

Parte 4 - POO | Ignacio CAVALLO

Clase 23 | 31/05

Conceptos Aprendidos

En Clase:

- Clases Abstractas
- El uso de *abstract*
- Debo admitir que ha sido más complejo de lo esperado poder internalizar las ideas que implican las clases Abstractas y como diferenciarlas de las Clases Padres.
- Supongo que con refuerzo externo, lograré comprenderlas de una mejor manera.
- Lo importante aquí es lo siguiente:

Un clase abstracta es como un **molde de un molde**. Básicamente, un **Inception de Clases**
Ignacio Cavallo

Fuera de clase:

- Uso de ArrayList
- Listado de Clases creadas por mi.
- Ahorro de código

Primera Parte

Se propuso la creación de una clase Electrodoméstico, la cual tendrá como objetivo ser la madre, de otras subclases. Aquí mis desafíos fue poder interpretar como diferenciar las distintas clases hijas, de su clase padre. Admito que mi desarrollo es mucho más simple que en otras entregas, pero todavía estoy procesando lo enseñado en clase. Sin embargo, siendo simple, termina cumpliendo el objetivo de demostrar un grado de entendimiento sobre el tema. A continuación muestro como se oponen dos electrodomesticos diferentes.

```
public class Tv extends Electrodomestico{
    private int pulgadas;

    public Tv(String marca, String modelo, int ancho, int largo, int profundidad,
double consumo, int pulgadas) {
        super(marca, modelo, ancho, largo, profundidad, consumo);
        this.pulgadas = pulgadas;
    }

    public void cambioCanal(int canal){
        System.out.println("Se cambió al canal: " + canal);
    }

    /* Microondas */
    public class Microondas extends Electrodomestico {
```

```

    public Microondas(String marca, String modelo, int ancho, int largo, int
profundidad, double consumo) {
        super(marca, modelo, ancho, largo, profundidad, consumo);
    }

    public void calentar(){
        System.out.println("Calentando....");
    }
}

```

El ejercicio completo está [aquí](#)

ElectroDoméstico

```

/* Electrodoméstico */
package com.nacho;

public class Electrodomestico {
    private String marca;
    private String modelo;
    private int ancho;
    private int largo;
    private int profundidad;
    private double consumo;
    private double consumoKw;
    private double consumoXDía;

    private boolean estaEncendido;

    public Electrodomestico(){}

    public Electrodomestico(String marca, String modelo, int ancho, int largo, int
profundidad, double consumo) {
        this.marca = marca;
        this.modelo = modelo;
        this.ancho = ancho;
        this.largo = largo;
        this.profundidad = profundidad;
        this.consumo = consumo;
        this.consumoKw = consumo/1000;
        this.consumoXDía = 0;
        this.estaEncendido = true;
    }

    public Electrodomestico(String marca, String modelo ){
        this.marca = marca;
        this.modelo = modelo;
    }
}

```

```

    public void onOff(){
        if(!this.estaEncendido){
            System.out.println("Encendiendo...");
        }else {
            this.estaEncendido = !this.estaEncendido;
            System.out.println("Apagando");
        }
    }

    public double volumen(){
        return this.ancho * this.largo * this.profundidad;
    }

    public void consumoDiario(double tiempoUso){
        this.consumoXDía = consumoKw * tiempoUso;
        System.out.println("Consumo diario es de: " + this.consumoXDía + " Kw/h");
    }
    public void consumoMensual(int cantDias){
        double totalMes = consumoXDía * cantDias;

        System.out.println("Consumo mensual es de: " + totalMes + " Kw/h");
    }
}

/* TV */

public class Tv extends Electrodomestico{
    private int pulgadas;

    public Tv(String marca, String modelo, int ancho, int largo, int profundidad,
double consumo, int pulgadas) {
        super(marca, modelo, ancho, largo, profundidad, consumo);
        this.pulgadas = pulgadas;
    }

    public void cambioCanal(int canal){
        System.out.println("Se cambió al canal: " + canal);
    }
}

package com.nacho;

public class Main {

    public static void main(String[] args) {

        var tv= new Tv("Smansung", "UHD43",100, 75, 10,11,43 );
        tv.onOff();
    }
}

```

```

        tv.cambioCanal(42);
        tv.consumoDiario(3.5);
        tv.consumoMensual(25);

        var microondas = new Microondas("Samsung", "MS02", 40,20, 60, 120 );
        microondas.onOff();
        microondas.consumoDiario(.5);
        microondas.consumoMensual(28);
        microondas.calentar();
    }
}

/* Microondas */

public class Microondas extends Electrodomestico {

    public Microondas(String marca, String modelo, int ancho, int largo, int
profundidad, double consumo) {
        super(marca, modelo, ancho, largo, profundidad, consumo);
    }

    public void calentar(){
        System.out.println("Calentando....");
    }
}

package com.nacho;

public class Main {

    public static void main(String[] args) {

        var tv= new Tv("Smansung", "UHD43",100, 75, 10,11,43 );
        tv.onOff();
        tv.cambioCanal(42);
        tv.consumoDiario(3.5);
        tv.consumoMensual(25);

        var microondas = new Microondas("Samsung", "MS02", 40,20, 60, 120 );
        microondas.onOff();
        microondas.consumoDiario(.5);
        microondas.consumoMensual(28);
        microondas.calentar();
    }
}

```

Segunda Parte.

Se nos explico sobre las clases abstractas. Aquellas clases que no son instanciadas, solo se usan como un molde. Nos dejaron de tarea crear de forma abstracta una persona y a su vez, tres diferentes subclases. Voy a

dejar el ejercicio completo a continuación. Nuevamente pido disculpas si la implementación no es la mejor, pero insisto, todavía estoy procesando lo enseñado.

```
package com.nacho;

public abstract class Persona {
    private String nombre;
    private String apellido;
    private int edad;
    private String prefijo ;

    public Persona(){}

    public Persona(String nombre, String apellido, int edad){
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad<0 ? 0 : this.edad;
        this.prefijo = this.nombre + " dice :";
    }

    public abstract void actividadLaboral();

    public abstract void actividadRecreativa();

    public abstract void dormir();

    public void hablar(String Frase){
        System.out.println(this.nombre+ " dice: " +Frase);
    }

    public abstract void contarHastaN(int n);

    @Override
    public String toString() {
        return "Persona{" +
            "nombre='" + nombre + '\'' +
            ", apellido='" + apellido + '\'' +
            ", edad=" + edad +
            '}';
    }
}

/* Adulto */

package com.nacho;

public class Adulto extends Persona {
    private String trabajo;
```

```

    public Adulto(String trabajo, String nombre, String apellido, int edad) {
        super(nombre, apellido, edad);
        this.trabajo = trabajo;
    }

    @Override
    public void actividadLaboral() {
        System.out.println("Mi trabajo es " + trabajo);
    }

    @Override
    public void actividadRecreativa() {
        System.out.println("No tengo tiempo para hacer deporte");
    }

    @Override
    public void dormir() {
        System.out.println("Duermo unas 6 horas diarias");
    }

    @Override
    public void contarHastaN(int n){

        for(var i=1; i<n; i++){
            System.out.println(i);
        }
    }
}

/* Infante */
package com.nacho;

public class Infante extends Persona {
    public Infante(String nombre, String apellido, int edad) {
        super(nombre, apellido, edad);
    }

    @Override
    public void actividadLaboral() {
        System.out.println("No tengo trabajo, soy un pibe!");
    }

    @Override
    public void actividadRecreativa() {
        System.out.println("Juego, corro, veo dibujitos, hago lo que quiero.");
    }

    @Override
    public void dormir() {
        System.out.println("Duermo más o menos unas 12 horas");
    }

    @Override

```

```

        public void contarHastaN(int n) {
            if (n > 10){
                System.out.println("Se contar hasta diez solamente.");
            }else{
                for(var i=1; i<10; i++){
                    System.out.println(i);
                }
            }
        }
    }

    /* Anciano */
    package com.nacho;

    public class Anciano extends Persona{
        public Anciano(String nombre, String apellido, int edad) {
            super(nombre, apellido, edad);
        }

        @Override
        public void actividadLaboral() {
            System.out.println("No tengo trabajo, ya me jubilé");
        }

        @Override
        public void actividadRecreativa() {
            System.out.println("Hablo con los vecinos, me voy al parque a dar una
            vuelta y juego a las cartas");
        }

        @Override
        public void dormir() {
            System.out.println("Duermo como tres horas.");
            System.out.println("Tipo 2 duermo una siesta.");
        }

        @Override
        public void contarHastaN(int n) {
            for(var i=1; i<n; i++){
                if(i == 11){
                    System.out.println("En que número me quedé??. Me olvidé");
                    break;
                }
                System.out.println(i);
            }
        }
    }
}

```