Junco de las Heras junco.heras@estudiante.uam.es
Marta Vaquerizo marta.vaquerizo@estudiante.uam.es

# PRÁCTICA 3
# ADSOF

## Ejercicio 1)

En este apartado se han creado las clases que se piden, y las que teníamos del diagrama de clases del ejercicio 2 de la práctica anterior. En especial, se han añadido el boolean frost, para el frigorífico, así como los double kilos, y double rpm para las lavadoras. Asimismo, se han añadido funciones para crear un Ticket, y para que fuese como la salida esperada, y para no repetir código, también se han creado funciones para cada una de las líneas del Ticket.

## Ejercicio 2)

En este segundo apartado, se ha creado la clase LecturaElectrodomesticos, una clase que lee un Electrodomestico de un fichero. Se verifica que funciona con el TesterTienda2, que la salida es la esperada.

### Ejercicio 3)

En este apartado, se crea el método equals en todas las clases que heredan de la clase Electrodomestico. Este método comprueba si dos electrodomésticos son iguales mirando que son de distinta clase, y si lo son, se mira si tienen el mismo modelo y marca. Se verifica que funciona con el método contains del ArrayList junto con el TesterTienda2, dando la salida esperada.

## Ejercicio 4)

En este apartado, se crean más funciones para calcular precios y totales como resumenVentas, para que puedan imprimirse fácilmente. También se ha incluido un ArrayList, static, en Venta para llevar un registro de las ventas, y se ha incluido el método anular(), para poder anular la última venta. Se comprueba con el TesterTienda3, dando la salida esperada.

### Ejercicio 5)

En este apartado, se añaden dos clases más, VentaCanarias, que hereda de VentaDomicilio, y TelevisionCurva, que hereda de Television. Este apartado está pensado para ver cuántos cambios realizábamos en las demás clases, y hemos podido comprobar que como hacíamos los métodos genéricos no hemos tenido realmente que cambiar código de las clases anteriores. Se ha creado TesterTienda5.java, un Tester con el que podemos ver si se imprimen bien los datos, y si los números son los correctos.

### Ejercicio 6)

Este último apartado, como es opcional se ha puesto en un paquete distinto, ya que no sabíamos cuántos cambios íbamos a realizar.

Al final se ha creado una clase Stock, con un atributo catalogo, que es ArrayList<Electrodomestico>, donde se guarda cada Electrodomestico que se crea. Cada vez que se ejecuta el constructor de Electrodomestico, se añade al catalogo de Stock. Para ello todos los métodos y el atributo son static. Cuando se hace una venta, se borra el Electrodomestico vendido del Stock, y se añade el que el electrodoméstico viejo, en caso de que lo haya en la venta. Cuando se anula la última venta, el Electrodomestico que se había vendido se vuelve a añadir, y el Electrodomestico viejo se devuelve, es decir, se quita de Stock. Se crea el TesterTienda6.java para ver cómo se imprime el Stock, primero todos a la vez, y luego según el tipo de Electrodomestico que es.

Junco de las Heras junco.heras@estudiante.uam.es
Marta Vaquerizo marta.vaquerizo@estudiante.uam.es

Se adjunta el diagrama de clases con todas las clases desarrolladas en los 6 ejercicios: