

Una empresa desea diseñar una aplicación para competir con plataformas como Steam y Origin pero de videojuegos. Inicialmente se cuenta con una biblioteca reducida, pero se espera agregar más títulos en el futuro. La aplicación conoce todos los juegos que ofrece, así como a los usuarios registrados, que son personas, y las ventas realizadas.

Del juego se conoce el título, el desarrollador (solo el nombre de la empresa), el precio base y una edad sugerida.

Los usuarios se componen de los datos básicos de una persona más un nombre de usuario específico para su uso en la aplicación y una colección de juegos. Pueden ser Plata, Oro o Diamante, lo que involucra descuentos al comprar juegos según la siguiente tabla:

Nivel	Descuento
Plata	5%
Oro	10%
Diamante	15%

Una venta está compuesta por una fecha, uno o más juegos, un usuario, un medio de pago y un monto (que se calcula una única vez). Los pagos por débito no tienen recargo, pero con tarjeta de crédito se adiciona un impuesto del 3% sobre el total.

El sistema debe responder a la siguiente funcionalidad:

- a) **agregarJuego(Juego unJuego)**
“Agrega un nuevo juego a la plataforma”
- b) **agregarVarios(Vector<Juego> juegos)**
“Agrega varios juegos nuevos a la plataforma al mismo tiempo”
- c) **nuevaVenta(Vector<Juego> juegos, Usuario usuario, Pago medioDePago)**
“Realiza una nueva venta de los juegos recibidos al Usuario pasado como parámetro con determinado medio de pago y la registra en el sistema. Los juegos especificados pasan a ser parte de la biblioteca del usuario.”
- d) **montoTotalVentas()**
“Retorna el monto total de las ventas realizadas hasta el momento”
- e) **promedioDeJuegos()**
“Retorna el promedio de juegos en la biblioteca de los jugadores”

Se solicita:

1. Diseñar una solución en UML que contenga todas las clases, relaciones, atributos y métodos necesarios.
2. Implementar la solución completa propuesta en el punto 1 en JavaLike.