src\main\Canal.java

```java
public class Mensaje {
    private Canal emisor;
    private String mensaje;
}


public class Transmision {
    private String titulo;
    private List<String> categoria ;
    private final LocalDateTime fechaDeInicio = LocalDateTime.now();
    private LocalDateTime fechaDeFinalizacion = null;
    private int participantesMaximos = 0;
    private int participantesActuales = 0;
    private List<Mensaje> historialDelChat = new ArrayList<>();

    public Transmision(String titulo, List<String> categoria) {
        this.titulo = titulo;
        this.categoria = categoria;
    }
    public void finalizar() {
        if(this.fechaDeFinalizacion != null) {
            throw new IllegalStateException("La transmision ya termino");
        }
       this.fechaDeFinalizacion = LocalDateTime.now();
    }

    public void recibirMensaje (Mensaje mensaje) {
        this.historialDelChat.add(mensaje);
    }

    public void añadirParticipante(){
        if( ++this.participantesActuales > this.participantesMaximos) {
            this.participantesMaximos = this.participantesActuales;
        }
    }
    public void removerParticipante() {
        this.participantesActuales--;
    }
}
```



```java
public class RepositorioDeCanales {
    private static final RepositorioDeCanales instancia = new RepositorioDeCanales();
    private List<Canal> canales = new ArrayList<>();

    private RepositorioDeCanales(){

    }

    public static RepositorioDeCanales getIntancia(){
        return instancia;
    }

    public Boolean contieneA(Canal canal){
        return canales.stream().anyMatch(c -> c.sonElMismo(canal));
    }

    public List<Canal> getCanales(){
        return new ArrayList<>(this.canales);
    }
    public  void  añadirCanal(Canal canal){
        if(this.contieneA(canal)){
            throw new IllegalArgumentException("Este canal  ya se encuentra registrado");
        }
        canales.add(canal);
    }
    public void removerCanal(Canal canal){
        if(!this.contieneA(canal)){
            throw new IllegalArgumentException("Este canal no se encuentra registrado");
        }
        canales.removeIf(c -> c.sonElMismo(canal));
    }
}
```

```java
public class Canal {
    private String nombreDelCanal; //Se asume que es unico
    private List<Transmision> transmicionesAnteriores = new ArrayList<Transmision>();
    private Transmision transmicionActual = null;
    private List<Canal> suscripciones = new ArrayList<Canal>();
    private Integer suscriptores = 0 ;
    private List<Integer> muestrasDeApoyoObtenidas = new ArrayList<Integer>();
    private Map<String, Integer> muestrasDeApoyoOtorgadas = new HashMap<>();

    public Canal(String nombreDelCanal) {
        this.nombreDelCanal = nombreDelCanal ;
    }

    public void iniciarTransmicion(String titulo, List<String> categoria) throws ExistingTransmitionExeption {
        if(transmicionActual != null) {
            throw new ExistingTransmitionExeption("Este canal ya posee una transmicion en curso");
        }
        transmicionActual = new Transmision(titulo,categoria);

    }
    public void terminarTransmicion() throws NonExistingTransmitionExeption {
        if(transmicionActual == null) {
            throw new NonExistingTransmitionExeption("Este canal no se encuentra transmitiendo");
        }
        transmicionActual.finalizar();
        transmicionesAnteriores.add(transmicionActual);
        transmicionActual = null;
    }
    public void otorgarApoyo(Canal canal, Integer apoyo) throws SupportAlreadyGivenE-xception {
        if(muestrasDeApoyoOtorgadas.containsKey(canal)){
            throw new SupportAlreadyGivenException("Ya se le otorgo apoyo a este canal");
        }
        muestrasDeApoyoOtorgadas.put(canal.nombreDelCanal , apoyo);
    }
    public void recibirApoyo(Integer apoyo){
        muestrasDeApoyoObtenidas.add(apoyo);
    }
    public void suscribirse(Canal canal) {
        suscripciones.add(canal);
    }
    public void aumentarSuscriptores () {
        suscriptores++;
    }
    public String getNombreDelCanal(){
        return this.nombreDelCanal;
    }
    public Boolean sonElMismo(Canal canal){
        return canal.getNombreDelCanal().equals(this.nombreDelCanal);
    }
}
```