

Carrera: Licenciatura en Sistemas

Materia: Orientación a Objetos II

Equipo docente:

Titular:	Lic. María Alejandra Vranić	alejandravranic@gmail.com
Ayudantes:	Lic. Gustavo Siciliano	gussiciliano@gmail.com
	Lic. Romina Mansilla	romina.e.mansilla@gmail.com

Año: 2020 Turno: Noche



Sistema de democratización de stock

(versión web con bases de datos)

En OO1 se nos solicitó llevar adelante el análisis, diseño e implementación de un software que democratice el stock de sucursales de venta. Este desarrollo se realizó con una aplicación en el paradigma orientado a objetos, escrito en Java, que contaba con tests Java manuales para verificar el funcionamiento del software. Ahora en OO2 se va retomar dicho sistema, agregando las capas explicadas en clase, una base de datos para las entidades, un ORM para manejar la persistencia y una interfaz web para la interacción con el usuario.

Repaso del contexto explicado en OO1

Como se mencionó antes, la finalidad del software es democratizar el stock de sucursales de venta de un cliente, de forma tal que cuando una persona vaya a una sucursal a comprar un producto, y ese local no cuente con el stock, el vendedor pueda usar el sistema para vender el producto usando el stock de otro local.

De los locales del cliente conocemos su ubicación (dirección, latitud y longitud), un código que los identifica, un número de teléfono y la persona que es el gerente de dicho local. De cada empleado conocemos su tipo (gerente o vendedor), su nombre y apellido, su fecha de nacimiento, su DNI y su franja horaria de trabajo. De cada producto conocemos su descripción, su precio unitario y su fecha de alta. Finalmente es importante saber que cada local cuenta con su propio stock de cada producto, haciendo una distinción en lotes. Cada lote es un ingreso de X cantidad de un producto al local, de forma tal que sabemos la cantidad, la fecha de ingreso y el producto en cuestión. De esta forma el stock de cada local se compone de la sumatoria de lotes de stock de cada producto. De los cliente conocemos su nombre y apellido, su fecha de nacimiento, su DNI y su mail.

El cliente llega a nosotros para resolver el problema del quiebre, que se da cuando un comprador quiere adquirir un producto en un local que es sucursal suya y no realiza la compra porque no hay stock en dicho lugar, pero en si, nuestro cliente pierde una venta potencial que puede satisfacer con el stock de otro local que también sea sucursal suya.

Para esto nuestro sistema debe permitir validar si hay stock del producto en cuestión en el local donde se está usando, si hay, realizar la compra. Sino, debe validar con el stock de las tiendas que estén más cerca hasta que encuentre una que lo pueda satisfacer, en ese caso, deberá acordar de palabra con el comprador si desea esperar la entrega del producto teniendo en cuenta la distancia del local en cuestión, si acepta se avanza con la compra y sino se cancela. Es importante destacar que un vendedor del local acciona este pedido y un vendedor del local con el stock lo debe aceptar, de forma tal que esté de acuerdo con ceder su stock a otra sucursal.

Finalmente, un punto importante a resaltar, es que cada vendedor tiene un sueldo básico y un plus que gana de forma dinámica en cada mes por todas sus ventas (que equivale a un %5 del precio del producto por unidad vendida). En el caso de las ventas que se hacen entre locales, al vendedor que acepta ceder su stock, le llega un 2% de la venta y al vendedor de la sucursal un 3%.

Para saber la distancia entre dos locales vamos a usar como input la latitud y longitud como se muestra en el siguiente método de ejemplo:

```
public static double distanciaCoord(double lat1, double lng1, double lat2, double lng2) {  
    double radioTierra = 6371; //en kilómetros  
    double dLat = Math.toRadians(lat2 - lat1);  
    double dLng = Math.toRadians(lng2 - lng1);  
    double sindLat = Math.sin(dLat / 2);  
    double sindLng = Math.sin(dLng / 2);  
    double va1 = Math.pow(sindLat, 2)  
        + Math.pow(sindLng, 2) * Math.cos(Math.toRadians(lat1)) * Math.cos(Math.toRadians(lat2));  
    double va2 = 2 * Math.atan2(Math.sqrt(va1), Math.sqrt(1 - va1));  
    return radioTierra * va2;  
}
```

Puntos que quedan fuera del scope del TPC:

- En la vida real se debería hacer un acuerdo mediante un check que se envía por celular al cliente de “acceso a los términos y condiciones de la venta” pero en este de laboratorio queda afuera).
- Otra opción que puede pasar, es que el stock no se llegue a satisfacer por un local, sino que se haga por varios, dicha opción queda por fuera.
- La entrega y logística del producto, vía moto o auto, quedan por fuera.

Puntos del TPC OO2

Proyecto Cuatrimestral

Grupos: 4 ó 5 estudiantes

Proyecto: Sistema de democratización de stock (versión web con bases de datos)

Herramientas de desarrollo:

IDE: Eclipse (<https://eclipse.org/>)

Diagramas: Dia (<http://live.gnome.org/Dia>)

Implementación:

Paradigma: Orientado a Objetos.

BD: MySQL

ORM: Hibernate

Tecnologías: Java con Servlets

Resolución: Implementación de la BD y su conexión, más una interfaz web para probar el software.

El proyecto deberá cumplir con los siguientes casos de uso:

(*) Nivel mínimo para aprobar el Proyecto Cuatrimestral.

1. Alta, baja y modificación de locales.
2. Alta, baja y modificación de empleados.
3. Alta, baja y modificación de clientes.
4. **Alta, baja (lógica y física solo si no tiene dependencias) y modificación productos. (*)**
5. Alta y consumo de stock.
6. Calculo de distancia entre locales.
7. **Validación de stock y posibilidades de locales a solicitar stock (si corresponde). (*)**
8. **Generación de solicitud de uso de stock de otro local. (*)**
9. **Aceptar o rechazar la solicitud de stock. (*)**
10. **Generación del pedido con stock propio. (*)**
11. **Generación del pedido con stock de otro local. (*)**
12. Generar factura.

13. Cierre del mes para definir el sueldo de los vendedores.
14. Emitir reporte de productos vendidos entre fechas por local.
15. Emitir ranking de producto más vendido.

En la presentación se tendrán que mostrar los flujos de usuario pensados para testear cada caso.

Documentación a presentar:

1. Listado de casos de uso (aproximadamente entre 30 y 40 CU).
2. Implementación.

Fechas de hitos y presentaciones:

Fechas a indicar y comunicar por la cátedra por intermedio del aula virtual.