

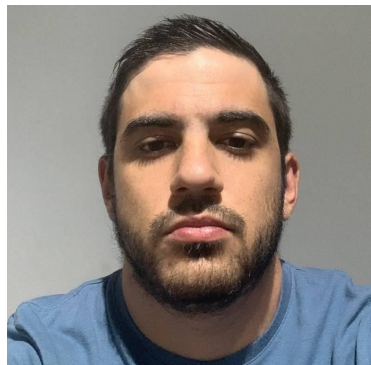
Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 2

Ignacio Lema Rodríguez - 267989



Gonzalo Rodríguez - 269042



Grupo M2B

Docente: Joaquín Rodríguez

ÍNDICE:

Diagrama UML Astah	3
Tabla canales HD y SD	4
CANALES HD	4
CANALES SD	4
Paquetes HD y SD	5
PAQUETES HD	5
PAQUETES SD	6
Códigos de Programación	7
Código dominio Administrador	7
Código dominio Paquete	17
Código dominio HD	20
Código dominio SD	23
Código dominio Canal	26

Diagrama UML Astah

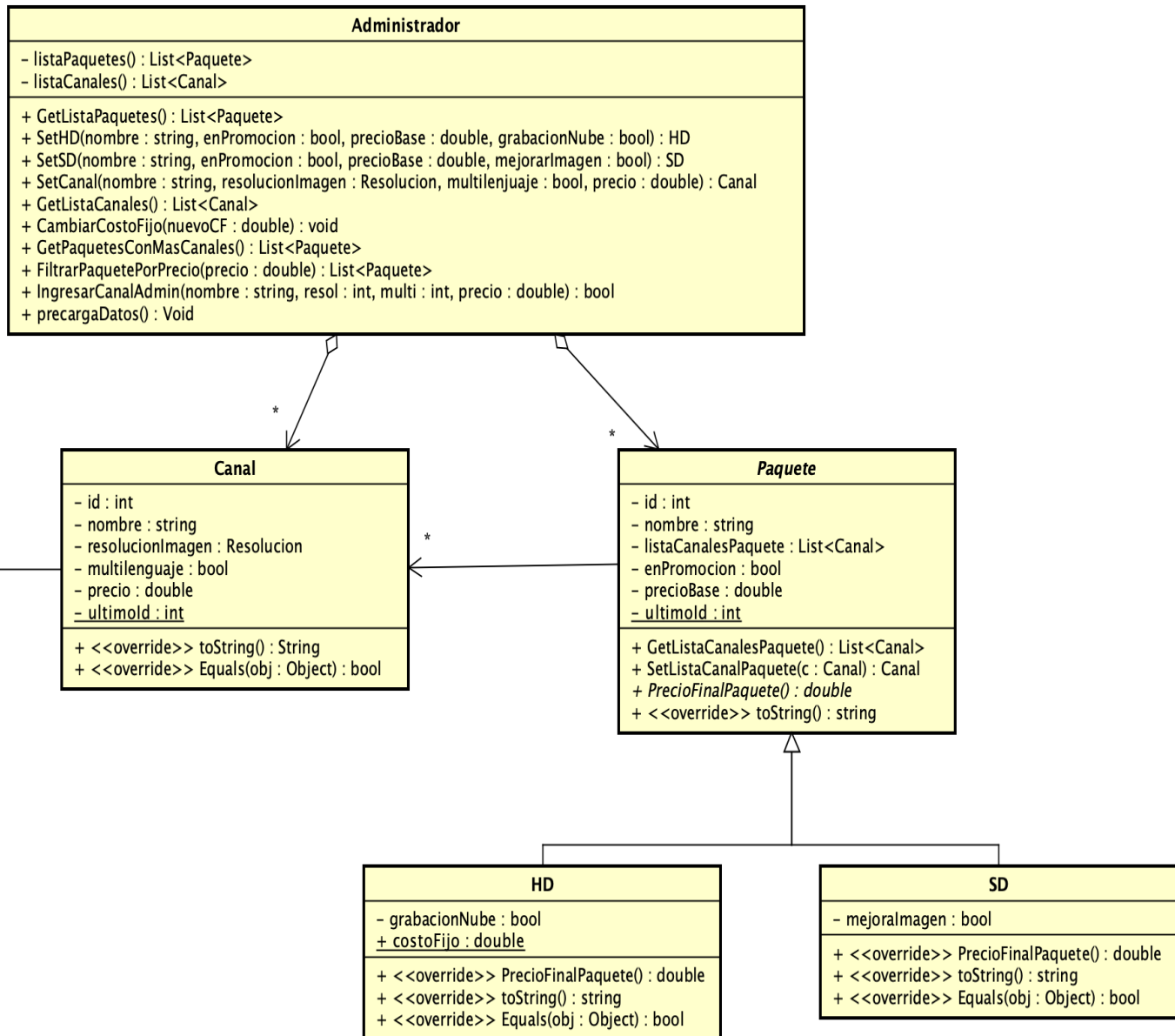


Tabla canales HD y SD

CANALES HD

Nombre Instancia	Nombre	Resolución	Multilinguaje	Precio
canalHD1	Discovery Kids	HD	True	55
canalHD2	FOX HD	HD	False	45
canalHD3	Warner HD	HD	True	35
canalHD4	History HD	HD	True	59
canalHD5	HBO HD	HD	False	34

CANALES SD

Nombre Instancia	Nombre	Resolución	Multilinguaje	Precio
canalSD1	Discovery Home	SD	False	54
canalSD2	ESPN	SD	True	99
canalSD3	HOLA TV	SD	False	22
canalSD4	VTV	SD	False	43.90
canalSD5	Canal 5	SD	False	24
canalSD6	Canal 4	SD	True	74

Paquetes HD y SD

PAQUETES HD

Nombre Instancia	Nombre	Promoción	Precio Base	Grabación Nube	Canales
paqueteHD1	Entretenimiento	False	567	True	canalHD1, canalHD2, canalHD3, canalHD4, canalHD5
paqueteHD2	Deporte	True	990	False	canalHD1, canalHD2, canalHD3, canalHD5
paqueteHD3	Dibujitos	False	340	True	canalHD1, canalHD2, canalHD3, canalHD4
paqueteHD4	Historia	True	758	False	canalHD1, canalHD2, canalHD3, canalHD4

PAQUETES SD

Nombre Instancia	Nombre	Promoción	Precio Base	Mejora de imagen	Canales
paqueteSD1	Películas	True	100	True	canalSD1, canalSD2, canalSD3, canalSD4
paqueteSD2	Series	False	124	False	canalSD1, canalSD2, canalSD3, canalSD4
paqueteSD3	Novelas Turcas	True	543	True	canalSD1, canalSD2, canalSD3, canalSD4, canalSD5, canalSD6
paqueteSD4	Boxeo	False	590	True	canalSD1, canalSD2, canalSD3, canalSD4, canalSD5, canalSD6

Códigos de Programación

Código dominio Administrador

```
using System;

using System.Collections.Generic;


namespace ObligatorioP2

{

    public class Administrador

    {

        private List<Paquete> listaPaquetes = new List<Paquete>();

        private List<Canal> listaCanales = new List<Canal>();


        public Administrador()

        {

            Console.Clear();

            precargaDatos();

        }


        public List<Paquete> GetListaPaquetes()

        {

            return listaPaquetes;

        }

    }

}
```

```

public HD SetHD(string nom, bool enProm, double preBas, bool grabNub)

{

    //VALIDACION PARA EVITAR CREAR PAQUETE CON MISMO NOMBRE QUE OTRO

    HD paqueteResultante = null;

    HD aux = new HD();

    aux.Nombre = nom;

    int existeHD = GetListaPaquetes().IndexOf(aux);

    if (existeHD.Equals(-1))

    {

        HD nuevoPaquete = new HD(nom, enProm, preBas, grabNub);

        listaPaquetes.Add(nuevoPaquete);

        paqueteResultante = nuevoPaquete;

    }

    return paqueteResultante;

}

```

```

public SD SetSD(string nom, bool enProm, double preBas, bool mejIma)

{

    //VALIDACION PARA EVITAR CREAR PAQUETE CON MISMO NOMBRE QUE OTRO

    SD paqueteResultante = null;

    SD aux = new SD();

    aux.Nombre = nom;

    int existeSD = GetListaPaquetes().IndexOf(aux);

```



```

        if (existeSD.Equals(-1))

        {

            SD nuevoPaquete = new SD(nom, enProm, preBas, mejIma);

            listaPaquetes.Add(nuevoPaquete);

            paqueteResultante = nuevoPaquete;

        }

        return paqueteResultante;

    }

    public Canal SetCanal(string nom, Canal.Resolucion resol, bool multi, double pre)

    {

        Canal nuevoCanal = null;

        if (pre > 0 && !nom.Equals("") && (resol == Canal.Resolucion.HD || resol == Canal.Resolucion.SD) &&
!multi.Equals(null))

        {

            //VALIDACION QUE EL NOMBRE DEL CANAL NO EXISTA

            Canal aux = new Canal();

            aux.Nombre = nom;

            int existeCanal = GetListaCanales().IndexOf(aux);

            if (existeCanal.Equals(-1))

            {

                nuevoCanal = new Canal(nom, resol, multi, pre);

```

```

        listaCanales.Add(nuevoCanal);

    }

}

return nuevoCanal;

}

public List<Canal> GetListaCanales()

{

    return listaCanales;

}

public bool IngresarCanalAdmin(string nombre, int resol, int multi, double pre)

{

    bool ingresado = false;

    Canal.Resolucion definicion = 0;

    if (resol.Equals(1))

    {

        definicion = Canal.Resolucion.HD;

    }

    else if (resol.Equals(2))

    {

```

```

        definicion = Canal.Resolucion.SD;

    }

    bool esMulti = false;

    if (multi.Equals(1))

    {

        esMulti = true;

    }

    Canal nuevoCanal = SetCanal(nombre, definicion, esMulti, pre);

    if (nuevoCanal != null)

    {

        ingresado = true;

    }

    return ingresado;

}

public void CambiarCostoFijo(double nuevoCF)

{

    if (nuevoCF > 0)

    {

        HD.costoFijo = nuevoCF;
    }
}

```

```

    }

}

public List<Paquete> GetPaquetesConMasCanales()

{

    List<Paquete> listaRet = new List<Paquete>();

    double minCant = double.MinValue;

    foreach (Paquete p in GetListaPaquetes())

    {

        if (p.GetListaCanalesPaquete().Count > minCant)

        {

            minCant = p.GetListaCanalesPaquete().Count;

            listaRet.Clear();

            listaRet.Add(p);

        }

        else if (p.GetListaCanalesPaquete().Count == minCant)

        {

            listaRet.Add(p);

        }

    }

    return listaRet;

}

public List<Paquete> FiltrarPaquetePorPrecio(double precio)

```

```

{

    List<Paquete> paquetesFiltrados = new List<Paquete>();

    foreach (Paquete p in GetListaPaquetes())

    {

        if (p.PrecioBase > precio)

        {

            paquetesFiltrados.Add(p);

        }

    }

    return paquetesFiltrados;

}

public void precargaDatos()

{

    //CANALES HD

    Canal canalHD1 = SetCanal("Discovery Kids HD", Canal.Resolucion.HD, true, 55);

    Canal canalHD2 = SetCanal("FOX HD", Canal.Resolucion.HD, false, 45);

    Canal canalHD3 = SetCanal("Warner HD", Canal.Resolucion.HD, true, 35);

    Canal canalHD4 = SetCanal("History HD", Canal.Resolucion.HD, true, 59);

    Canal canalHD5 = SetCanal("HBO HD", Canal.Resolucion.HD, false, 34);

    Canal canalHD6 = SetCanal("HBO HD", Canal.Resolucion.SD, true, 44); //ESTE NO DEBERIA CREARSE

    //CANALES SD

    Canal canalSD1 = SetCanal("Discovery Home", Canal.Resolucion.SD, false, 54);

```

```

Canal canalSD2 = SetCanal("ESPN", Canal.Resolucion.SD, true, 99);

Canal canalSD3 = SetCanal("HOLA TV", Canal.Resolucion.SD, false, 22);

Canal canalSD4 = SetCanal("VTV", Canal.Resolucion.SD, false, 43.90);

Canal canalSD5 = SetCanal("Canal 5", Canal.Resolucion.SD, false, 24);

Canal canalSD6 = SetCanal("Canal 4", Canal.Resolucion.SD, true, 74);

Canal canalSD7 = SetCanal("Canal 5", Canal.Resolucion.HD, true, 26); // NO DEBERIA CREARSE


//PAQUETES HD

HD paqueteHD1 = SetHD("Entretenimiento", false, 567, true);

HD paqueteHD5 = SetHD("Entretenimiento", false, 567, true); // NO SE CREA PORQUE YA EXISTE

paqueteHD1.SetListaCanalPaquete(canalHD1);

paqueteHD1.SetListaCanalPaquete(canalHD2);

paqueteHD1.SetListaCanalPaquete(canalHD3);

paqueteHD1.SetListaCanalPaquete(canalHD4);

paqueteHD1.SetListaCanalPaquete(canalHD4); // ESTE CANAL NO SE AGREGA PORQUE YA EXISTE

paqueteHD1.SetListaCanalPaquete(canalHD5);

paqueteHD1.SetListaCanalPaquete(canalHD5); // ESTE CANAL NO SE AGREGA PORQUE YA EXISTE


HD paqueteHD2 = SetHD("Deporte", true, 990, false);

paqueteHD2.SetListaCanalPaquete(canalHD1);

paqueteHD2.SetListaCanalPaquete(canalHD2);

paqueteHD2.SetListaCanalPaquete(canalHD3);

paqueteHD2.SetListaCanalPaquete(canalHD5);

```

```
HD paqueteHD3 = SetHD("Dibujitos", false, 340, true);
```

```
paqueteHD3.SetListaCanalPaquete(canalHD1);
```

```
paqueteHD3.SetListaCanalPaquete(canalHD2);
```

```
paqueteHD3.SetListaCanalPaquete(canalHD3);
```

```
paqueteHD3.SetListaCanalPaquete(canalHD4);
```

```
HD paqueteHD4 = SetHD("Historia", true, 758, false);
```

```
paqueteHD4.SetListaCanalPaquete(canalHD1);
```

```
paqueteHD4.SetListaCanalPaquete(canalHD2);
```

```
paqueteHD4.SetListaCanalPaquete(canalHD3);
```

```
paqueteHD4.SetListaCanalPaquete(canalHD4);
```

```
//PAQUETES SD
```

```
SD paqueteSD1 = SetSD("Películas", true, 100, true);
```

```
paqueteSD1.SetListaCanalPaquete(canalSD1);
```

```
paqueteSD1.SetListaCanalPaquete(canalSD2);
```

```
paqueteSD1.SetListaCanalPaquete(canalSD3);
```

```
paqueteSD1.SetListaCanalPaquete(canalSD4);
```

```
SD paqueteSD2 = SetSD("Series", false, 124, false);
```

```
paqueteSD2.SetListaCanalPaquete(canalSD1);
```

```
paqueteSD2.SetListaCanalPaquete(canalSD2);
```

```
paqueteSD2.SetListaCanalPaquete(canalSD3);
```

```
paqueteSD2.SetListaCanalPaquete(canalSD4);
```

```

SD paqueteSD3 = SetSD("Novelas Turcas", true, 543, true);

paqueteSD3.SetListaCanalPaquete(canalSD1);

paqueteSD3.SetListaCanalPaquete(canalSD2);

paqueteSD3.SetListaCanalPaquete(canalSD3);

paqueteSD3.SetListaCanalPaquete(canalSD4);

paqueteSD3.SetListaCanalPaquete(canalSD5);

paqueteSD3.SetListaCanalPaquete(canalSD6);


SD paqueteSD4 = SetSD("Boxeo", false, 590, true);

SD paqueteSD5 = SetSD("Boxeo", false, 590, true); // NO DEBERIA CREARSE PORQUE YA EXISTE

paqueteSD4.SetListaCanalPaquete(canalSD1);

paqueteSD4.SetListaCanalPaquete(canalSD2);

paqueteSD4.SetListaCanalPaquete(canalSD3);

paqueteSD4.SetListaCanalPaquete(canalSD4);

paqueteSD4.SetListaCanalPaquete(canalSD5);

paqueteSD4.SetListaCanalPaquete(canalSD6);

}

}

}

```


Código dominio Paquete

```
using System;

using System.Collections.Generic;


namespace ObligatorioP2

{

    public abstract class Paquete

    {

        protected static int ultimoId = 1;


        public int Id { get; set; }

        public string Nombre { get; set; }

        private List<Canal> listaCanalesPaquete = new List<Canal>();

        public bool EnPromocion { get; set; }

        public double PrecioBase { get; set; }


        public Paquete(string nombre, bool enPromocion, double precioBase)

        {

            Id = ultimoId;

            ultimoId++;

            Nombre = nombre;

            EnPromocion = enPromocion;

            PrecioBase = precioBase;
```

```

    }

    public Paquete() // para utilizar equals

    {

    }

    public List<Canal> GetListaCanalesPaquete()

    {

        return listaCanalesPaquete;

    }

    public Canal SetListaCanalPaquete(Canal c)

    {

        Canal ret = null;

        Canal aux = new Canal();

        aux.Nombre = c.Nombre;

        bool existeCanal = listaCanalesPaquete.Contains(aux);

        if (!existeCanal)

        {

            listaCanalesPaquete.Add(c);

            ret = c;

        }

```

```
        return ret;
    }

    public abstract double PrecioFinalPaquete();

    public override string ToString()
    {
        return " Nombre " + Nombre + " En promocion " + EnPromocion + " Precio base " + PrecioBase;
    }
}
}
```

Código dominio HD

```
using System;

using System.Collections.Generic;


namespace ObligatorioP2

{

    public class HD : Paquete

    {

        public static double costoFijo = 456.90;

        public bool GrabacionNube { get; set; }


        public HD(string nombre, bool enPromocion, double precioBase, bool grabacionNube) : base(nombre,
enPromocion, precioBase)

        {

            GrabacionNube = grabacionNube;

        }

        public HD() // Constructor para utilizar equals

        {

        }

        public override double PrecioFinalPaquete()

        {
```

```

double costoHD = PrecioBase;

foreach (Canal c in GetListaCanalesPaquete())

{

    costoHD += c.Precio;

}

if (GrabacionNube)

{

    costoHD += costoFijo;

}

if (EnPromocion)

{

    costoHD *= 0.50;

}

return costoHD;

}

public override string ToString()

{

    return base.ToString() + " Grabacion en nube " + GrabacionNube + " Costo Final " + PrecioFinalPaquete();

}

```

```
public override bool Equals(object obj)

{

    return obj is HD hD &&

        Nombre == hD.Nombre;

}

}
```

Código dominio SD

```
using System;

using System.Collections.Generic;


namespace ObligatorioP2

{

    public class SD : Paquete

    {

        public bool MejoraImagen { get; set; }


        public SD(string nombre, bool enPromocion, double precioBase, bool mejoraImagen) : base(nombre,
enPromocion, precioBase)

        {

            MejoraImagen = mejoraImagen;

        }


        public SD()

        {

        }


        public override double PrecioFinalPaquete()

        {

            double costoSD = PrecioBase;


            foreach (Canal c in GetListaCanalesPaquete())
```

```

{

    costoSD += c.Precio;

}

```

```

if (MejoraImagen)

```

```

{

    costoSD *= 1.20;

}

```

```

if (EnPromocion)

```

```

{

    costoSD *= 0.85;

}

```

```

return costoSD;

```

```

}

```

```

public override string ToString()

```

```

{

    return base.ToString() + " Mejora imagen " + MejoraImagen + " Costo Final " + PrecioFinalPaquete();

}

```

```

public override bool Equals(object obj)

```

```

{

    return obj is SD sD &&

```



```
        Nombre == sD.Nombre;  
    }  
}  
}
```

Código dominio Canal

```
using System;

using System.Collections.Generic;


namespace ObligatorioP2

{

    public class Canal

    {

        private static int ultimoID = 1;


        public int Id { get; set; }

        public string Nombre { get; set; }

        public Resolucion ResolucionImagen { get; set; }

        public bool Multilenguaje { get; set; }

        public double Precio { get; set; }


        public enum Resolucion

        {

            SD = 576,

            HD = 1080

        }

    }

}
```

```

public Canal()

{

}

public Canal(string nombre, Resolucion resolucionImagen, bool multilenguaje, double precio)

{

    Id = ultimoID;

    ultimoID++;

    Nombre = nombre;

    ResolucionImagen = resolucionImagen;

    Multilenguaje = multilenguaje;

    Precio = precio;

}

public override string ToString()

{

    return " ID " + Id + " Nombre " + Nombre + " Resolucion " + ResolucionImagen + " Multilenguaje " +
Multilenguaje + " Precio " + Precio;

}

public override bool Equals(object obj)

{

    return obj is Canal canal &&

        Nombre == canal.Nombre;

```

}

}

}