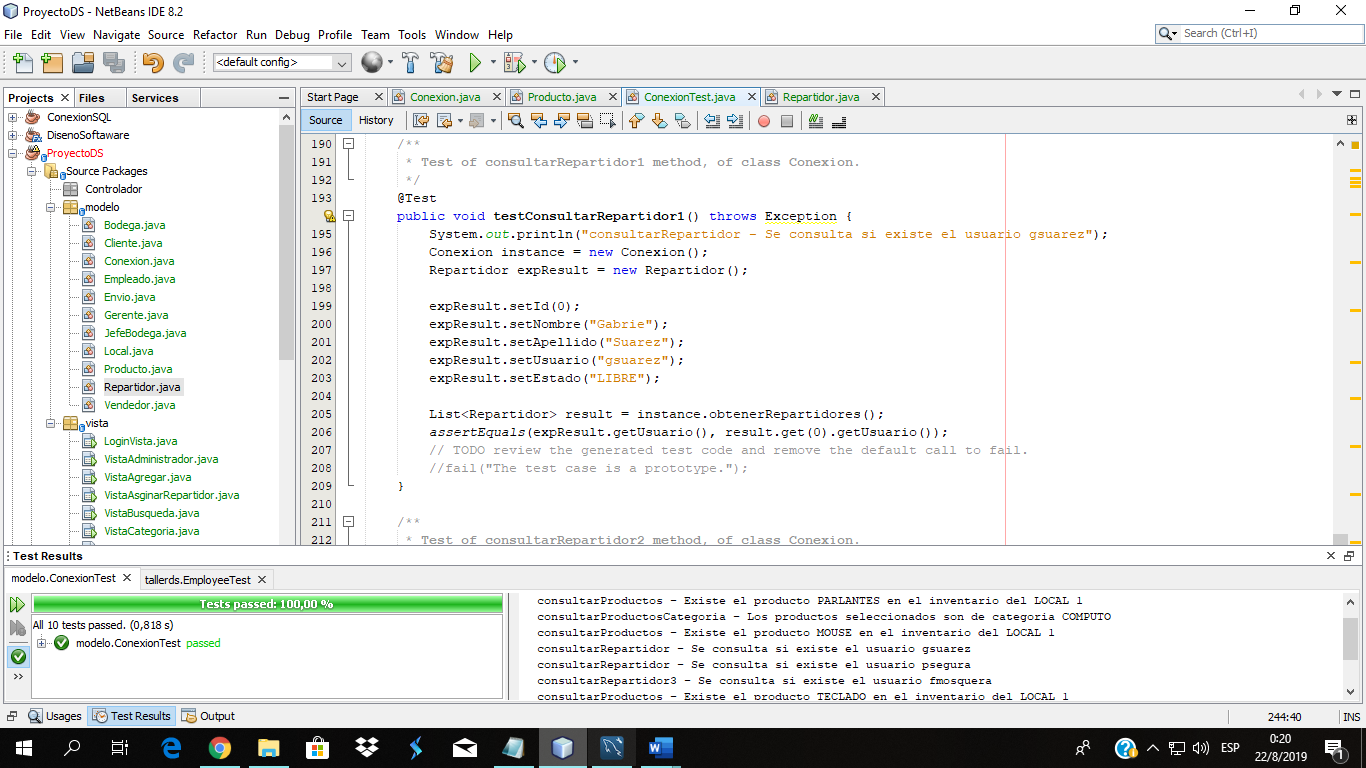


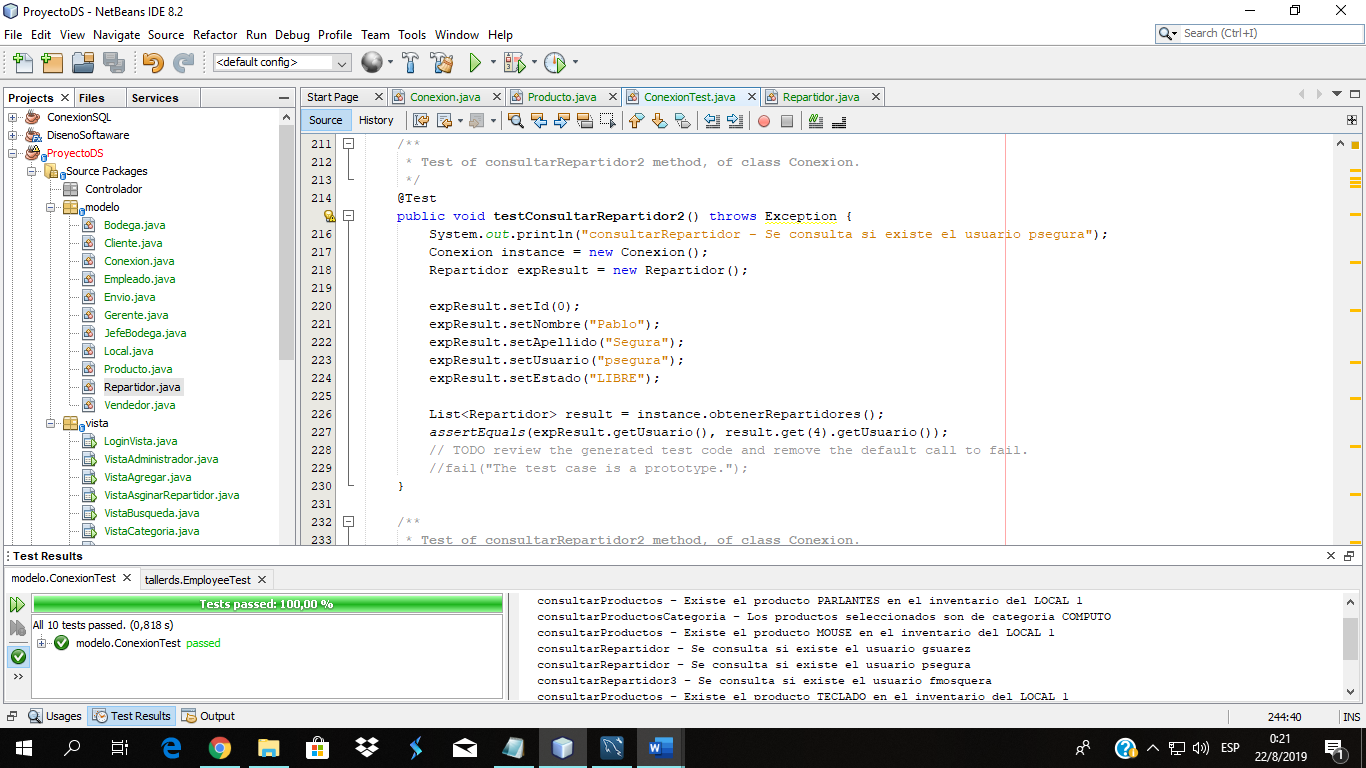


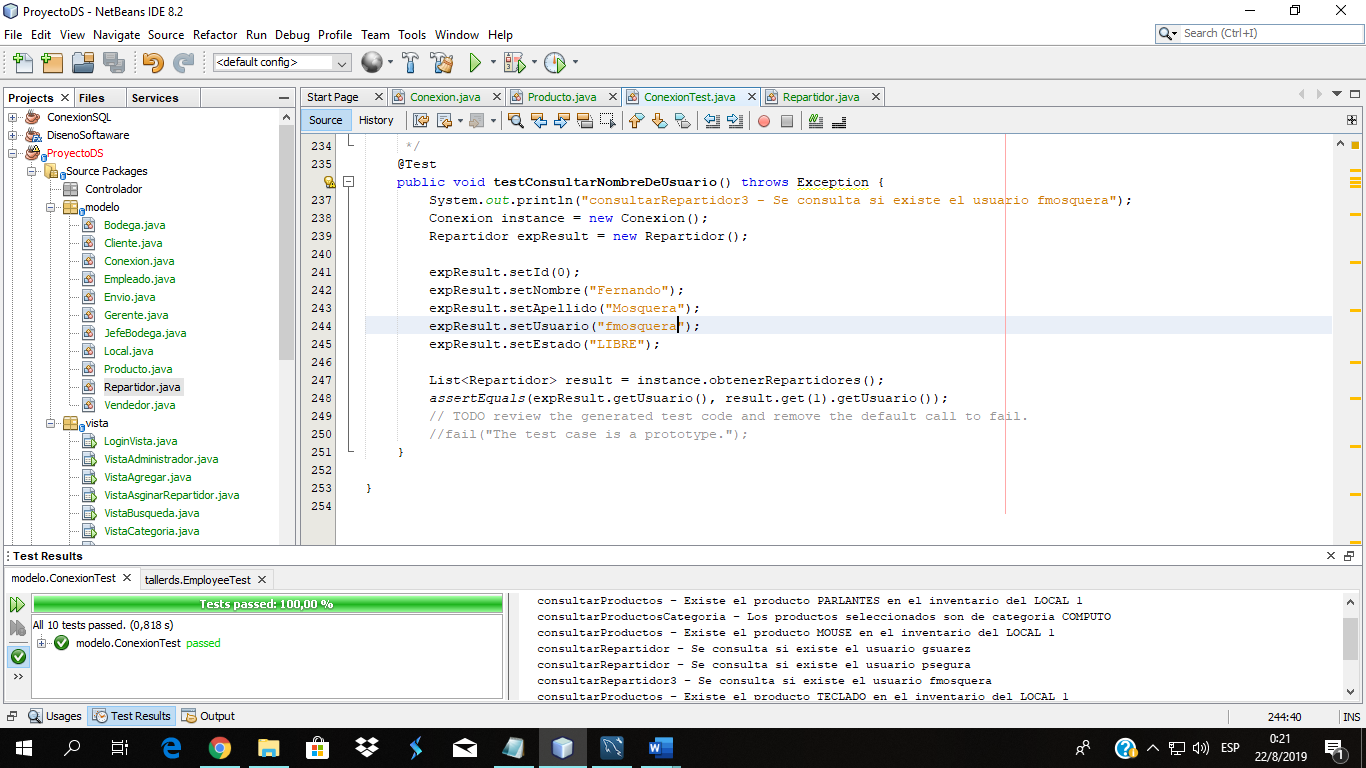








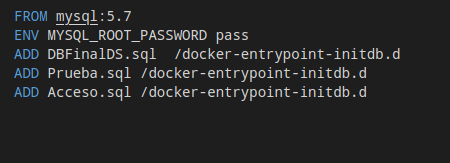




# Dockers

MySQL con Docker

Crear la imagen del contenedor con la base de datos y insertar datos de prueba.



Creamos la imagen a travez de un Dockerfile.

Nos ubicamos en el directorio del Dockerfile.



Creamos el contenedor



**docker run** corre/crea un contenedor.

**-d**indica que el contenedor permanecerá corriendo en segundo plano.

**-p 33061:3306**conecta el puerto 33061 de nuestro sistema operativo con el puerto 3306 del contenedor.

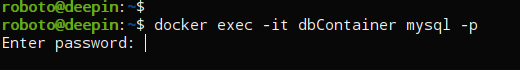
**--name dbcontainer** le da un nombre personalizado al contenedor.

**dbimagen**  indica la imagen a partir de la cual se va a crear el contenedor, después de los dos puntos se indica la versión específica de la imagen.

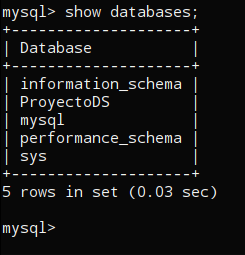
Verificamos que el contenedor este corriendo



Ingresamos al contenedor con la clave previamente establecidad.



Verificamos que la DataBase se a creado.



Ahora en la clase conexion agregamos los atributos adecuados.



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