**

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Proyecto de Diseño de Software

Segundo Avance

Grupo 7

Integrantes:

* Edmundo Alvarez
* Daniel Arroyo
* Oscar Pallazhco
* Patricio Tovar

2018-2S

**Índice**

[Patrones de Diseño 3](#_Toc535184232)

[Diagrama de casos de uso 4](#_Toc535184233)

[Diagramas de la Base de Datos 5](#_Toc535184234)

[Diagrama de clases 6](#_Toc535184235)

Descripción de Proyecto

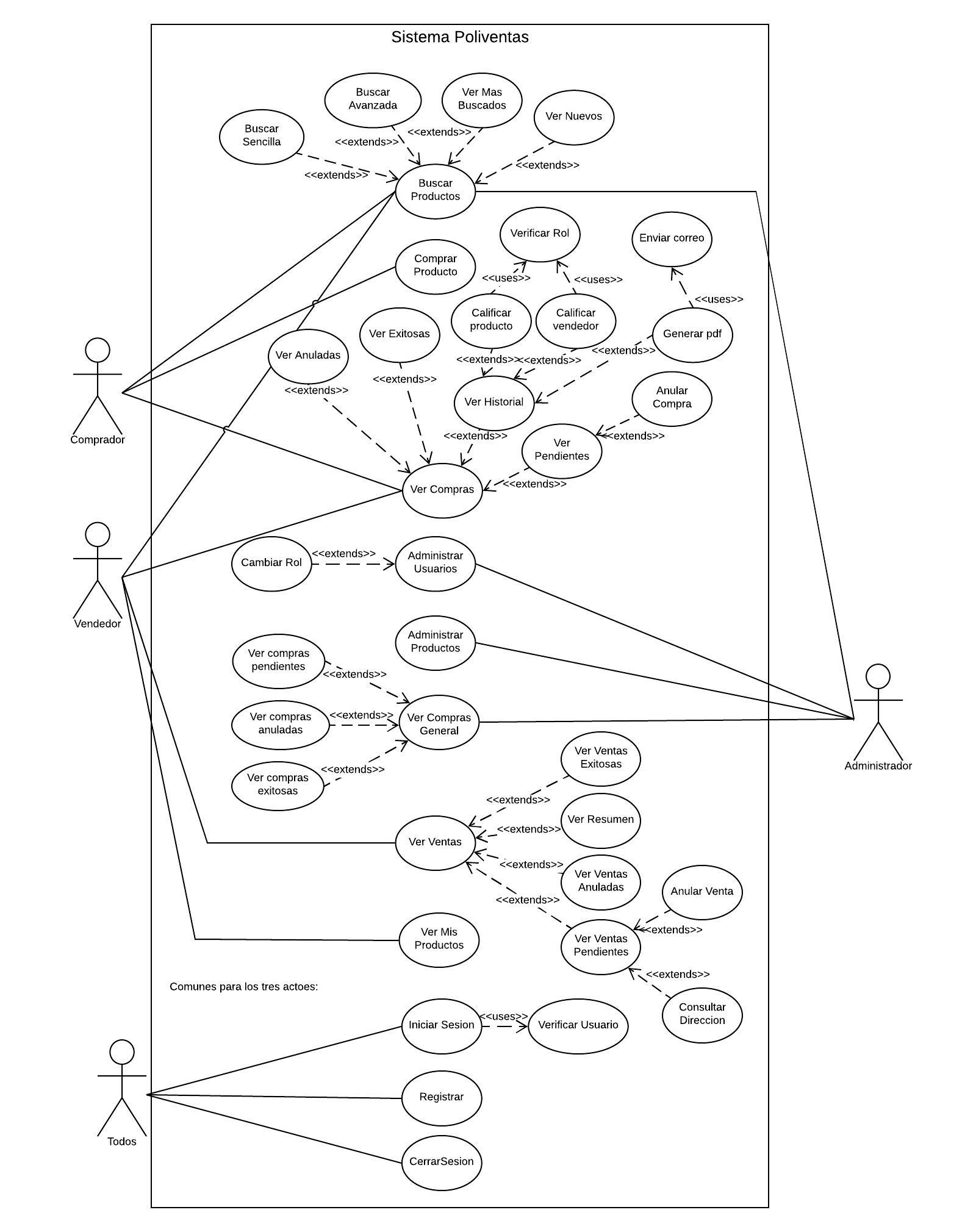
*Este proyecto consiste en un sistema de compra-venta electrónico para estudiantes, en donde ellos puedan vender sus productos y comprar los de otros. Se utilizó Java como lenguaje de programación y FXML para diseño de la interfaz o vistas(prototipo), además de LucidChart para los diagramas, patrones de diseño y MVC(modelo-controlador-vistas) en el diagrama de clases, y conocimiento de bases relacionales para la realización de modelos conceptuales y lógicos.*

# Patrones de Diseño

En este sistema se utilizaron los siguientes patrones de diseño:

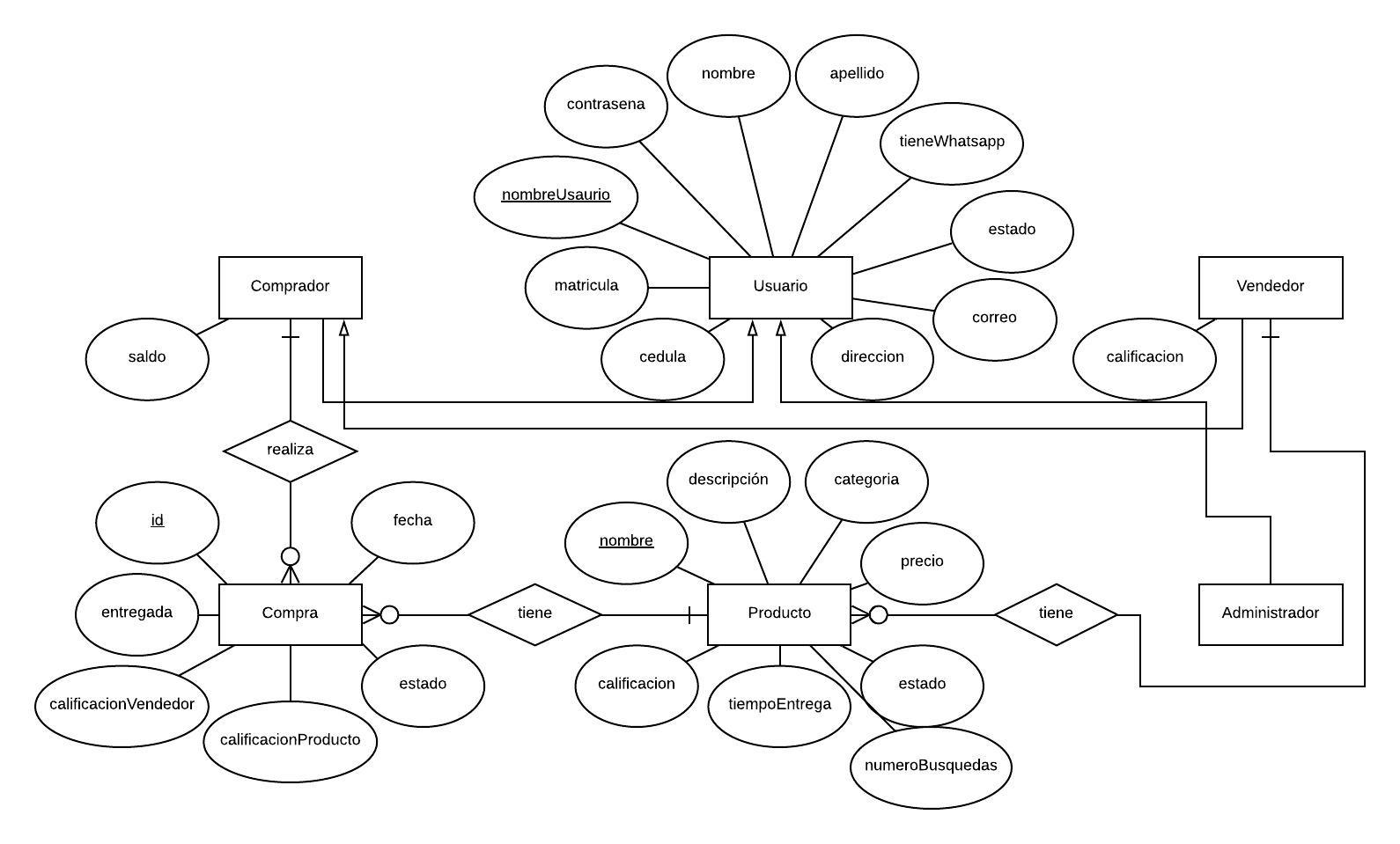
* Singleton: La conexión a la Base de Datos debe ser accedida por varias clases del sistema, por lo que se crea una instancia global de esta utilizando el patrón Singleton. El resultado de esto es la clase SingleConnectionBD.
* Strategy: Para implementar la funcionalidad de poder pagar con tarjeta o por medio de la aplicación se utilizó el patrón Strategy, de manera que la interfaz StrategyPago permite ser implementada por ambos métodos de pago (PagoAplicacionStrategy y PagoEfectivoStrategy) e incluso se puedan agregar nuevos métodos de pago en un futuro.
* Facade: Para reducir el acceso a todas las funcionalidades del sistema se utiliza este patrón, de manera que se crean clases Vistas que muestran solamente las funciones que puede realizar cada tipo de usuario.
* Observer: Como los vendedores deben ser notificados cuando una nueva compra de sus productos sea pagada para poder establecer el lugar de reunión con el comprador, es necesario que una clase esté atenta a este evento y envíe la notificación. Así, la clase Compra (publicador) notifica al vendedor (suscriptor único) del producto que fue comprado cuando cambie su estado de pago por medio del método setearPagado que usa a su vez el método notificarVendedor. En este caso, no es necesaria una lista de suscriptores pues se notifica solo a un vendedor, ni un suscriptor abstracto, pues los únicos suscriptores serán vendedores concretos.

# Diagrama de casos de uso

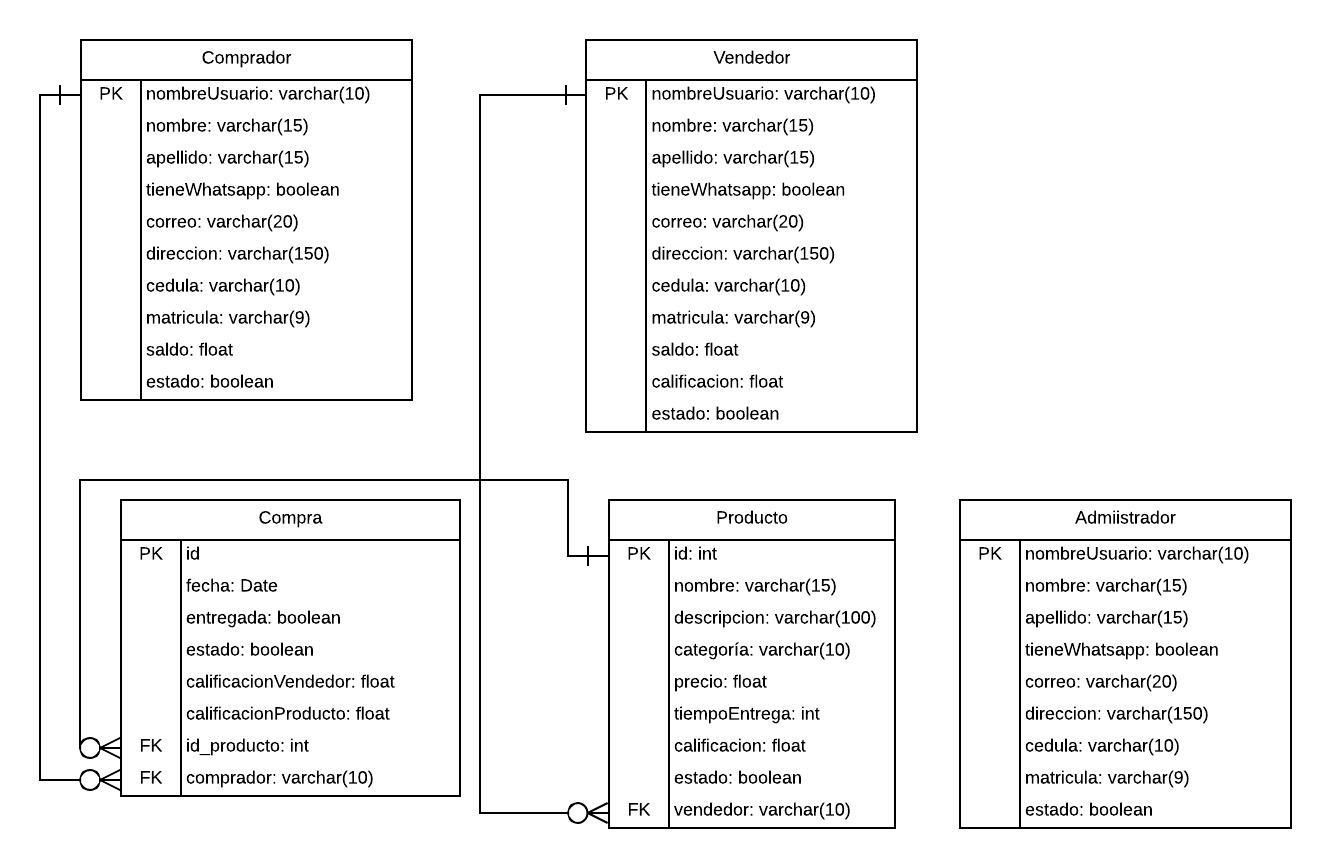


# Diagramas de la Base de Datos

Modelo Conceptual



Modelo Lógico



# Diagrama de clases

