

Diagrama de clases

Para la implementación inicial de la aplicación se parte de una clase abstracta *Electrodomestico*, de la cual heredan *Television* y otra clase abstracta llamada *ElectrodomesticoVolumen*. De esta última heredan las clases *Lavadora* y *Frigorifico*. Por otro lado se tiene la clase *Venta*, que contiene al menos un objeto de clase *Electrodomestico* y que es extendida por otra clase *VentaDomicilio*, que gestionará aquellas ventas en las que haya que enviar el producto. A estas clases las acompañan la clase *Dimension* y la enumeración *ClaseEnergetica*.

Posteriormente, en los apartados 2 y 3, se añade al diagrama la clase *LecturaElectrodomesticos*, que contiene la función *leer*. Además, se modifica el método *equals* de *Electrodomestico* para gestionar correctamente las repeticiones.

En el apartado 4, se modifica la clase *Venta* añadiéndole variables de clase que llevan un registro de las ventas realizadas mediante una especie de pila (*pilaVentas* y *top*), y métodos de clase que manipula dichas variables (*ultima*, *anular* y *resumenVentas*).

Por último, en el apartado 5, se añaden las clases *VentaCanarias* y *TelevisionCurva*. La primera hereda de *Venta* y se encarga de gestionar el coste de los portes de los pedidos realizados desde las Canarias. Nótese que su constructor no admite la asignación de un electrodoméstico viejo a entregar y que redefine el método *setElectrodomesticoViejo* para que no haga nada. La segunda extiende la clase *Television*, añadiéndole dimensiones y peso. Desde un punto de vista más abstracto, *TelevisionCurva* debería heredar de *Television* y de *ElectrodomesticoVolumen*, pero al no soportar Java la herencia múltiple, se optó por la alternativa implementada.

P3: Diagrama de Clases

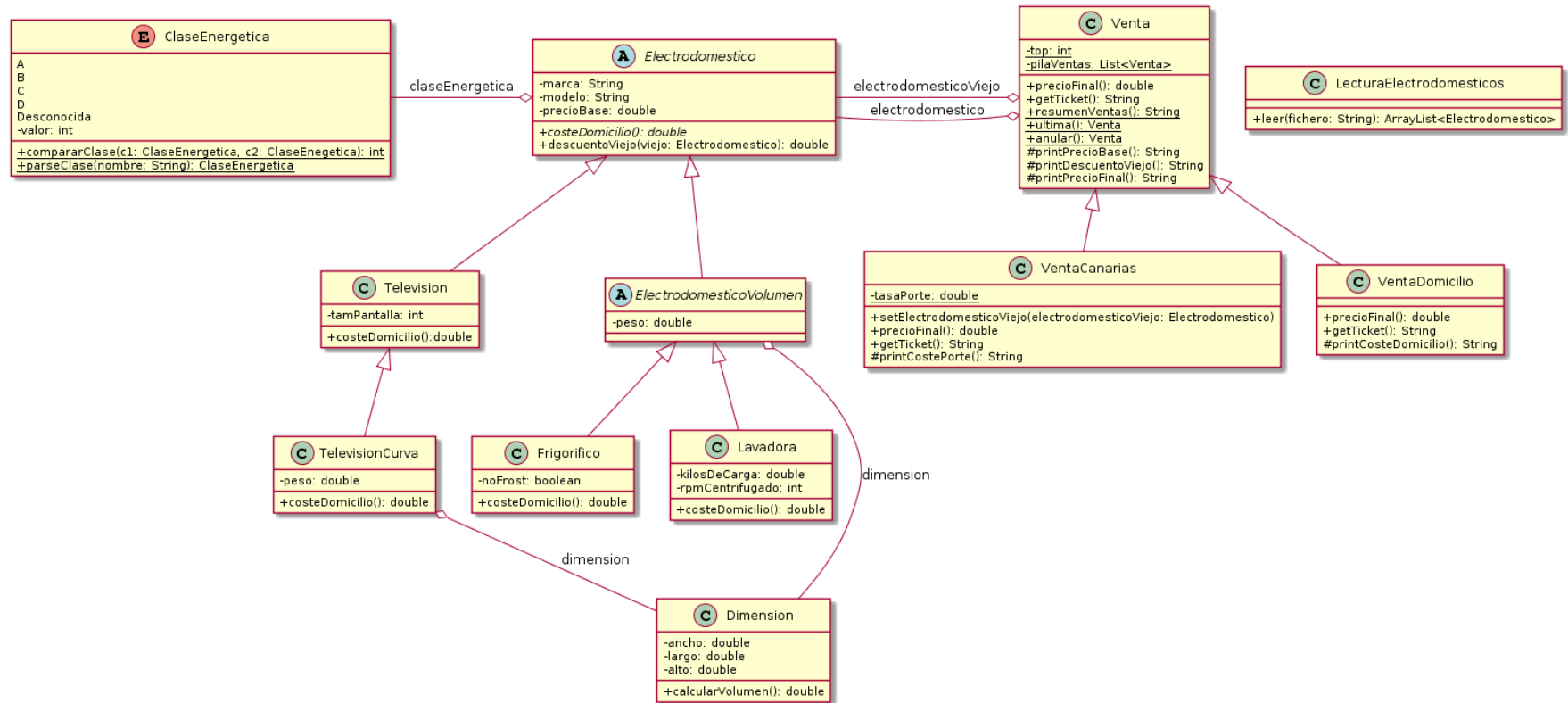


Figura 1: Diagrama de clases de la aplicación de la práctica 3