



Trabajo Práctico IX: Persistencia

Entrada y Salida de Datos

Serialización

Almacenamiento en archivos

La interfaz serializable

Herencia y serialización

XML

Objetivos de la práctica:

Serializar objetos con atributos primitivos.

Serializar objetos con atributos de tipo serializable.

Serializar objetos no serializables y analizar las posibles soluciones.

Utilizar archivos en formato XML.

Lograr persistencia de sistemas utilizando serialización

Ejercicio 1

La clase **Vehiculo**, posee los atributos N_Chasis (String), anio_fabricacion (int).

La clase **Automovil** se extiende de **Vehiculo**, y agrega los atributos marca, modelo y patente de tipo String. Además tiene un atributo de tipo **Motor**.

La clase **Motor** tiene los atributos n_Serie y combustible de tipo de String.

Se desea lograr la persistencia, tanto Binaria como XML de la clase **Flota** que tiene un nombre y una colección de automóviles. Escribe también una **Prueba** que en su método main para comprobar la funcionalidad.

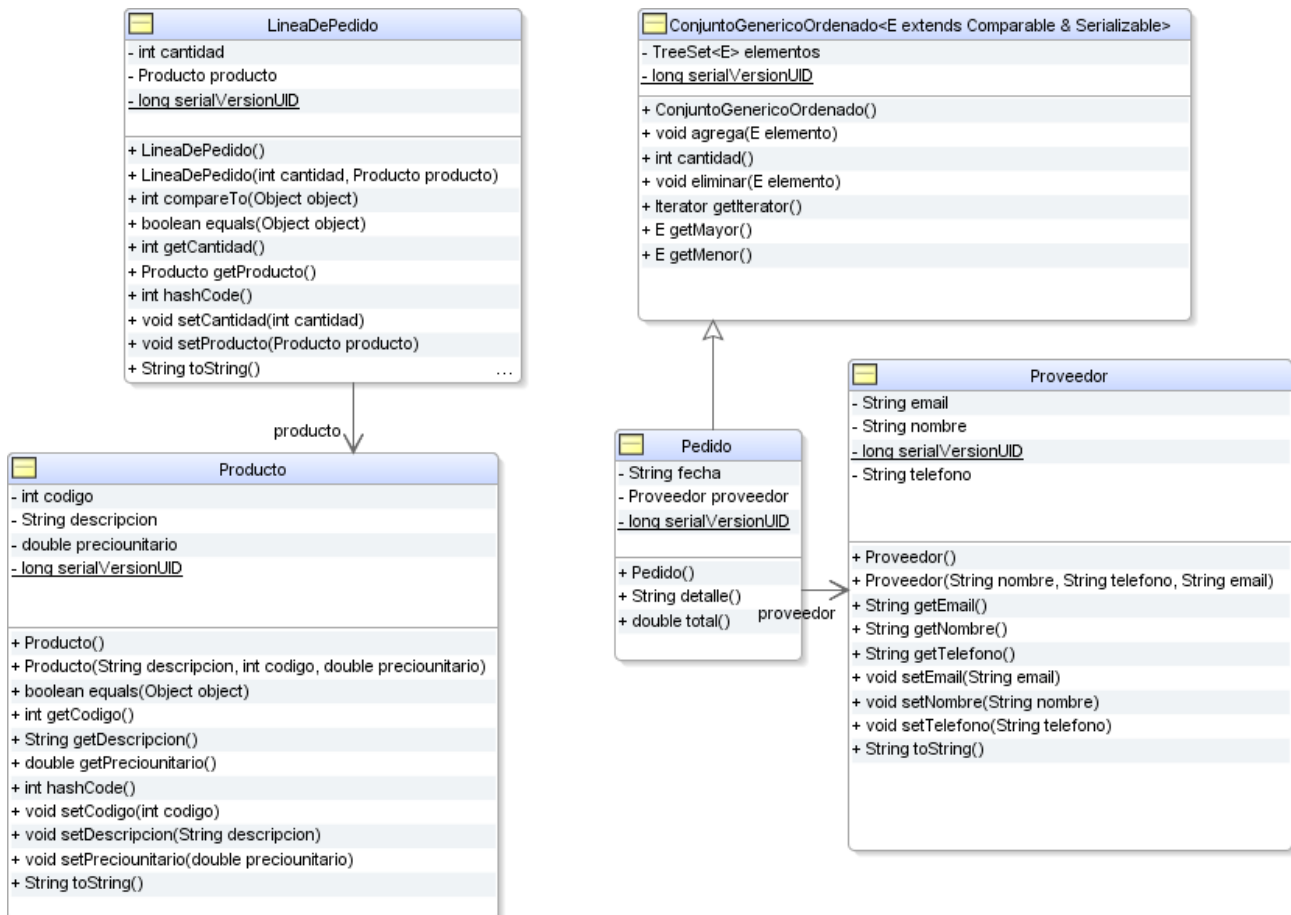
Inciso b)

Considera la forma de solucionar este problema si las 4 clases involucradas (**Vehiculo**, **Automovil**, **Motor** y **Flota**) no implementan la interfaz Serializable, no tienen setters (pero si getters) y no disponen de un constructor sin parámetros (solo tienen constructores un constructor cada una que solicita todos los atributos por parámetro).



Ejercicio 2

Tomaremos como base el ejercicio de la guía anterior que gestionaba un negocio, el cual manejaba las clases **Proveedor**, **Producto**, **Pedido**, y **LineaDePedido**, pero en este caso, implementarán la interfaz **Serializable**.



Un pedido involucra un proveedor (al cual se le requerirán varios productos), una fecha, y un conjunto de líneas de pedido cada línea de pedido consta de un producto específico, y de una cantidad.

La clase **LineaDePedido** deberá sobrescribir el método **equals**, considerando como criterio el producto pedido. Deberá también reescribir el método **toString**.

La clase **Pedido** se extiende de un **ConjuntoGenericoOrdenado** de **LineasDePedido** deberá permitirnos obtener un detalle del conjunto de líneas de pedido y obtener el precio total de un pedido.

La clase **Pedido** tiene un atributo booleano llamado "entregado" que indica si el proveedor ha realizado la entrega del pedido. Al momento de crearse el pedido, el valor de "entregado" será false.

Inciso a)

Deberás lograr persistencia de este negocio, de manera poder:

Listar todos los proveedores registrados.



Listar todos los productos registrados.

Listar los pedidos que fueron entregados y los que no.

Listar los proveedores que adeudan un pedido.

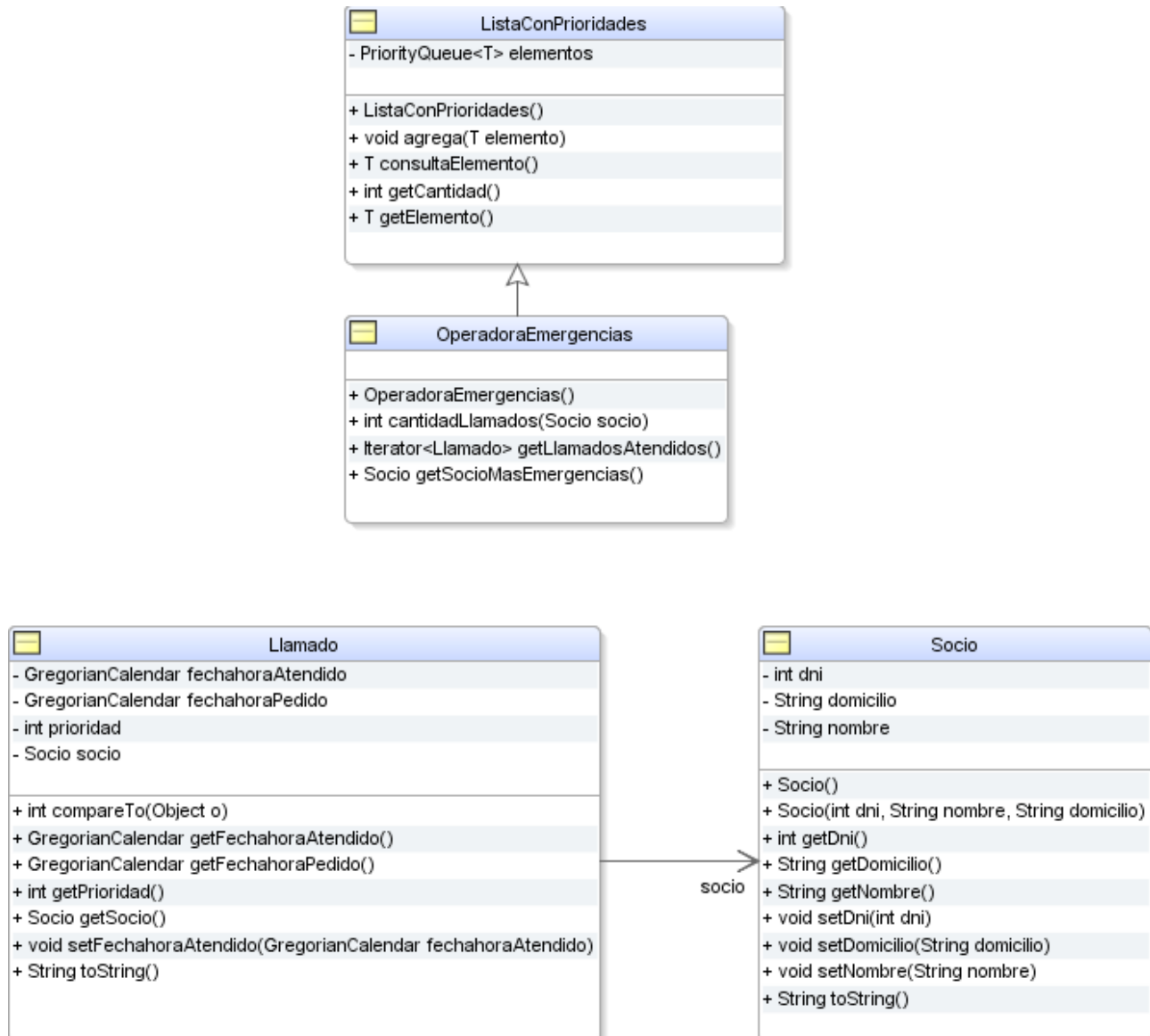
Inciso b)

Realiza una aplicación visual sencilla para implementar el sistema.



Ejercicio 3

Tomaremos el ejercicio de la empresa de emergencias médicas de la guía anterior, el diagrama de clases (sin realizar las modificaciones necesarias) es el siguiente:



El sistema debe implementar persistencia, y deberá ser capaz de:

Listar todos los socios

Listar los llamados que aún no han sido atendidos.

Mostrar el próximo llamado que será atendido.

Listar el histórico de los llamados atendidos



Ejercicio 4

Tomando el ejercicio de la guía 6 referente al juego de estrategia, se deberá lograr persistencia.

Analiza las diferentes formas de lograr la persistencia del mapa en su totalidad, y luego implementa la que resulte más conveniente.

Cuando la vitalidad de un Personaje sea menor o iguala cero, el personaje deberá ser eliminado del mapa. Se deberán conservar los sucesivos cambios que se sucedan en el mapa.