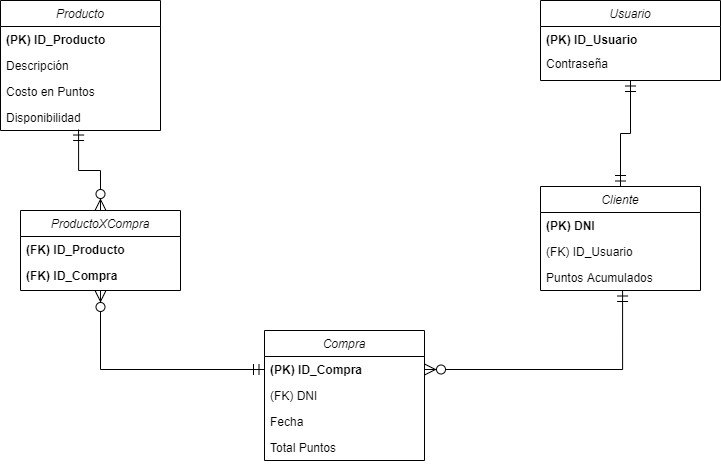
Antonio Romero

# TP Integrador – Programación con Objetos Avanzada

1.- Modelo de Datos

Las entidades del sistema son:

* Usuario: La entidad usuario contiene el ID y contraseña del usuario, su principal función es almacenar los datos de log-in del usuario de cada cliente.
* Cliente: La entidad cliente contiene la cantidad de puntos que ha acumulado el cliente, datos necesarios para la facturación y el envió de los productos que canjee.
* Compra: La entidad compra se compone del total de puntos de la compra, la fecha y se enlaza con la lista de productos que está comprando el cliente.
* ProductoXCompra: La entidad ProductoXCompra cumple la función de enlace entre la tabla Producto y Compra debido a la relación de Muchos a Muchos que existe entre estas 2 entidades.
* Producto: Contiene los datos del producto, principalmente su descripción, disponibilidad y su costo en puntos.

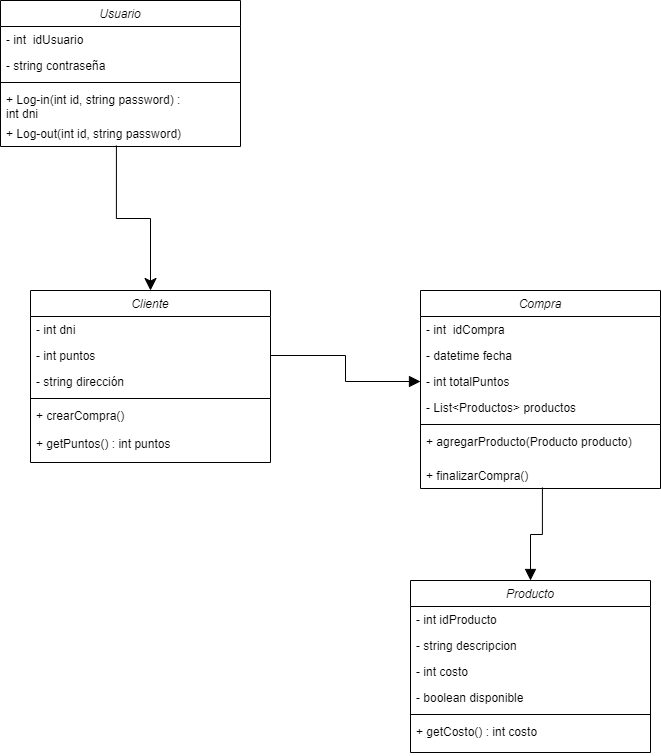


Vamos a utilizar a **Microsoft SQL Server** para la base de datos ya que la aplicación va a tener mayormente un enfoque transaccional donde las transacciones pueden ser abortadas cuando el cliente intenta canjear un producto que no está disponible o que no tiene los puntos necesarios para canjear el producto, por lo que nos beneficia una base de datos relacional.

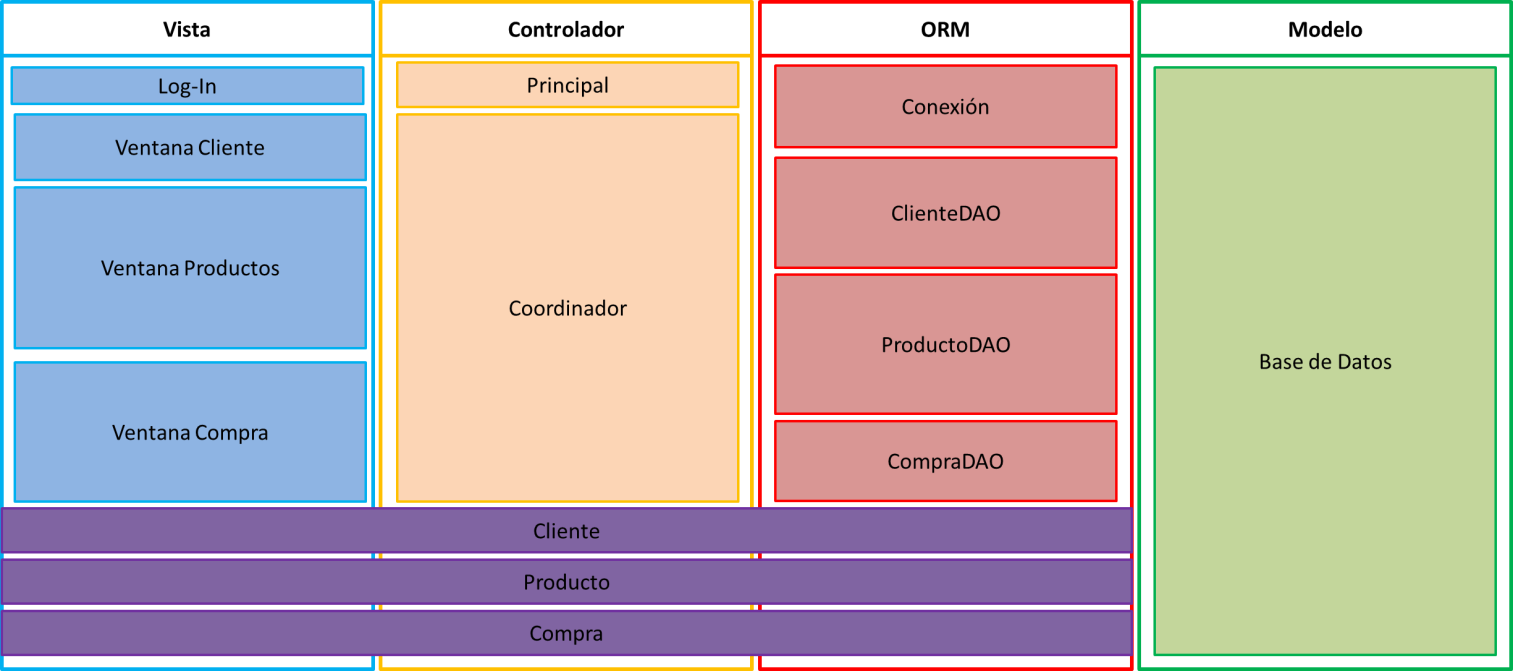
2.- Modelo de clases

Las clases principales a utilizar:

* Usuario: Almacena los datos ingresados por el usuario para hacer el Log-in y Log-out. También se encarga de la validación de ingreso de sesión. Es la primera clase en usarse y su ciclo de vida es tan largo como el proceso.
* Cliente: Almacena los datos del cliente, principalmente para llevar cuenta de la cantidad de puntos y sirve para cargar nuevos canjes al sistema.
* Compra: Lleva el detalle de los productos que se están canjeando y se encarga de validar que los productos que se están intentando canjear no superan la cantidad de puntos que tiene el cliente.
* Producto: Contiene los datos de los productos, principalmente el coste en puntos y si el producto está disponible para ser canjeado.



3.- Patrón de diseño



El patrón de diseño que vamos a utilizar es MVC compuesto por 4 capaz, la capa de Vista, la capa Controlador, la capa de ORM y la capa Modelo.

* La capa vista se compone de las ventanas que van a mostrar la información del cliente y la lista de productos que se pueden canjear.
* La capa controlador se encargará de manejar la lógica del negocio. Su principal función es verificar que el usuario que esta logueando existe en la base de datos y coincide con un cliente existente, verificar que los productos que se estén canjeando no excedan la cantidad de puntos que tiene el cliente y verificar que el producto que se está intentando canjear está disponible.
* La capa de ORM se encargara de hacer las solicitudes de datos desde la aplicación hacia la base de datos, existiendo una clase para el acceso de cada una de las entidades clave.
* La capa modelo almacena todos los datos de los usuarios, clientes, productos y las canjes que se han hecho a través de la aplicación.

A lo largo de las capas Vista, Controlador y ORM se utilizaran las clases Cliente, producto y compra que servirán para manejar la información de dichas entidades, mientras que en la capa Modelo se dispondrá únicamente de la base de datos.

La tecnología a utilizar será Java y el ORM será Hibernate.

4.-Implementar Inyección de dependencias.

La clase Compra tiene un listado de productos donde para añadir productos se ingresa el objeto producto sin dar datos de la clase (salvo el métodos getCosto() para obtener el coste en puntos del producto)