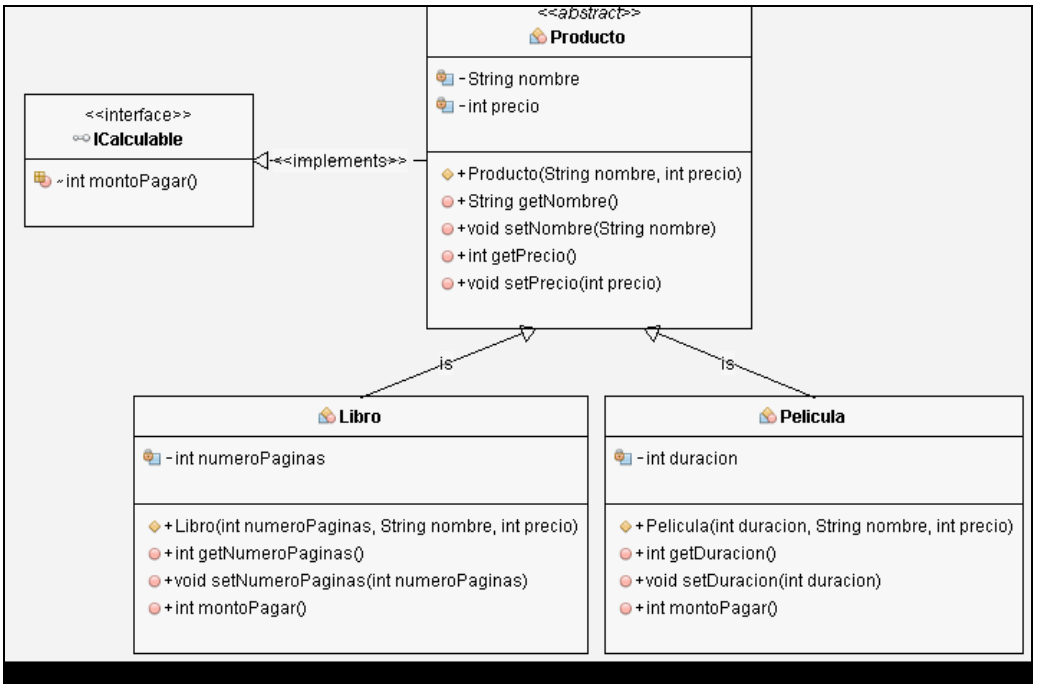
RECOPILACIÓN DE EJERCICIOS

DESCRIPCIÓN GENERAL

Para cada ejercicio se pide implementar las clases que aparecen en el diagrama entregado y la clase ***MainControl*** que le permita probar ***TODOS*** los métodos de la clase que maneja la colección.

EJERCICIO 1 – LIBROS Y PELICULAS



El monto a pagar se calcula dependiendo del tipo de producto:

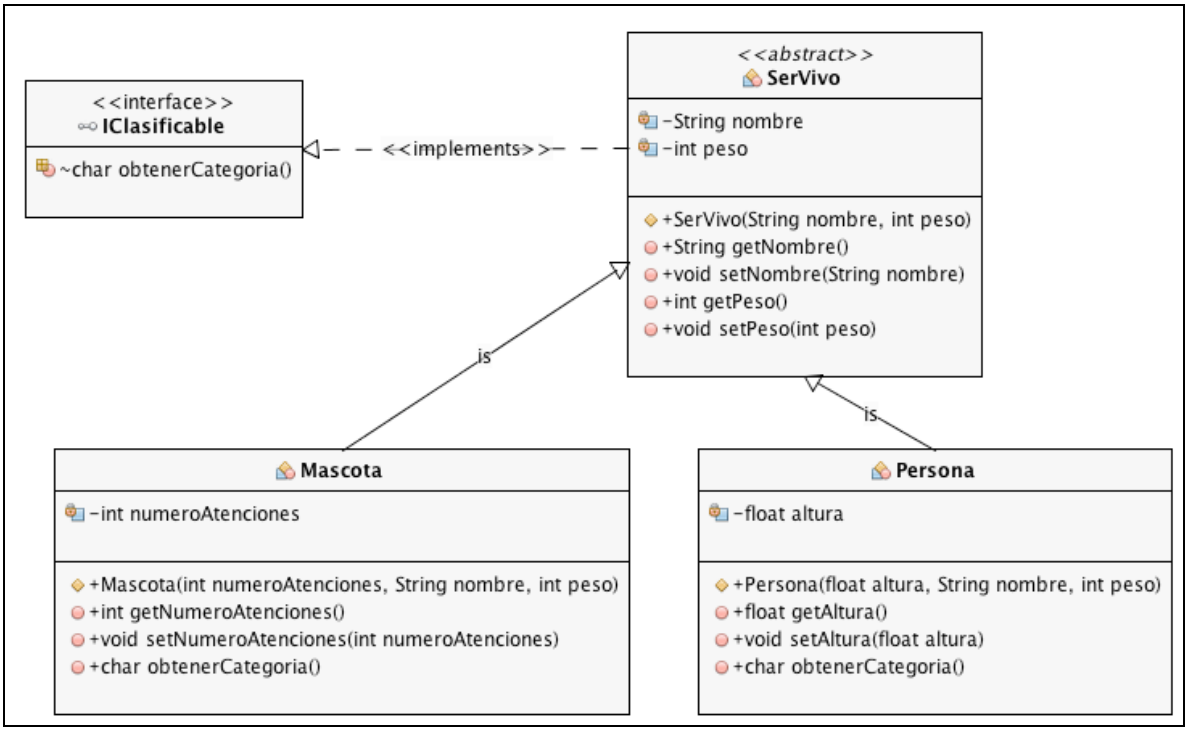
LIBRO	Monto_pagar = precio*1.19
PELICULA	Monto_pagar = precio*1.05

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarProducto(Producto p)	Agrega un producto a la colección considerando que los nombres de los productos SON ÚNICOS.
public int totalLibros()	Retorna el total de libros presentes en la colección.
public int totalProductos()	Retorna el total de productos presentes en la colección.
public int totalPeliculas()	Retorna el total de películas presentes en la colección.

MÉTODO	DESCRIPCIÓN
public void eliminarProductos(int tope)	Eliminar de la colección todos los productos cuyo monto a pagar sea mayor o igual a tope.
public void mostrarProductos()	Muestra los datos de los productos presentes en la colección. Deberá mostrar el tipo de producto (libro o película).

EJERCICIO 2 – SERES VIVOS



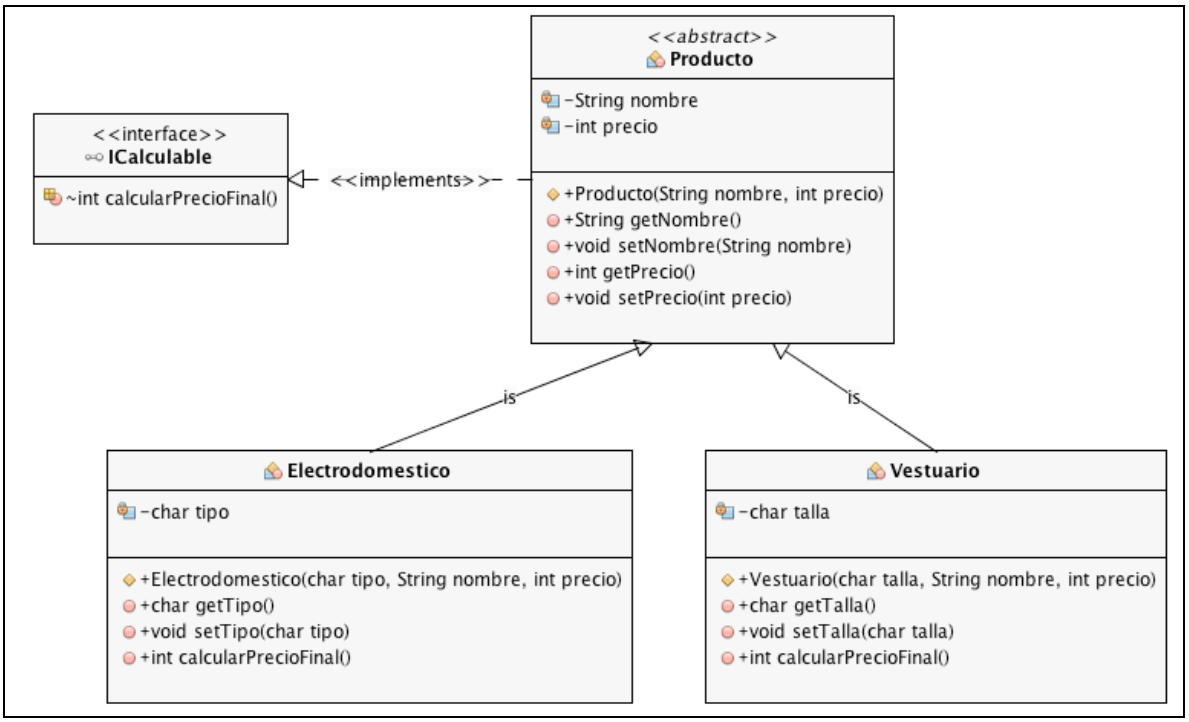
La categoria se obtiene dependiendo del ser vivo. Las personas son categoría 'A' si su altura es mayor a 1.50 y 'B' en caso contrario. Para el caso de las mascotas, son categoría 'A' si su peso es menor o igual a 3 y 'B' en caso contrario.

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarSerVivo(SerVivo p)	Agrega un ser vivo a la colección considerando que los nombres de los seres vivos SON ÚNICOS.
public int totalMascotas()	Retorna el total de mascotas presentes en la colección.
public int totalPersonas()	Retorna el total de personas presentes en la colección.
public int totalSeresVivos()	Retorna el total de seres vivos presentes en la colección.

MÉTODO	DESCRIPCIÓN
public void eliminarSeresVivos(char categoria)	Eliminar de la colección todos los seres vivos cuya categoría es igual al valor pasado como argumento.
public void mostrarSeresVivoss()	Muestra los datos de los seres vivos presentes en la colección. Deberá mostrar el tipo de ser vivo (mascota o persona).

EJERCICIO 3 –PRODUCTOS



El tipo de electrodoméstico debe ser 'V': vintage o 'N': normal. Las tallas del vestuario debe ser 'S','M' o 'L'. El precio final se calcula dependiendo del tipo de producto.

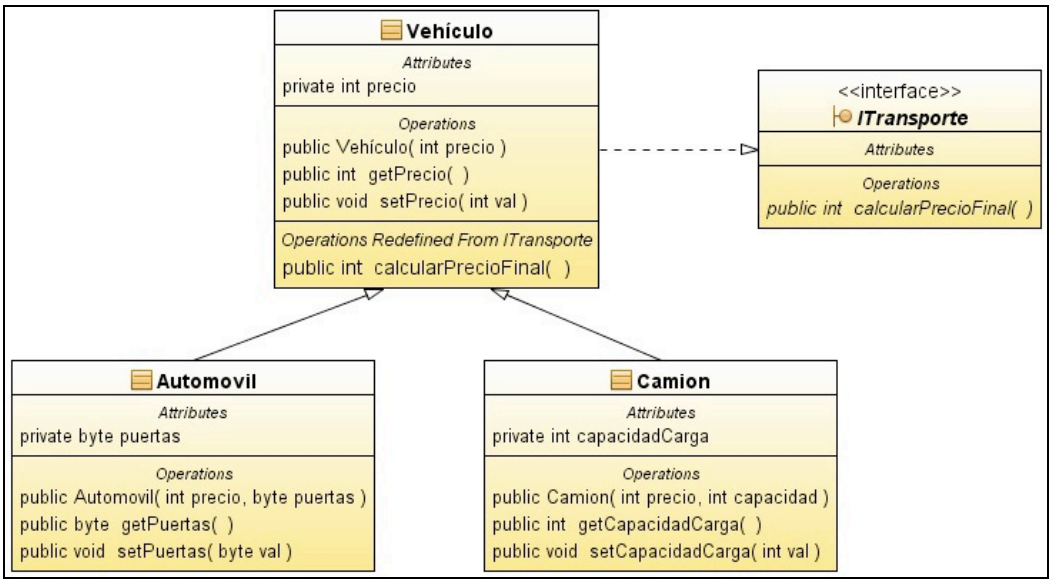
ELECTRODOMESTICO	Si el tipo es vintage su Precio_final = precio*0.9 De lo contrario su Precio_final = precio*0.95
VESTUARIO	Si la talla es 'S' o 'M' su Precio_final = precio*0.85 De lo contrario su Precio_final = precio*0.8

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarProducto(Producto p)	Agrega un producto a la colección considerando que los nombres de los productos SON ÚNICOS.
public int totalElectrodomestico()	Retorna el total de electrodomésticos presentes en la colección.

MÉTODO	DESCRIPCIÓN
public int totalProductos()	Retorna el total de productos presentes en la colección.
public int totalVestuario()	Retorna el total de vestuarios presentes en la colección.
public void eliminarProductos(int tope)	Eliminar de la colección todos los productos cuyo precio final sea mayor o igual a tope (argumento del método).
public void mostrarProductos()	Muestra los datos de los productos presentes en la colección. Deberá mostrar el tipo de producto (vestuario o electrodoméstico).

EJERCICIO 4 – VEHICULOS



El precio final se calcula dependiendo del tipo de vehículo:

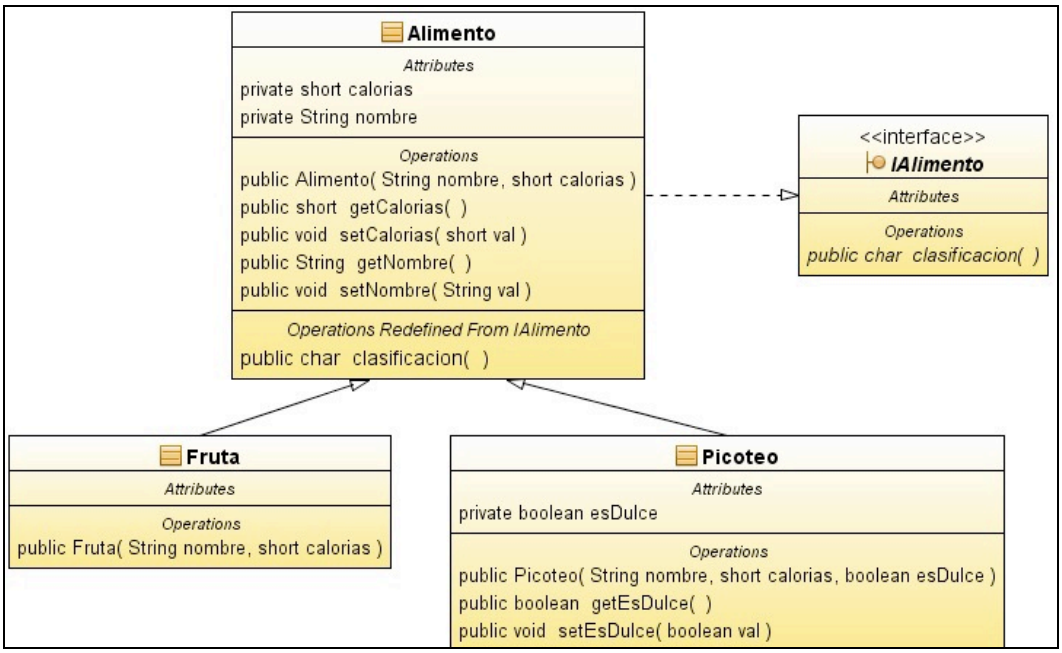
AUTOMÓVIL	Precio_final = precio*1.2
CAMION	Precio_final = precio*1.35

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarVehiculo(Vehiculo v)	Agrega un vehículo a la colección considerando que los precios de los vehículos SON ÚNICOS.
public int totalAutomoviles()	Retorna el total de automóviles presentes en la colección.

MÉTODO	DESCRIPCIÓN
public int totalVehiculos()	Retorna el total de vehículos presentes en la colección.
public int totalCamiones()	Retorna el total de camiones presentes en la colección.
public void eliminarVehículos(int tope)	Eliminar de la colección todos los vehículos cuyo precio final sea menor a tope.
public void mostrarVehiculos()	Muestra los datos de los vehículos presentes en la colección. Deberá mostrar el tipo de vehículo (automóvil o camión).

EJERCICIO 5 - ALIMENTOS

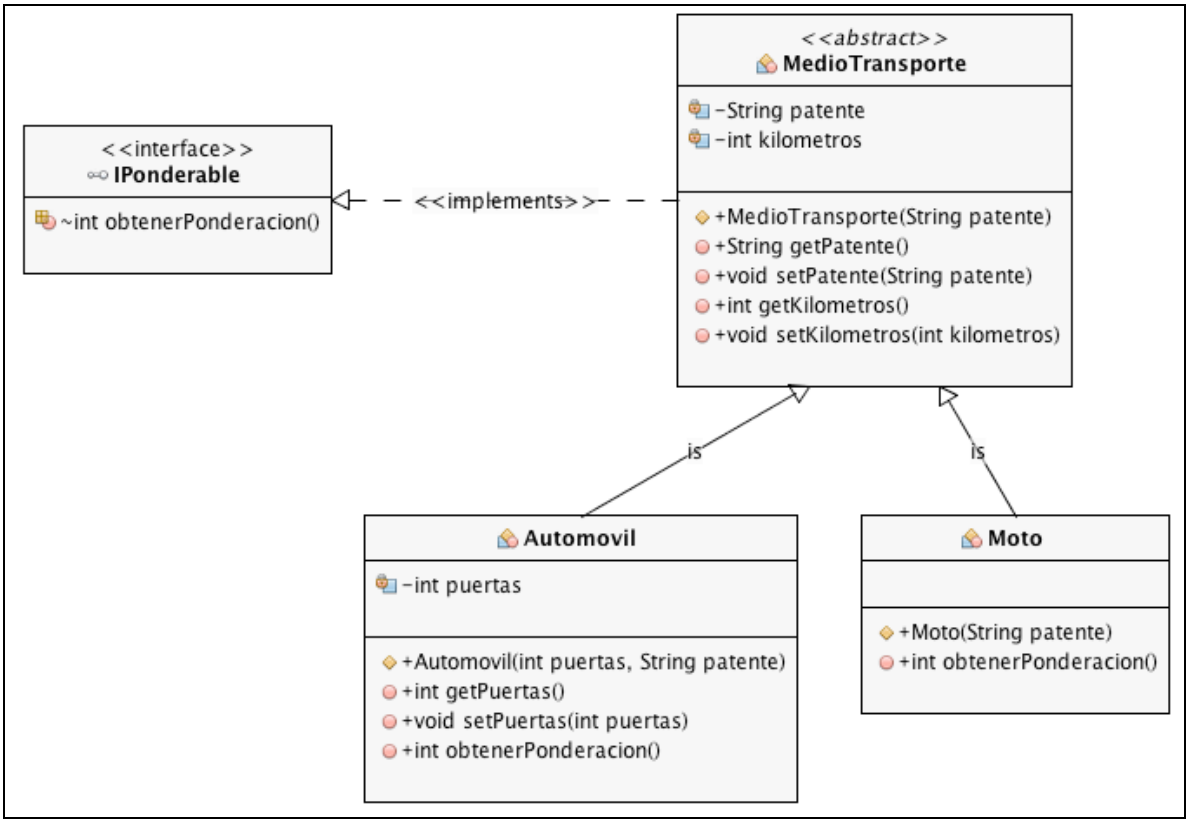


La clasificación de los alimentos depende del tipo, las frutas son saludables ('S') si sus calorias están dentro del rango: 20 y 50; en caso contrario son poco saludables ('P'). Para el caso del picoteo: son saludables ('S') si sus calorias están dentro del rango: 10 y 40 y no son dulces; en caso contrario son poco saludables ('P').

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarAlimento(Alimento a)	Agrega un alimento a la colección considerando que los nombres de los alimentos SON ÚNICOS.
public int totalFrutas()	Retorna el total de frutas presentes en la colección.
public int totalAlimentos()	Retorna el total de alimentos presentes en la colección.
MÉTODO	**DESCRIPCIÓN**
public int totalPicoteos()	Retorna el total de picoteos presentes en la colección.
public void eliminarAlimentos(char tipo)	Eliminar de la colección todos los alimentos cuya clasificación sea igual a tipo (argumento del método)
public void mostrarAlimentos()	Muestra los datos de los alimentos presentes en la colección. Deberá mostrar el tipo de alimento (fruta o picoteo).

EJERCICIO 6 – MEDIOS DE TRANSPORTE

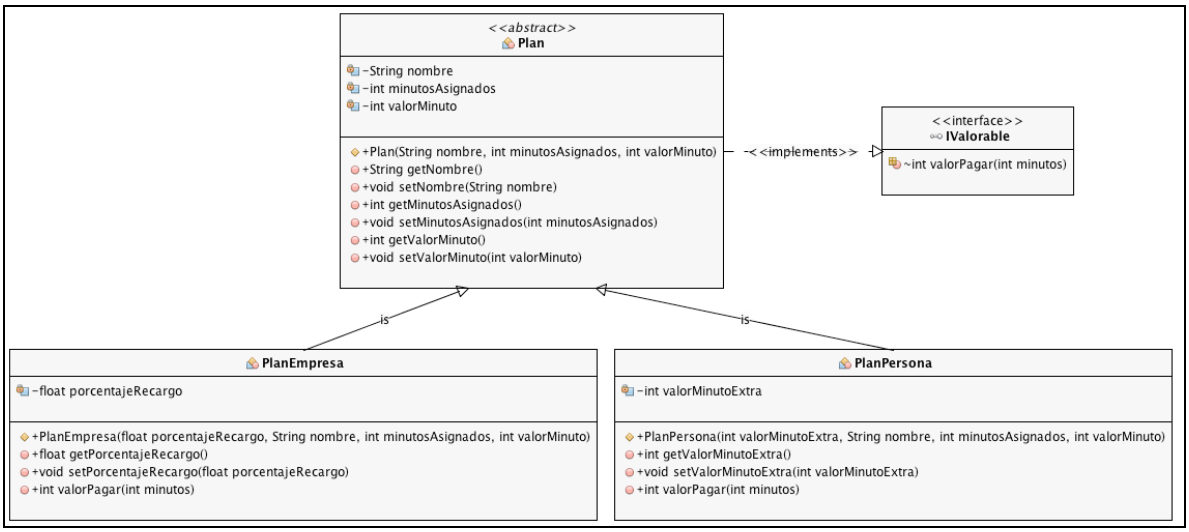


La ponderación depende del medio de transporte si es moto se pondera con un 5 y si es auto depende de su número de puertas: si tiene 3 puertas pondera en 6 y si tiene 5 puertas pondera en 7.

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarMedio(MedioTransporte m)	Agrega un medio de transporte a la colección considerando que las patentes de los medios de transporte SON ÚNICOS.
public int totalAutomoviles()	Retorna el total de automóviles presentes en la colección.
public int totalMedios()	Retorna el total de medios de transporte presentes en la colección.
public int totalMotos()	Retorna el total de motos presentes en la colección.
public void eliminarMedios(int ponderacion)	Eliminar de la colección todos los medios de transporte cuya ponderación sea igual a ponderación (argumento).
public void mostrarMedios()	Muestra los datos de los medios de transporte presentes en la colección. Deberá mostrar el tipo de medio de transporte (automóvil o moto).

EJERCICIO 7 – PLANES TELEFÓNICOS



El valor a pagar depende del tipo de plan.

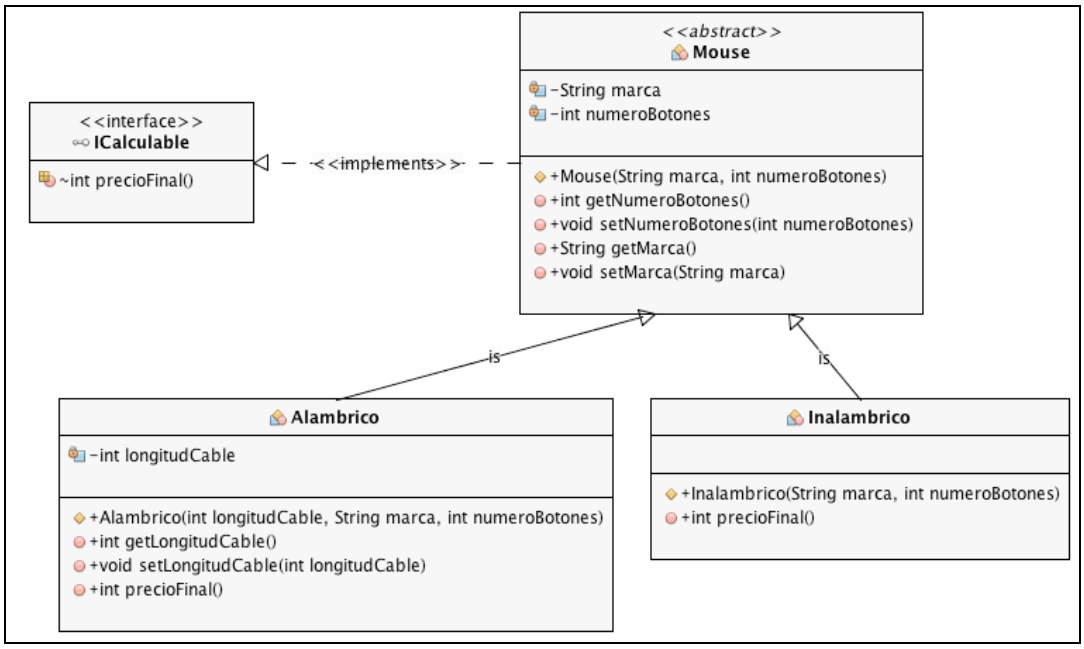
PLAN EMPRESA	En caso de que los minutos hablados (argumento del método) supere los minutos asignados Valor a pagar = (minutos asignados * valor minuto)*(1+porcentajeRecarga) En caso contrario Valor a pagar = (minutos asignados * valor minuto)

PLAN PERSONA	En caso de que los minutos hablados (argumento del método) supere los minutos asignados Valor a pagar = (minutos asignados * valor minuto) + (minutos extra*valor minuto extra) En caso contrario Valor a pagar = (minutos asignados * valor minuto)

Requerimientos de colección

MÉTODO	**DESCRIPCIÓN**
public boolean agregarPlan(Plan p)	Agrega un plan a la colección considerando que los nombres de los planes SON ÚNICOS.
public int totalPlanesEmpresa()	Retorna el total de planes empresa presentes en la colección.
public int totalPlanes()	Retorna el total de planes presentes en la colección.
public int totalPlanesPersona()	Retorna el total de planes persona presentes en la colección.
public void eliminarPlanes(int minimo)	Eliminar de la colección todos los planes cuyo valor a pagar sea mayor o igual al valor entregado como argumento.
public void mostrarPlanes()	Muestra los datos de los planes presentes en la colección. Deberá mostrar el tipo de plan (empresa o persona).

EJERCICIO 8 – MOUSE



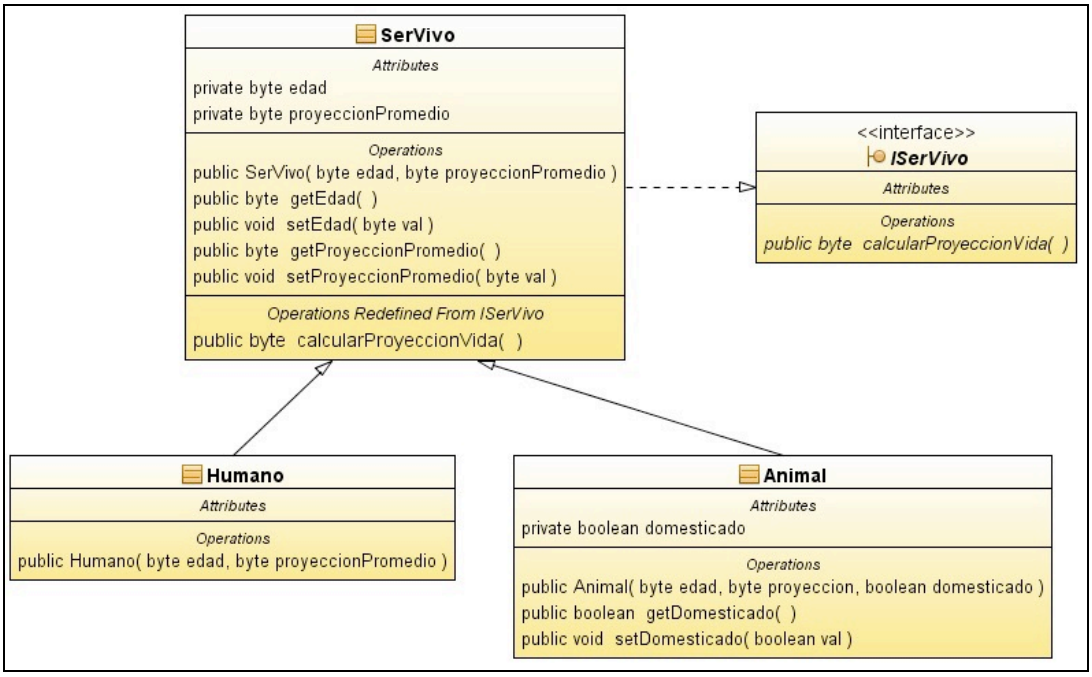
El precio final se calcula dependiendo del tipo de mouse:

ALAMBRICO	Si la longitud del cable es de 30 entonces el precio final es de 900 en caso contrario es de 800.
INALAMBRICO	Si tiene 2 botones es de 1000 en caso contrario es de 1500.

Requerimientos de colección

MÉTODO	**DESCRIPCIÓN**
public boolean agregarMouse(Mouse m)	Agrega un mouse a la colección considerando que los marcas de los mouse SON ÚNICAS.
public int totalMouseAlambricos()	Retorna el total de mouse alámbricos presentes en la colección.
public int totalMouse()	Retorna el total de mouse presentes en la colección.
public int totalMouseInalambricos()	Retorna el total de mouse inalámbricos presentes en la colección.
public void eliminarMouse(int inferior, int superior)	Eliminar de la colección todos los mouse cuyo precio final se encuentre dentro del rango inferior y superior (argumentos).
public void mostrarMouse()	Muestra los datos de los mouse presentes en la colección. Deberá mostrar el tipo de mouse (inalámbrico o alámbrico).

EJERCICIO 9 – SER VIVO



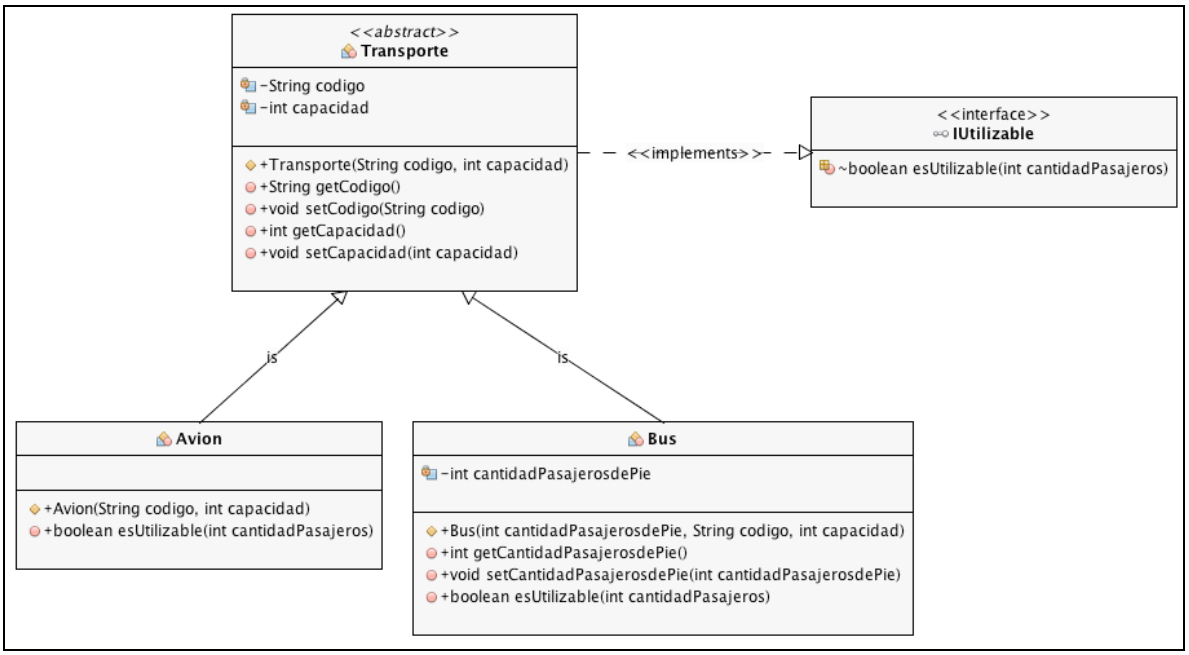
La proyección de vida se calcula dependiendo del tipo de ser vivo:

ANIMAL	Si está domesticado entonces proyección de vida = proyección promedio*1.1 De lo contrario proyección de vida = proyección promedio*1.05
HUMANO	Si la edad es menor a 60 entonces proyección de vida = proyección promedio*1.03 De lo contrario proyección de vida = proyección promedio*1.5

Requerimientos de colección

MÉTODO	**DESCRIPCIÓN**
public boolean agregarSerVivo(SerVivo s)	Agrega un ser vivo a la colección considerando que las edades de los seres vivos SON ÚNICAS.
public int totalHumanos()	Retorna el total de humanos presentes en la colección.
public int totalSeresVivos()	Retorna el total de seres vivos presentes en la colección.
public int totalAnimales()	Retorna el total de animales presentes en la colección.
public void eliminarSeresVivos(byte limite)	Eliminar de la colección todos los seres vivos cuya proyección sea mayor a límite.
public void mostrarSeresVivos()	Muestra los datos de los seres vivos presentes en la colección. Deberá mostrar el tipo de ser vivo (humano o animal).

EJERCICIO 10 – TRANSPORTES



Saber si un transporte es utilizable (método de la interfaz) depende del tipo de transporte. En el caso del avión se debe comparar la cantidad de pasajeros (argumento del método) con la capacidad del avión. Para el caso del bus se debe considerar además la cantidad de pasajeros de pie.

Requerimientos de colección

MÉTODO	DESCRIPCIÓN
public boolean agregarTransporte(Transporte t)	Agrega un transporte a la colección considerando que los códigos de los transportes SON ÚNICOS.
public int totalAviones()	Retorna el total de aviones presentes en la colección.
public int totalTransportes()	Retorna el total de transportes presentes en la colección.
public int totalBuses()	Retorna el total de buses presentes en la colección.
public void eliminarTransportes(int pasajeros)	Eliminar de la colección todos los transportes que no son utilizables para la cantidad de pasajeros indicada como argumento.
public void mostrarTransportes()	Muestra los datos de los transportes presentes en la colección. Deberá mostrar el tipo de transporte (avión o bus).