1. Identificación o establecimiento del problema a resolver.

Se desea implementar un programa que contabilice las ventas y gestione el personal de una zapatería, el programa debe realizar cuatro funciones: **registrar compra** y generar comprobante de pago, **mostrar inventario, actualizar inventario**, **administrar empleados**.

El producto, que son zapatos, se organiza en tres categorías: zapatos de uso casual, zapatos deportivos y zapatos para niños, estos a su vez se dividen en dos subcategorías: zapatos para dama y para caballero. Pueden variar en marca, talla, año y por supuesto, precio.

2. Aplicación del proceso de Análisis del problema, para identificar entradas, procesos y salidas necesarias para la resolución del problema.

En la función de **registrar compra** se debe capturar el código del producto para mapearlo al inventario y conocer su precio. También se debe capturar la clave de empleado que también se mapeará a una lista de empleados. Esta función también generará un comprobante de pago al registrar cada compra.

En la función de **mostrar inventario** se imprimirá en pantalla una lista detallada de todos los productos disponibles en el almacén, esta función es útil para conocer qué productos se encuentran disponibles en la tienda.

La función **actualizar inventario** será llamada cuando la tienda sea surtida, esto es, cuando nuevos productos sean añadidos a la tienda, en cuyo caso es necesario actualizar el inventario.

Por último, la función **administrar empleados** tendrá a su vez tres funciones: **dar de alta/baja a un empleado** y **actualizar lista de empleados** y **aumentar sueldo.** La primera función servirá para actualizar la lista de empleados cuando un nuevo empleado ingrese a trabajar, o cuando un empleado renuncie o sea despedido, para cuyo caso se calcula su indemnización y/o liquidación según sea el caso. La función **actualizar lista de empleados** servirá para actualizar el sueldo de un empleado, así como sus prestaciones y otros datos. **Aumentar sueldo**, como su nombre

lo indica sirve para aumentar el sueldo de un vendedor si este cumple con los requisitos (antigüedad adecuada, buen vendedor, etc).

Para acceder a cada función se requerirá ingresar una clave del empleado y su respectiva contraseña, pues no todos los empleados tienen acceso a todas las funciones.

Dos tablas externas serán necesarias para el uso de esta aplicación: una tabla de todos los productos con sus respectivos códigos, y una tabla de todos los empleados con sus respectivas claves y contraseñas.

3. Aproximación a una solución software.

a). Identificación de clases, atributos y métodos del producto.

Se identifican las siguientes clases:

Zapatería: Esta clase contendrá al método main(), en este método se desplegará el menú.

Zapato: Clase que abstracta tendrá como atributos la talla, el precio, el año, la marca, descripción (color y material) y la categoría (categoría: caballero, dama o niños).

Boleados: Clase que hereda a clase zapato y tiene como atributo el tipo de zapato boleado.

Tenis: Clase que hereda a clase zapatos y tiene como atributo el tipo de tenis.

Deportivos: Clase que hereda a clase zapato, tiene como atributo el tipo de deporte.

Tacón: Clase que hereda a la clase zapato, tiene como atributo el tipo de tacón.

Ballet: Clase que hereda a la clase zapato, tiene como atributo tipo de zapatilla.

Patines: hereda a la clase deportivos y tiene como atributos el tipo de rueda.

Actualizar: interfaz que tiene dos métodos estáticos: agregar() y quitar().

Inventario: Clase que contiene un atributo estático arrayList de tipo zapato para ir guardando los zapatos que hay disponibles, también tiene como atributos la cantidad. Como método estático tiene mostrarInventario(). Esta clase implementa a la interfaz actualizar.

Empleado: Clase abstracta que tiene como atributos el nombre, apellido materno, apellido paterno, número de seguro social, sueldo y antigüedad. Tiene como método estático registrarVenta().

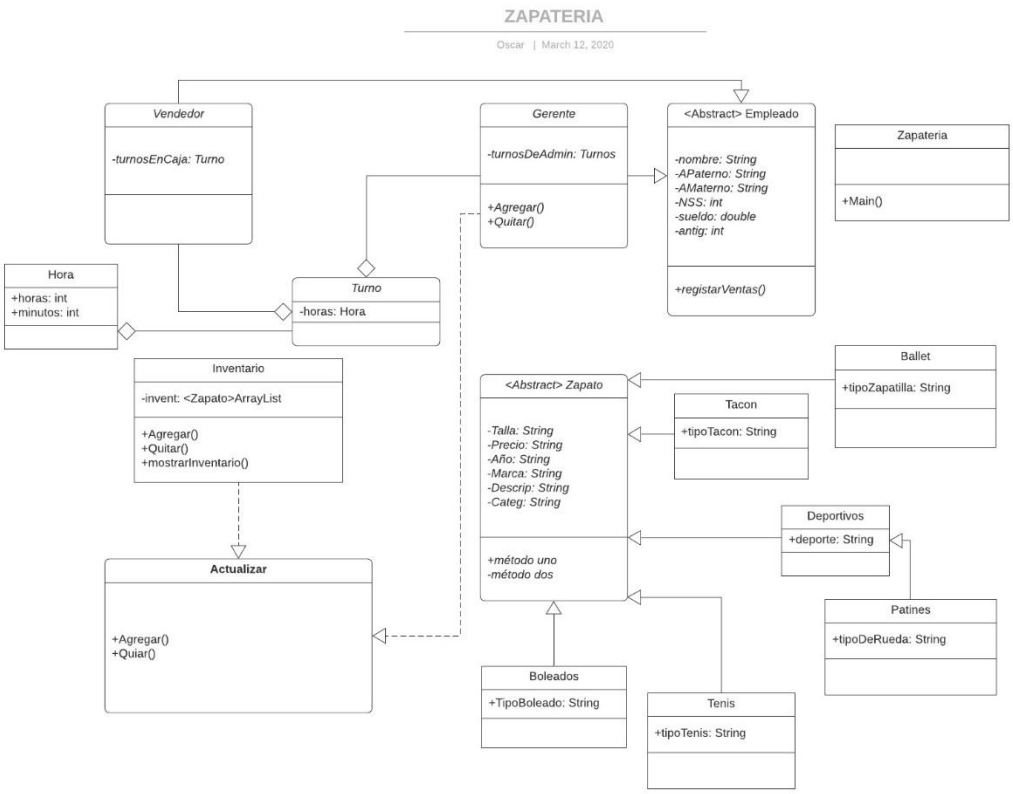
Vendedor: Clase que hereda a empleado y tiene como atributos turnos en la caja.

Gerente: Clase que hereda a Empleado y que tiene como atributos turnos de administración. Implementa a la interfaz actualizar.

Turno: Clase que tiene como atributo un ArrayList de tipo horas.

Horas: Clase que tiene como atributos horas y minutos.

b. Elaboración los diagramas UML (Jerarquía de clases, Jerarquía de las clases detalladas o diagrama de clases)



---

4. Descripción dónde o como implementarían los temas de herencia, polimorfismo y/o interfaces (Documentación basada en el diagrama UML correspondiente).

Vemos que la clase Zapato es heredada por los diferentes tipos de zapato que la tienda ofrece (Tacón, Tenis, Boleados, Patines, Deportivos, Ballet). Además, las clases Gerente y Vendedor heredan a la clase empleado, de ahí se aplicó herencia. Ahora, las clases Empleado e Inventario implementan a la interfaz Actualizar. Por último, nótese que las clases heredadas (Empleado y Zapato) son abstractas (polimorfismo).