**Escuela Superior Politécnica del Litoral**

**Programación Orientada a Objetos**

**Proyecto Primer Parcial**

**202O – 2T**

**Objetivos:**

Desarrollar en los estudiantes las habilidades de la programación orientada a objetos básica incluyendo la aplicación de los conceptos de manejo de clases y sus relaciones, paquetes, atributos y métodos de instancia y de clase, manejo de listas, visibilidad de atributos y métodos, entrada por teclado y salida por consola

**Descripción:**

Lo han contratado a usted y a sus compañeros para crear un sistema que permita a los pequeños agricultores vender sus productos evitando intermediarios.

Los proveedores que deseen vender sus productos a través de la plataforma deben primero registrarse en la misma. Una vez registrados pueden ingresar información de sus productos, realizar consultas de los pedidos realizados y gestionar el estado de los pedidos.

Los clientes registrados en la plataforma pueden consultar información de los productos, realizar pedidos usando distintos medios de pago y consultar el estado de sus pedidos.

A los clientes solo se le deben mostrar la información de los productos de proveedores que vivan en un radio de 50 km de distancia.

Al inicio la aplicación muestra un menú numérico con las siguientes opciones:

1. Registro
2. Iniciar Sesión
3. Salir

* **Registro**

La plataforma debe permitir el auto-registro de los proveedores y compradores

De los proveedores se pide registrar su nombre, número identificación, dirección, ubicación (coordenadas lat y longitud), número contacto, correo electrónico.

De los compradores se quiere registrar nombre, número de identificación, dirección, (coordenadas lat y longitud), correo electrónico y los datos de su forma de pago preferida.

Todos los usuarios deben de tener un nombre de usuario único que los identifique en la plataforma y una clave asociada al nombre de usuario para el ingreso a la misma.

* **Iniciar sesión**

Para poder usar la plataforma los usuarios deben iniciar sesión con su nombre de usuario y contraseña. Si las credenciales ingresadas son válidas se debe mostrar un menú numérico con las opciones que cada usuario puede realizar.

**La plataforma muestra distintas opciones de acuerdo al tipo de usuario.**

* **Opciones Proveedores**

Dentro de la plataforma los proveedores deben poder registrar los productos que ellos van a ofrecer. De cada producto ellos deben registrar un código, nombre, breve descripción, categorías (un producto puede tener una o varias categorías) y precio. Las categorías de los productos son: cárnicos, vegetales, frutas, lácteos y conservas.

Los proveedores deben poder consultar la información de los pedidos que les han hecho y gestionarlos. De cada pedido se muestra su código, fecha en el que se realizó, productos que se pidieron (códigos, nombres, cantidades), datos del cliente (nombre, dirección, correo de contacto), datos del método de pago y el total del pedido.

Los proveedores deben poder gestionar el estado de los pedidos. Inicialmente un pedido tiene un estado de solicitado. El proveedor debe poder cambiar su estado a procesando cuando se está armando el pedido y ha despachado cuando se procedió a entregar el pedido. Se debe guardar un registro de en qué fecha se hizo cada cambio de estado.

Los proveedores además deben ser capaces de consultar y editar la información de sus productos registrados filtrando por categoría y nombre (si se deja vacío los filtros no se debe filtrar por los mismos). De cada producto se debe mostrar el código, nombre y categoría. A continuación, la plataforma debe pedir el código del producto que desea editar. Se puede editar todos los campos a excepción del código.

* **Opciones Compradores**

Cuando el usuario ingresa a la plataforma este puede consultar información de los productos que se ofrecen en la misma cuyos proveedores se encuentran hasta 50 km de distancia del usuario. Para ello el usuario puede filtrar por la categoría y nombre (completo o parcial) y rango de precio del producto. Ninguno de los filtros es obligatorio. Se puede filtrar por más de un filtro a la vez. En la consulta se deben mostrar todos los artículos que coincidan con el criterio de búsqueda. De cada artículo se muestra código, nombre, categoría y precio.

Después de mostrar los resultados de búsqueda el sistema debe permitir al usuario ingresar el código del producto que desea consultar o realizar otra búsqueda. Cuando el usuario ingresa el código del producto en el que está interesado aparece la información detallada del mismo (código, nombre y precio). A continuación, el sistema debe preguntar al usuario si desea agregar el producto al carrito de compras. Si el usuario ingresa que si se debe preguntar cuántos artículos quiere adquirir y a continuación se debe guardar la información del producto en el carrito de compra y regresar al usuario al menú principal.

Debe haber una opción desde la cual el usuario pueda consultar los productos en el carrito de compras. Al ingresar a esta opción se debe mostrar la lista de los productos en el carrito (código, nombre, cantidad, precio unitario, subtotal) y el total a pagar. A continuación, el programa debe preguntar al usuario si quiere eliminar un producto del carrito o quiere efectuar la compra de los productos del carrito.

Si el usuario ingresa eliminar producto se debe pedir el código del producto a eliminar del carro, el programa remueve producto del carrito y vuelve a mostrar los elementos restantes en el carrito.

Si el usuario ingresa la opción comprar se pide al usuario la forma de pago: **tarjeta de crédito** o **paypal**. Para el caso de pago con tarjeta se debe registrar el tipo de tarjeta (visa, diners, etc), número de la tarjeta y nombre del titular de la misma. Para el caso de pago con **paypal** se debe registrar el nombre de usuario y contraseña de paypal. (Ver especificación de cómo se procesa el pago en requerimientos finales)

Si se procesó el pago correctamente se debe finalizar la compra y generar un pedido con un código único por cada proveedor. Por ejemplo, si en la compra había tres productos de los cuales el producto 1 y producto 2 pertenecían al proveedor 1 y el producto 3 pertenecía al proveedor 3, se debería generar dos pedidos: uno con los productos del proveedor 1 y otro con los productos del proveedor 2. Los pedidos tendrán un estado de procesando. **Además, se debe enviar un correo electrónico con la información de la compra al cliente.**

Los compradores deben ser capaces de consultar los pedidos realizadas. Se debe mostrar el código del pedido, fecha que se realizó el pedido, el monto del pedido, la forma de pago (y si es tarjeta se debe mostrar el número de la tarjeta con la que se hizo el pago), lista de los productos en el pedido y el estado del pedido (solicitado, procesando, entregado).

**REQUERIMIENTOS FINALES**

1. Cuando se pide la ubicación de los proveedores y compradores estos ingresan ubicaciones reales y el programa internamente debe llevar esas direcciones a coordenadas usando alguna librería externa que permita hacer geocoding (ver anexos para ver ejemplos de librerías)
2. Para calcular la distancia entre los clientes y los proveedores de los use la fórmula de use la fórmula de distancia Harvesine (Ver anexos).
3. Cree una clase llamada MétodoPago y dos clases especializadas que hereden de ella llamadas PagoTarjeta y PagoPaypal. La clase base debe tener un método abstracto llamado procesarPago que debe ser sobre-escrito por las clases especializadas. Para la clase PagoTarjeta este método genera un código que le llega al correo del cliente y debe ser ingresado por el cliente para confirmar la validez del mismo. Para la clase PagoPaypal este método genera un valor al azar entre 100 y 1000 que representará la cantidad de dinero que el cliente tiene en su cuenta y se procesará el pago si los fondos son suficientes.
4. Debe haber definido un su sistema una función llamada **inicializarSistema**()que cree al menos 2 proveedores, 3 productos por proveedor y un cliente.
5. SE DARÁ PUNTOS EXTRAS POR AGREGAR FUNCIONALIDAD DE GEOCODING QUE PERMITA QUE EL USUARIO SOLO INGRESE LA DIRECCIÓN Y SE OBTENGAN DE FORMA AUTOMÁTICA LAS COORDENAS

**CONSIDERACIONES GENERALES**

* El proyecto debe ser realizado en grupo de 3 estudiantes.
* Se deben aplicar los Principios de Programación Orientada a Objetos en el diseño de la solución.
* El desarrollo de la solución debe ser en Java 11.
* Realice un diseño en capas, separar la capa de presentación de la capa del modelo.
* Agrupe sus clases en paquetes
* El sistema debe realizar las validaciones respectivas para no permitir el ingreso de tipo de datos que no corresponde con los valores posibles y que no se caiga.
* Si el programa se cae en medio de la revisión este será calificado sobre la mitad de los puntos.
* Todas las opciones del menú deberían tener la posibilidad de regresar hacia atrás.
* Todos los grupos deben usar un sistema de control de versiones para compartir su código con los miembros de su equipo. Use la plataforma bitbucket (<https://bitbucket.org/)> para guardar el repositorio de su proyecto.
* Incluir un archivo README con las credenciales de los usuarios del sistema y las instrucciones que usted crea necesarias para poder usar el sistema.

**ENTREGABLES DEL PROYECTO:**

* .zip con las fuentes de su Proyecto con el nombre proyecto1\_grupoNUMEROGRUPO.
* .jar de su proyecto con el nombre proyecto1\_grupoNUMEROGRUPO.
* .pdf con el informe del proyecto (se envía modelo del informe con lo que debe tener).
* Javadoc del proyecto

**RÚBRICA DEL PROYECTO**

El proyecto tiene un puntaje de **31.25 puntos** del componente teórico del primer parcial de la materia.

Su nota final se calculará de la siguiente forma:

**Puntaje\_Final = Punto\_Proyecto X Sustentacion (0-1) X Participacion (0-1)**

**Evaluación Participación**

La participación se evaluará en base a lo que cada miembro del grupo contribuya a al repositorio de su proyecto.

**Funcionalidad (50 %)**

* Se valorará la apariencia del proyecto, sin que esto implique que el estudiante deba utilizar recursos adicionales a los aprendidos en clase. (El proyecto es en consola)
* Se verificará el cumplimiento de toda la funcionalidad requerida, así como las correctas validaciones de los datos.
* El proyecto debe ser debidamente probado antes de presentarlo al profesor. Por lo tanto Se espera que el programa no se caiga al ejecutarlo, ni tenga un comportamiento no esperado (Si se cae en la presentación, son 5 puntos menos por cada caída)

**Documentación (10 %)**

* Javadoc del proyecto generado.
* Documentación interna del proyecto.

**Abstracción y Uso de Objetos (20 %)**

* Creación de Clases pertinentes con sus propiedades y métodos
* Correcta interacción de objetos. Que los objetos se comuniquen entre si y no solo sean llamados todos en el main.
* Usar recursos de herencia e interfaces para la solución.

**Modularidad y Encapsulamiento (20 %)**

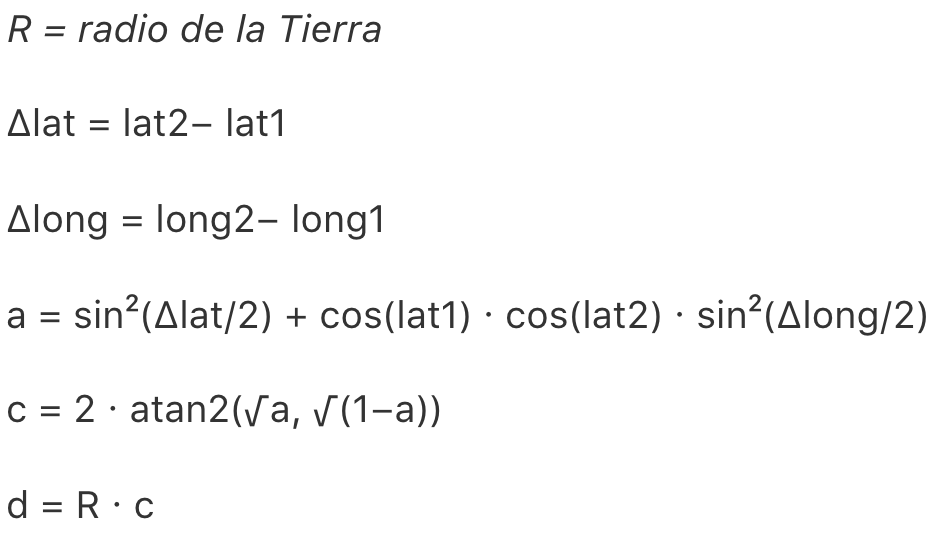
* Dividir el problema usando los métodos correspondientes. No escribir bloques inmensos de código.
* Encapsular correctamente el proyecto. Crear paquetes donde se agrupen clases que se relacionen.
* No olvidar que para acceder a las propiedades de las clases se debe proveer los métodos get y set.

**Penalidades**

* **NO SE REVISARÁ PROYECTOS QUE NO CUMPLAN CON EL 70% DE LA FUNCIONALIDAD**
* 10 porciento menos por no seguir buenas prácticas de programación (convención para nombre de clases, nombre de atributos y métodos, falta de documentación de funciones usando Java Doc, uso de constantes cuando es necesario, o algún fallo grave que se detecte en la forma de programar).

**ANEXOS**

**Fórmula para calcular la distancia Harvesine**



donde

R -> 6378 km

lat2 y long2 -> latitud y longitud del punto de llegada en radianes

lat1 y long1 -> latitud y longitud del punto de partida en radianes

d -> la distancia entre los dos puntos

Nota: Los valores de latitud y longitud dados en la data está en grado centígrados, por lo que los debe transformar en radianes antes de hallar la distancia.

La librería math de Python contiene la funcione geométricas que se usan en esta fórmula.