Basándose sobre el modelo que se ha estado trabajando de registro de Facturas y canjes de medicamentos para una compañía farmacéutica realice los cambios necesarios en el modelo y en el proyecto suministrado para incorporar lo siguiente:

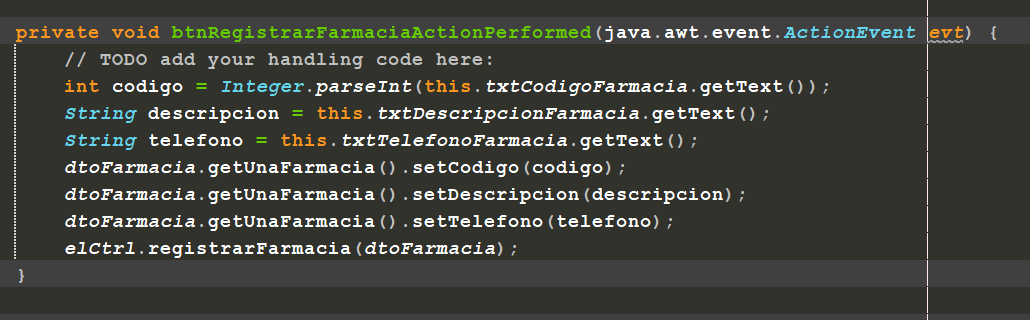
1. En el módulo de mantenimientos de farmacias que se pueda agregar y recuperar la lista de farmacias de manera similar a la que se realizó con el mantenimiento de los productos. En el caso de la gestión de productos, se declaró un DTO propiedad del controlador que la vista se encargó de llenar y el método que se utilizó para el registro se encargó de tomar los datos internos del DTO.

A. ¿Cómo sería la programación si el DTO fuera declarado por la vista y más bien se lo envía al controlador por parámetro para su uso?

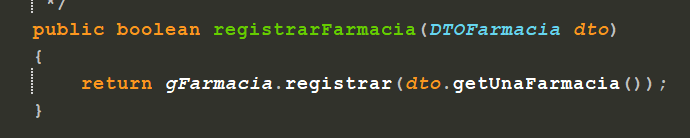
Pruebe este cambio en el mantenimiento de farmacias y muestre evidencia de los resultados.

Evidencia registrar farmacia

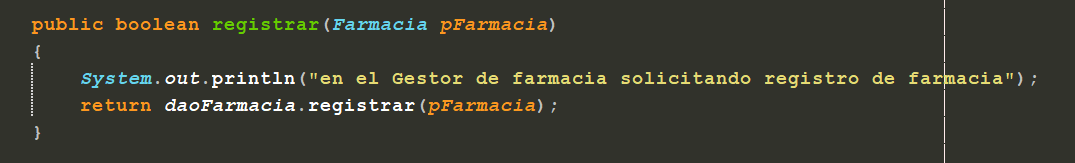
GUI



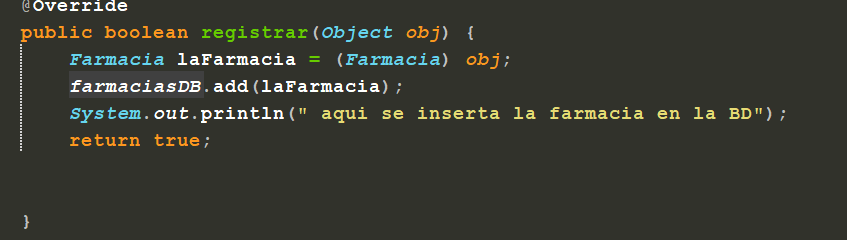
Controlador



GestorFarmacia

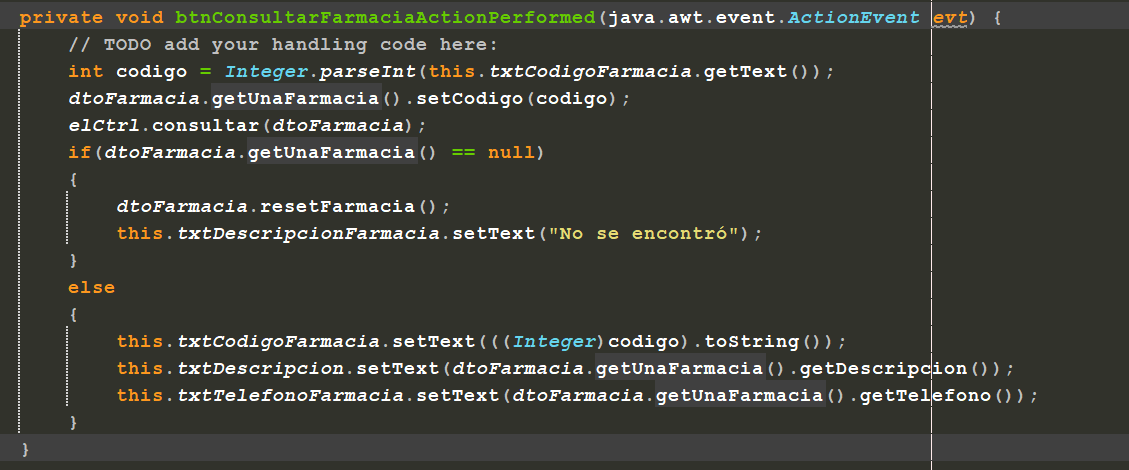


DAO Farmacia

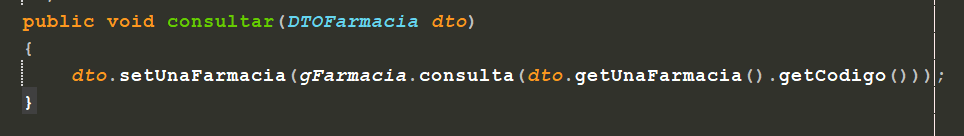


Evidencia consultar farmacia

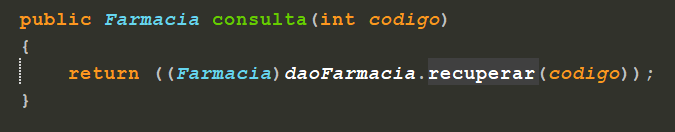
GUI



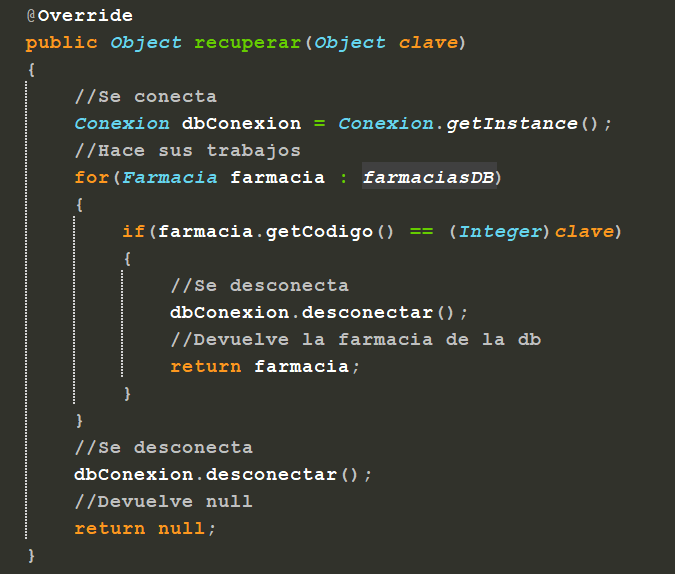
Controlador



GestorFarmacia



DAO Farmacia



2. Diseñe una vista sencilla para el registro de un cliente y conéctela con el controlador para solicitar el registro de un cliente en el sistema.

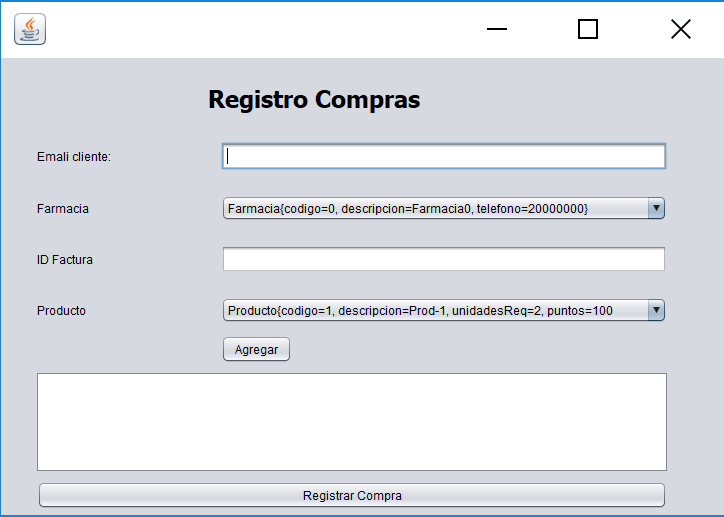
1. ¿Se justifica la presencia de un DTO particular para la captura de los datos del cliente, si así fuera que estructura tendría este patrón complementario?
2. Incluya la programación de registro de un cliente utilizando su propuesta de diseño.
3. ¿Qué propone para que ambas vistas puedan acceder a un único controlador? Justifique su respuesta y refleje en el modelo y en el código su propuesta de cambio para garantizar el cumplimiento de este objetivo.

*Aporte en esta área del documento su propuesta de cambio, su justificación y los screenshots de los trozos de código generados para demostrar la implementación a nivel de modelo y código de cumplimiento de este objetivo.*

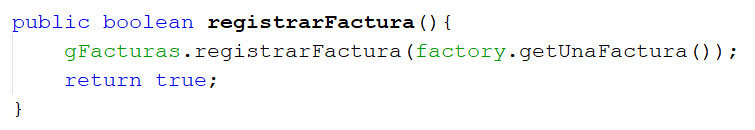
*Muestre además los screenshots necesarios para ilustrar el flujo de conexiones desde la vista de registro del cliente hasta su registro a traves del DAO correspondiente. Recuerde que estamos usando el DAO como elemento de recuperación de los registros que simulan estar persistidos en una BD.*

3. En caso de registrar una compra de ciertos productos en una farmacia determinada por parte de un cliente:

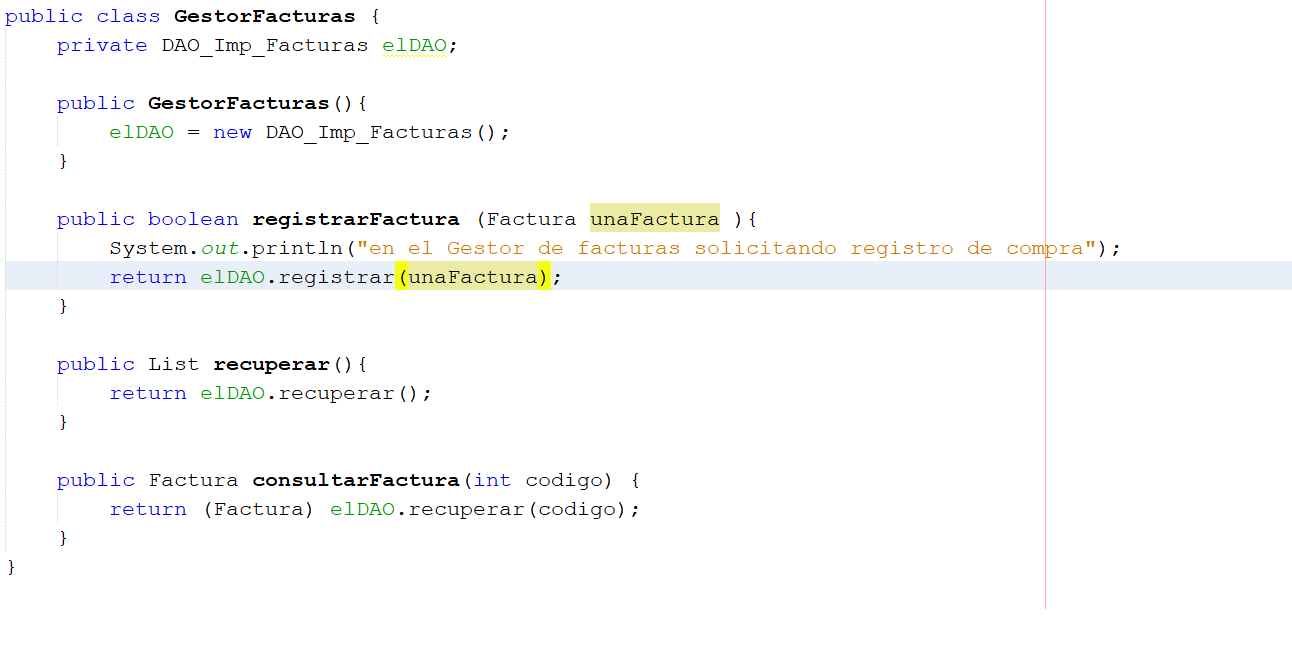
A. Los métodos necesarios para registrar la compra son: recuperar la lista de farmacias y de productos que forman parte del sistema de canjes, para poderlos desplegar en la interfaz gráfica y que el usuario solo seleccione una opción. Además, es necesario un método de registrar compra en el controlador que reciba una factura con toda la información de la compra, le pase esta información a un Gestor de Facturas y que él a su vez, a través del método “registrar” llame al DTO de Factura para registrar la compra en la base de datos.

B.

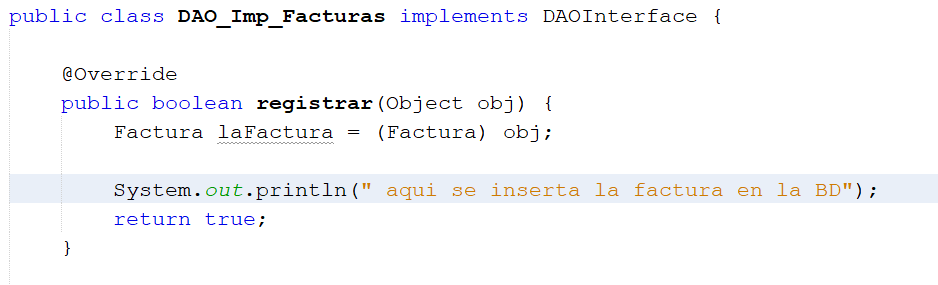
El método en el controlador para registrar la compra:



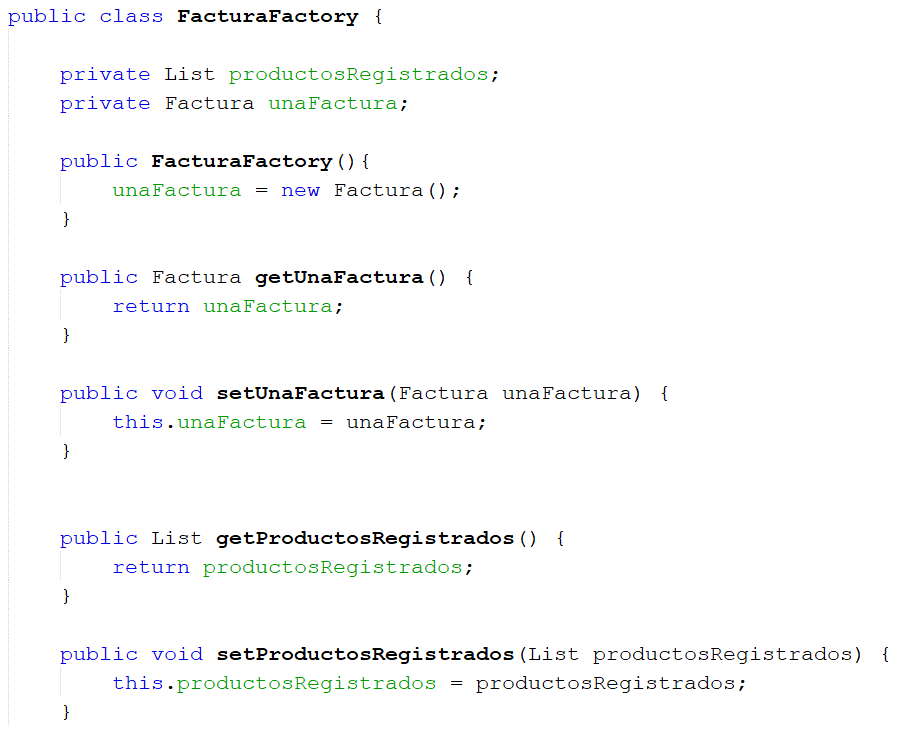
La clase de Gestor de Facturas:



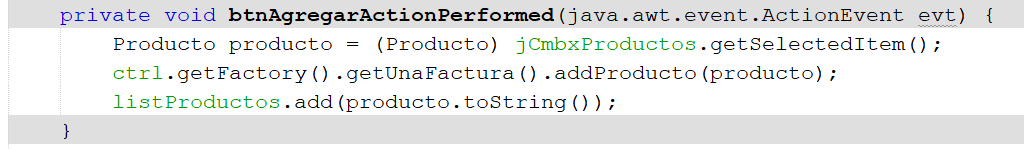
DAO de facturas para registrarlo en la base de datos:



1. Según el principio de Creador de GRASP, la clase encargada de crear una instancia es aquella que tenga la información necesaria para instanciarla. Como el objeto factura es complejo (tiene cliente, farmacia, y una lista de productos), sí se justifica el uso de un creador, pues así se aumenta la claridad del código y facilita la mantenibilidad del programa (si llegara a modificarse algún elemento de esta clase compleja, solo se debe modificar la clase creadora y no todos los lugares donde se pida una instancia nueva).
2. El patrón escogido es un Factory, debido a que se trata de un objeto complejo (contiene una lista de productos, además de otros atributos). No es un Abstract Factory ni otro patrón creacional porque se trata de una única clase, que no forma parte de una relación de herencia.
3. Muestre evidencias de funcionamiento en este apartado de la misma forma que lo ha realizado en los puntos anteriores.



A través de la clase Factura se realizan los sets de los atributos correspondientes. La Vista llama a esta clase para darle los datos recuperados y almacenarlos en un objeto de tipo Factura creado por el Factory. El Controlador después de pide la instancia, como se aprecia en el punto B.



Agregando los productos seleccionados en la vista.

