

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

SELECCIÓN DE CONTENIDOS ONESEG A TRAVÉS DE CANAL DE RETORNO (ANEXOS)

Tesis para optar el Título de Ingeniero de las Telecomunicaciones, que
presenta el bachiller:

Diego Rodolfo Vargas Romero

ASESOR: MAGISTER Marco Antonio Mayorga Montoya

Lima, Febrero del 2013

ANEXO N° 1 – Parámetros del segmento OFDM

MODO		Modo 1		Modo 2		Modo 3	
Ancho de Banda		3000/7 = 428,57 KHz					
Separación entre frecuencias portadoras		250/63 kHz		125/63 KHz		125/126 KHz	
Numero de portadoras	Total	108	108	216	216	432	432
	Datos	96	96	192	192	384	384
	SP ^a	9	0	18	0	36	0
	CP ^a	0	1	0	1	0	1
	TMCC ^b	1	5	2	10	4	20
	AC1 ^c	2	2	4	4	8	8
	AC2 ^c	0	4	0	9	0	19
Esquema de modulación de la portadoras		QPSK 16 QAM 64 QAM	DQPSK	QPSK 16 QAM 64 QAM	DQPSK	QPSK 16 QAM 64 QAM	DQPSK
Símbolos por cuadro		204					
Tamaño de símbolo efectivo		252 us		504 us		1008 us	
Intervalo de guarda		63 us (1/4) 31,5 us (1/8) 15,75 us (1/16) 7,875 us (1/32)		126 us (1/4) 63 us (1/8) 31,5 us (1/16) 15,75 us (1/32)		252 us (1/4) 126 us (1/8) 63 us (1/16) 31,5 us (1/32)	
Longitud de cuadro		64,26 ms (1/4) 57,834 ms (1/8) 54,621 ms (1/16) 53.015 ms (1/32)		128,52 ms (1/4) 115,668 ms (1/8) 109,242 ms (1/16) 106,029 ms (1/32)		257,04 ms (1/4) 231,336 ms (1/8) 218,484 ms(1/16) 212,058 ms (1/32)	
Frecuencia de muestreo de la IFFT		512/63 = 8,12698 MHz					
Entrelazado interno		Código convolucional (1/2,2/3,3/4,5/6,7/8)					
Codificador externo		RS (204,188)					
^a SP y CP son usados por el receptor para fines de sincronización y demodulación ^b MCC es información de control ^c AC se usa para transmitir información adicional. AC1 está disponible en igual número en todos los segmentos, mientras que AC2 está disponible solamente en segmento de modulación diferencial.							

ANEXO N° 2 – Tasa de datos de un único segmento

Modulación de la portadora	Código Convolutional	Número de TSP transmitidos por cuadro	Tasa de datos Kbps ^a			
			Intervalo guarda 1/4	Intervalo guarda 1/8	Intervalo guarda 1/16	Intervalo guarda 1/32
DQPSK QPSK	1/2	12/24/48	280,85	312,06	330,42	340,43
	2/3	16/32/64	374,47	416,08	440,56	453,91
	3/4	18/36/72	421,28	468,09	495,63	510,65
	5/6	20/40/80	468,09	520,10	550,70	567,39
	7/8	21/42/84	491,50	546,11	578,23	595,76
16QAM	1/2	24/48/84	561,71	624,13	660,84	680,87
	2/3	32/64/128	748,95	832,17	881,12	907,82
	3/4	36/72/144	842,57	936,19	991,26	1021,30
	5/6	40/80/160	936,19	1040,21	1101,40	1134,78
	7/8	42/84/168	983,00	1092,22	1156,47	1191,52
64QAM	1/2	36/72/144	842,57	936,19	991,26	1021,30
	2/3	48/96/192	1123,43	1248,26	1321,68	1361,74
	3/4	54/108/216	1263,86	1404,29	1486,90	1531,74
	5/6	60/120/240	1404,29	1560,32	1652,11	1702,17
	7/8	63/126/252	1474,50	1638,34	1734,71	1787,28

^a Esa tasa de datos representa la tasa de datos(bits) por segmento para parámetros transmisión:
 tasa de datos (bits) = TSP transmitidos x 188 (bytes/TSP) x 8 (bits/byte) x 1/longitud del cuadro.

ANEXO N° 3 – Tasa total de datos para 13 segmentos

Modulación de la portadora	Código Convolutional	Número de TSP transmitidos (Modos 1/ 2/ 3)	Tasa de datos Kbps ^a			
			Intervalo guarda $\frac{1}{4}$	Intervalo guarda $\frac{1}{8}$	Intervalo guarda $\frac{1}{16}$	Intervalo guarda $\frac{1}{32}$
DQPSK QPSK	1/2	156/312/624	3,651	4,056	4,295	4,425
	2/3	208/416/832	4,868	5,409	5,727	5,900
	3/4	234/468/936	5,476	6,085	6,443	6,638
	5/6	260/520/1040	6,085	6,761	7,159	7,376
	7/8	273/546/1092	6,389	7,099	7,517	7,744
16QAM	1/2	312/624/1248	7,302	8,113	8,590	8,851
	2/3	416/832/1664	9,736	10,818	11,454	11,801
	3/4	468/936/1872	10,953	12,170	12,886	13,276
	5/6	520/1040/2080	12,170	13,522	14,318	14,752
	7/8	546/1092/2184	12,779	14,198	15,034	15,489
64QAM	1/2	468/936/1872	10,953	12,170	12,886	13,276
	2/3	624/1248/2496	14,604	16,227	17,181	17,702
	3/4	702/1404/2808	16,430	18,255	19,329	19,915
	5/6	780/1560/3120	18,255	20,284	21,477	22,128
	7/8	819/1638/3276	19,168	21,298	22,551	23,234

NOTA En esta tabla, los mismos parámetros se especifican para todos los 13 segmentos. La tasa total de datos durante la transmisión jerárquica varía dependiendo de los parámetros de configuración jerárquica. El volumen transmitido por los 13 segmentos es igual a la suma de todos los volúmenes de datos transmitidos por esos segmentos, que puede ser determinado con la tabla del Anexo N°3.

ANEXO N° 5 – Instalación de Opencaster y FFmpeg

Avalpa Broadcasting Server es el nombre que recibe el servidor Broadcasting con dirección 192.168.1.100 y sistema operativo Linux Ubuntu 10.10 bajo la normativa GPL de nuestra red. Este servidor cumple las funciones de generación de contenidos ISDB-Tb, creación de EPG y de transmisión de contenidos para televisión digital terrestre. La instalación del OpenCaster se desarrollará bajo las recomendaciones del instituto LIFIA. Se procederá a descargar la versión OpenCaster2.4 del repositorio: <ftp://tvd.lifia.info.unlp.edu.ar/OpenCaster2.4/>. Posteriormente se colocará el archivo dentro de un directorio para su posterior instalación.

En esta carpeta encontraremos los siguientes archivos:

OpenCaster2.4.tgz.- Contiene la distribución original de Opencaster 2.4 distribuida por AVALPA. OpenCaster2.4-lifia-rev362.patch.- Es el archivo usado para aplicar los cambios de LIFIA a la distribución.

1.-Descomprimir la versión original de las fuentes, para esto debemos ubicarnos en el directorio donde se descargó: `$ tar zvf OpenCaster2.4.tgz`

2.-Instalar gcc: `$ sudo apt-get install build-essential`

3.-Instalar python: `$sudo apt-get install python-dev`

4.-Siguiente paso es ejecutar el parche desarrollado por LIFIA: `$cd OpenCaster2.4/
$patch -p1 <OpenCaster2.4-lifia-rev362.patch>`

5.-Instalar con privilegios root OpenCaster: `$sudo apt-get install`

6.-Una vez instalado, se prueba su correcto funcionamiento: `$python -c "from
dvbobjects.PSI.PAT import*"`

La instalación FFmpeg se desarrollará bajo los estándares GPL para la creación de contenidos de audio y video codificados bajo la norma H264 y AA-HEC respectivamente.

`$sudo apt-get install ffmpeg`

ANEXO N° 6 – Selección de Frecuencia de transmisión

Se determinará la frecuencia central con el uso de la siguiente formula:

$$f_c = 473 + 6(n - 14) + \frac{1}{7}$$



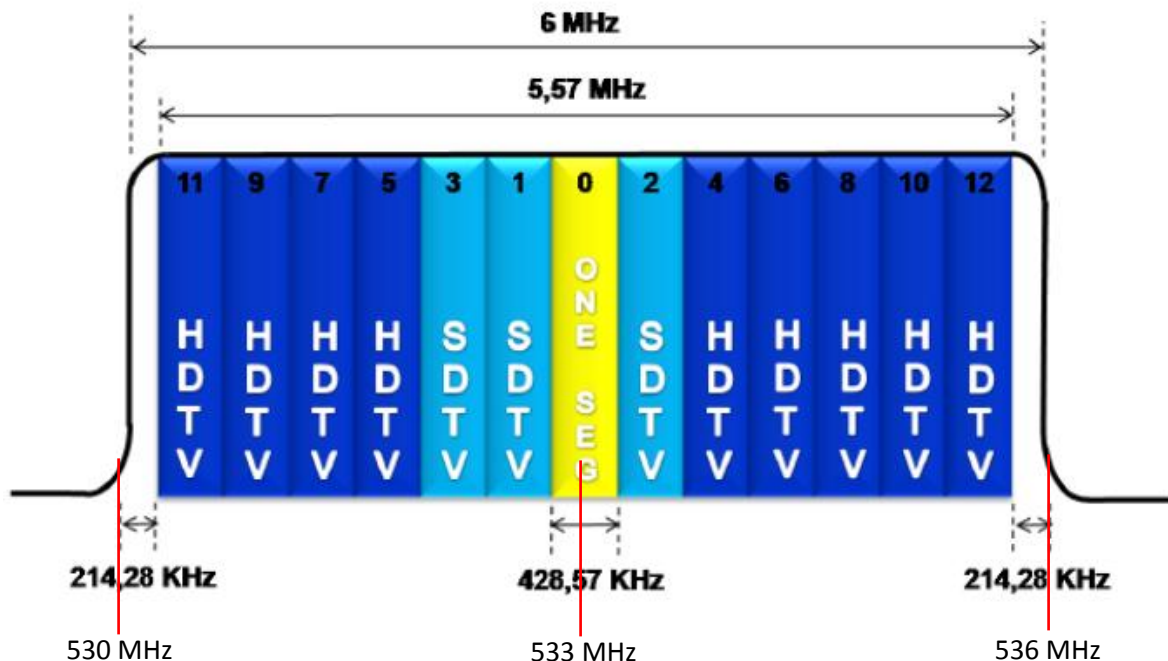
Bandas VHF y UHF

La regulación peruana define parte de las bandas VHF y UHF para el servicio de televisión terrestre analógica, sin embargo para televisión digital se utilizara solo la banda UHF ya que la banda VHF será utilizada para nuevas tecnologías de telefonía celular.

El contenido de televisión digital se transmitirá en la banda de frecuencias 626-632 MHz. En la frecuencia central 629 MHz se ubica el segmento onseg para transmisión celular.

Canal 40:

$$f_c = 629.143$$



Lista de canales de televisión con licencia en el país:

Canal	Razon Social	Norma	Indicativo on_id	Máxima e.r.p. (KW)	Estado
15	ALLIANCE S.A.C.	NTSC-M	OBV-4K	1000	AUTORIZADO
16	INSTITUTO NACIONAL DE RADIO Y TELEVISION DEL PERU - IRTP	ISDB-T	1852	240	AUTORIZADO
17	MISION PAX TV	NTSC-M	OAS-4G	1000	AUTORIZADO
18	ANDINA DE RADIODIFUSION S.A.C.	ISDB-T	1867	240	AUTORIZADO
19	ASOCIACION LAS MANOS DE DIOS	NTSC-M	OAS-4E	1000	AUTORIZADO
20	COMPAÑIA LATINOAMERICANA DE RADIODIFUSION S.A.	ISDB-T	1866	240	AUTORIZADO
22	EMPRESA RADIODIFUSORA 1160 S.A.	ISDB-T	154E	240	AUTORIZADO
23	TELEVISION NACIONAL PERUANA S.A.C.	NTSC-M	OAS-4M	1000	AUTORIZADO
24	COMPAÑIA PERUANA DE RADIODIFUSION S.A.	ISDB-T	1851	240	AUTORIZADO
25	ASOCIACION CULTURAL BETHEL	NTSC-M	OBV-4M	1000	AUTORIZADO
26	PANAMERICANA TELEVISION S.A.	ISDB-T	184E	240	AUTORIZADO
27	UNIVERSAL DE TELEVISION S.A.C.	NTSC-M	OAY-4G	1000	AUTORIZADO
28	ALLIANCE S.A.C.	ISDB-T	2F58	240	AUTORIZADO
29		ISDB-T		240	
30		ISDB-T		240	
31		ISDB-T		240	
32	TELEVISION NACIONAL PERUANA S.A.C.	ISDB-T	125A	240	AUTORIZADO
33	EMPRESA INTERAMERICANA DE RADIODIFUSION S.A.	NTSC-M	OAV-4Z	1000	AUTORIZADO
34	ASOCIACION CULTURAL ENTIDADES LATINOAMERICANAS COMUNICANDO EL EVANGELIO - ENLACE	ISDB-T	0B67	240	AUTORIZADO
36	ASOCIACION CULTURAL BETHEL	ISDB-T	2F5A	240	AUTORIZADO
38	RED BICOLOR DE COMUNICACIONES S.A.A.	ISDB-T	325F	240	AUTORIZADO
39	CARRETERO RAZA OSCAR GROVER	NTSC-M	OAL-4S	1000	AUTORIZADO
40		ISDB-T		240	
41	ASOCIACION CULTURAL ONDAS DEL PERU S.A.C.	NTSC-M	OCV-4R	1000	AUTORIZADO
42		ISDB-T		240	
43	CABALLERO ENRIQUEZ MARITZA	NTSC-M	OBL-4J	1000	AUTORIZADO
44		ISDB-T		240	
45	CENTRO DE COMUNICACION POPULAR Y PROMOCION DEL DESARROLLO DE VILLA EL SALVADOR - CECOPRODE-VES	NTSC-M	OAY-4F	1000	AUTORIZADO
46		ISDB-T		240	
47		ISDB-T		240	
48		ISDB-T		240	
49	MINISTERIO LA LUZ	NTSC-M	OBL-4A	1000	AUTORIZADO
50		ISDB-T		240	
51	EMPRESA TELED DE RADIODIFUSION S.A. CANAL 51UHF	NTSC-M	OBV-4I	1000	AUTORIZADO
53	ANDINA DE RADIODIFUSION S.A.C.	NTSC-M	OAL-4R	1000	AUTORIZADO
55	ASOCIACION VIDA TELEVISION (A.V.T.)	NTSC-M	OCV-4E	1000	AUTORIZADO
57	ASOCIACION CULTURAL ENTIDADES LATINOAMERICANAS COMUNICANDO EL EVANGELIO - ENLACE	NTSC-M	OAL-4Z	1000	AUTORIZADO
59	JESUS BROADCASTING NETWORK S.A.C.	NTSC-M	OAL-4V	1000	AUTORIZADO



ANEXO N°7 - Creación de BD sms_db

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `sms_db` DEFAULT CHARACTER SET latin1 ;
USE `sms_db` ;

-----
-- Table `sms_db`.`option_sms`
-----
CREATE TABLE IF NOT EXISTS `sms_db`.`option_sms` (
  `option_id` VARCHAR(50) NOT NULL ,
  `program_name` TEXT NOT NULL ,
  `file_route` TEXT NOT NULL ,
  PRIMARY KEY (`option_id`))

ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `sms_db`.`sms_in`
-----
CREATE TABLE IF NOT EXISTS `sms_db`.`sms_in` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `sms_text` VARCHAR(50) NOT NULL ,
  `sender_number` VARCHAR(45) NOT NULL ,
  `sent_dt` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`, `sms_text`),
  INDEX `fk_sms_in_option_sms` (`sms_text` ASC),
  CONSTRAINT `fk_sms_in_option_sms`
  FOREIGN KEY (`sms_text`)
  REFERENCES `sms_db`.`option_sms` (`option_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```


ANEXO N°8 – Diagramas UML

En el paquete beans se define las clases ProgramBean y SMSBean, estas clases representan los atributos de las tablas option_sms y sms_in respectivamente.

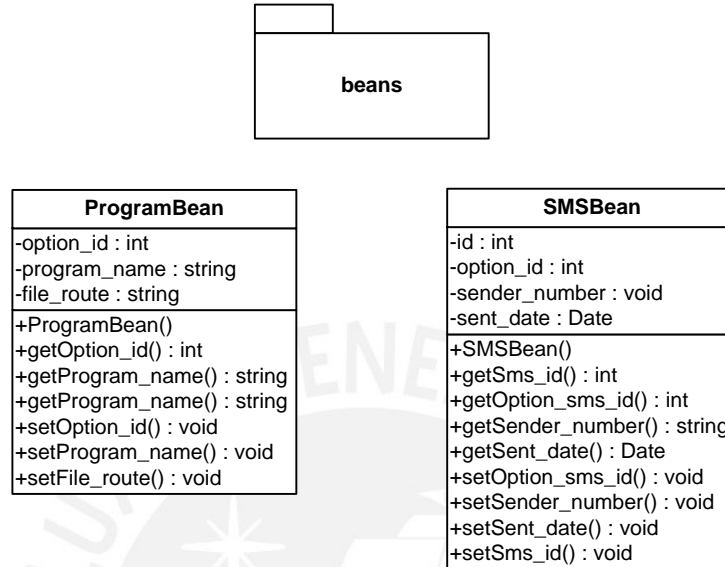


Diagrama UML paquete beans

La clase BaseDAO define la cadena de conexión ("jdbc:mysql://127.0.0.1:3306/sms_db","root", "root") a la base de datos SMS_db a través de la consulta al archivo database.properties.

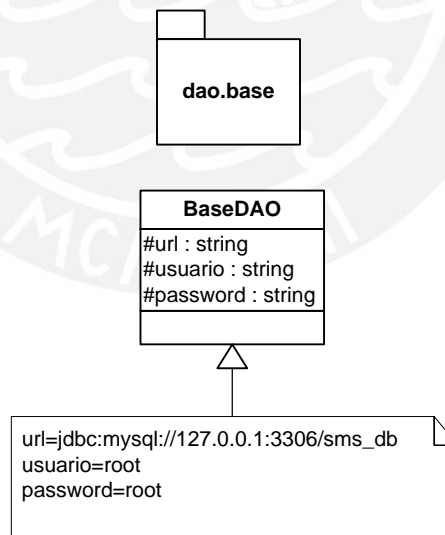


Diagrama UML paquete dao.base

El paquete dto contiene la clase Result, donde se especifica los campos a ser mostrados en la interfaz gráfica al usuario.

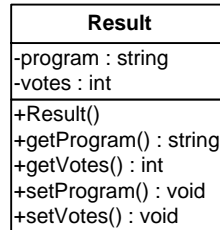
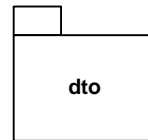


Diagrama UML paquete dto

La clase FileActions se encarga de realizar la acción de copiar el archivo del programa con mayor número de votos a la carpeta de transmisión. Esta acción será realizada por la función copyFile().

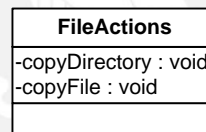


Diagrama UML paquete fileActions

El paquete daos contiene la clase ProgramDAO, encargado de extraer la data de la base de datos sms_db. En la función election() se obtiene el número de votos de cada programa ordenados de forma descendente. La función getFile() obtiene la ruta del programa con mayor número de votos.

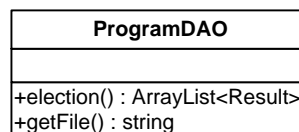
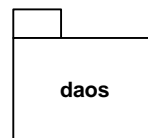


Diagrama UML paquete daos

La clase SMSWindow genera la interfaz gráfica utilizada por el administrador, esta clase ejecutará dos acciones: Obtener resultado y generar archivo.

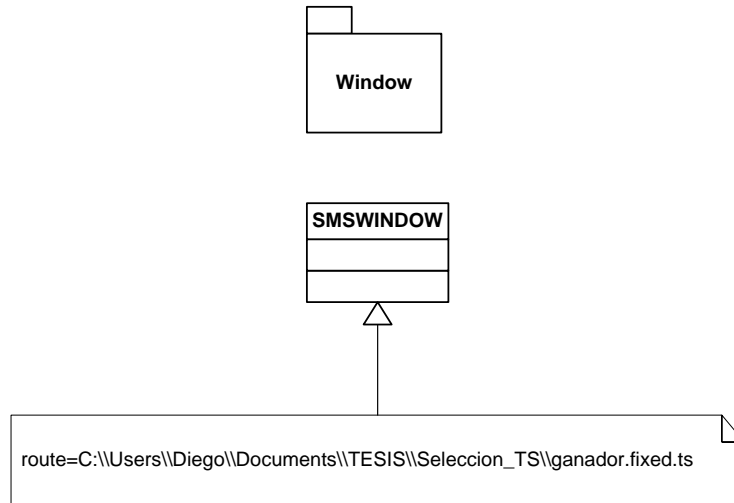


Diagrama UML paquete Window



ANEXO N°9 - Interface Gráfica de Administración

1) Clase: beans.ProgramBean.java

```
package beans;

public class ProgramBean {

    private int option_id;
    private String program_name;
    private String file_route;

    public ProgramBean() {
    }

    public ProgramBean(int option_id, String program_name) {
        this.option_id = option_id;
        this.program_name = program_name;
    }

    public int getOption_id() {
        return option_id;
    }

    public void setOption_id(int option_id) {
        this.option_id = option_id;
    }

    public String getProgram_name() {
        return program_name;
    }

    public void setProgram_name(String program_name) {
        this.program_name = program_name;
    }

    public String getFile_route() {
        return file_route;
    }

    public void setFile_route(String file_route) {
        this.file_route = file_route;
    }
}
```

2) Clase: beans.SMSBean.java

```
package beans;

import java.util.Date;
```

```
public class SMSBean {

    private int id;
    private int option_id;
    private String sender_number;
    private Date sent_date;

    public SMSBean() {
    }

    public SMSBean(int sms_id, int option_id, String sender_number, Date sent_date) {
        this.id = sms_id;
        this.option_id = option_id;
        this.sender_number = sender_number;
        this.sent_date = sent_date;
    }

    public int getOption_sms_id() {
        return option_id;
    }

    public void setOption_sms_id(int option_sms_id) {
        this.option_id = option_sms_id;
    }

    public String getSender_number() {
        return sender_number;
    }

    public void setSender_number(String sender_number) {
        this.sender_number = sender_number;
    }

    public Date getSent_date() {
        return sent_date;
    }

    public void setSent_date(Date sent_date) {
        this.sent_date = sent_date;
    }

    public int getSms_id() {
        return id;
    }

    public void setSms_id(int sms_id) {
        this.id = sms_id;
    }
}
```


3) Clase: dao.base.BaseDAO.java

```
package dao.base;

import java.util.ResourceBundle;

public abstract class BaseDAO {

    protected String url;
    protected String usuario;
    protected String password;

    public BaseDAO() {
        ResourceBundle bundle = ResourceBundle.getBundle("dao.base.database");
        url = bundle.getString("url").trim();
        usuario = bundle.getString("usuario").trim();
        password = bundle.getString("password").trim();
    }
}
```

4) Archivo: dao.base.database.properties

```
url=jdbc:mysql://127.0.0.1:3306/sms_db
usuario=root
password=root
```

5) Clase: daos.ProgramDAO.java

```
package daos;

import dao.base.BaseDAO;
import dto.Result;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
public class ProgramDAO extends BaseDAO {

    public ArrayList<Result> election() throws SQLException {

        ArrayList<Result> result = new ArrayList<Result>();
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try {

            conn = DriverManager.getConnection(url, usuario, password);
```

```

stmt = conn.createStatement();
String sql = "select option_sms.program_name as Programa, Count(*) as Votos"
    + " from sms_in, option_sms"
    + " where sms_in.sms_text=option_sms.option_id"
    + " group by option_sms.option_id order by Votos desc";

rs = stmt.executeQuery(sql);
while (rs.next()) {
    Result r = new Result();
    r.setProgram(rs.getString(1));
    r.setVotes(rs.getInt(2));
    result.add(r);
}

} catch (SQLException ex) {
    Logger.getLogger(ProgramDAO.class.getName()).log(Level.SEVERE, null, ex);
} finally {
    if (rs != null) {
        rs.close();
        rs = null;
    }
    if (stmt != null) {
        stmt.close();
        stmt = null;
    }
    if (conn != null) {
        conn.close();
        conn = null;
    }
}
return result;
}

public String getFile() throws SQLException {

String result = "";
Connection conn = null;
Statement stmt = null;
ResultSet rs = null;

try {

    conn = DriverManager.getConnection(url, usuario, password);
    stmt = conn.createStatement();
    String sql = "select option_sms.file_route "
        + "from sms_in, option_sms "
        + "where sms_in.sms_text=option_sms.option_id "
        + "group by option_sms.option_id order by Count(*) desc "
        + "limit 1";

```

```

        rs = stmt.executeQuery(sql);
        if (rs.next()) {
            result = rs.getString(1);
        }

    } catch (SQLException ex) {
        Logger.getLogger(ProgramDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        if (rs != null) {
            rs.close();
            rs = null;
        }
        if (stmt != null) {
            stmt.close();
            stmt = null;
        }
        if (conn != null) {
            conn.close();
            conn = null;
        }
        return result;
    }
}
}
}

```

6) Clase: dto.Result.java

```

package dto;

public class Result {

    private String program;
    private int votes;

    public Result() {
    }

    public Result(String program, int votes) {
        this.program = program;
        this.votes = votes;
    }

    public String getProgram() {
        return program;
    }

    public void setProgram(String program) {
        this.program = program;
    }

    public int getVotes() {

```

```
        return votes;
    }

    public void setVotes(int votes) {
        this.votes = votes;
    }
}
```

7) Clase: fileActions.FileActions.java

```
package fileActions;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class FileActions {

    public void copyDirectory(File srcDir, File dstDir) {
        try {
            if (srcDir.isDirectory()) {
                if (!dstDir.exists()) {
                    dstDir.mkdir();
                }

                String[] children = srcDir.list();
                for (int i = 0; i < children.length; i++) {
                    copyDirectory(new File(srcDir, children[i]), new File(dstDir, children[i]));
                }
            } else {
                copyFile(srcDir, dstDir);
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }

    public void copyFile(File src, File dst) throws IOException {

        dst.delete();
        InputStream in = new FileInputStream(src);
        OutputStream out = new FileOutputStream(dst);

        byte[] buf = new byte[1024];
        int len;
        while ((len = in.read(buf)) > 0) {
```

```

        out.write(buf, 0, len);
    }
    in.close();
    out.close();
}
}

```

8) Clase: Window.SMSWINDOW.java

```
package Window;
```

```

import daos.ProgramDAO;
import dto.Result;
import fileActions.FileActions;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

```

```
public class SMSWINDOW extends javax.swing.JFrame {
```

```

    public SMSWINDOW() {
        initComponents();
        setResizable(false);
    }

```

```

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

```

```

        jScrollPane1 = new javax.swing.JScrollPane();
        tabla = new javax.swing.JTable();
        titulo = new javax.swing.JLabel();
        Button = new javax.swing.JButton();
        InsertFile = new javax.swing.JButton();

```

```
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```

        tabla.setAutoCreateRowSorter(true);
        tabla.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "",
javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.TOP));
        tabla.setFont(new java.awt.Font("Tahoma", 1, 11));
        tabla.setForeground(new java.awt.Color(0, 0, 255));
        tabla.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null},
                {null, null, null},

```



```

        {null, null, null},
        {null, null, null}
    },
    new String [] {
        "ID", "Programa", "Votos"
    }
) {
    Class[] types = new Class [] {
        java.lang.Integer.class, java.lang.String.class, java.lang.String.class
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }
});
jScrollPane1.setViewportView(tabla);

titulo.setFont(new java.awt.Font("Tahoma", 1, 12));
titulo.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
titulo.setText("Resultados de la Semana:");

Button.setText("Obtener Resultado");
Button.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        ButtonMouseClicked(evt);
    }
});
Button.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ButtonActionPerformed(evt);
    }
});

InsertFile.setText("Generar Archivo");
InsertFile.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        InsertFileMouseClicked(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addGroup(layout.createSequentialGroup()
                            .addContainerGap()
                            .addComponent(Button, javax.swing.GroupLayout.PREFERRED_SIZE, 147,
                                javax.swing.GroupLayout.PREFERRED_SIZE)
                        )
                    )
                )
            )

```

```

        .addGap(18, 18, 18)
        .addComponent(InsertFile, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup())
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(titulo)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
423, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(24, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(titulo)
            .addGap(41, 41, 41)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 47,
Short.MAX_VALUE)
        );

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(InsertFile)
        .addComponent(Button))
        .addGap(22, 22, 22)
);

pack();
} // </editor-fold>

private void ButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void ButtonMouseClicked(java.awt.event.MouseEvent evt) {
    if (evt.getSource().equals(Button)) {
        ProgramDAO dAO = new ProgramDAO();
        JOptionPane jOptionPane = new JOptionPane();
        try {
            Object data[][] = new Object[3][3];
            int i = 0;
            for (Result r : dAO.election()) {
                data[i][0] = i + 1;
                data[i][1] = r.getProgram();
                data[i][2] = r.getVotes();
                i++;
            }
            tabla.setModel(new javax.swing.table.DefaultTableModel(

```

```

        data,
        new String[]{
            "ID", "Programa", "Votos"
        }) {

        Class[] types = new Class[]{
            java.lang.Integer.class, java.lang.String.class, java.lang.String.class
        };
    });

    for (Result r : dAO.election()) {
    }
} catch (SQLException ex) {
    Logger.getLogger(SMSWINDOW.class.getName()).log(Level.SEVERE, null, ex);
    JOptionPane.showConfirmDialog(this, "ERROR. Revisar el
Log", "Respuesta", JOptionPane.ERROR_MESSAGE);
}

}

}

private void InsertFileMouseClicked(java.awt.event.MouseEvent evt) {
    if (evt.getSource().equals(InsertFile)) {
        JOptionPane JOptionPane = new JOptionPane();
        try {
            FileActions actions = new FileActions();
            ResourceBundle bundle = ResourceBundle.getBundle("Window.route_selection_TS");
            ProgramDAO dAO = new ProgramDAO();
            String origen_file = dAO.getFile();
            String destination_file = bundle.getString("route").trim();
            File file1 = new File(origen_file);
            File file2 = new File(destination_file);
            actions.copyFile(file1, file2);
            JOptionPane.showConfirmDialog(this, "Se copio el archivo
correctamente", "Respuesta", JOptionPane.PLAIN_MESSAGE);
        } catch (IOException ex) {
            Logger.getLogger(SMSWINDOW.class.getName()).log(Level.SEVERE, null, ex);
            JOptionPane.showConfirmDialog(this, "ERROR. Revisar el
Log", "Respuesta", JOptionPane.ERROR_MESSAGE);
        } catch (SQLException ex) {
            Logger.getLogger(SMSWINDOW.class.getName()).log(Level.SEVERE, null, ex);
            JOptionPane.showConfirmDialog(this, "ERROR. Revisar el
Log", "Respuesta", JOptionPane.ERROR_MESSAGE);
        }
    }
}

public static void main(String args[]) {

```

```
java.awt.EventQueue.invokeLater(new Runnable() {  
  
    public void run() {  
        new SMSWINDOW().setVisible(true);  
    }  
});  
}  
// Variables declaration - do not modify  
private javax.swing.JButton Button;  
private javax.swing.JButton InsertFile;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable tabla;  
private javax.swing.JLabel titulo;  
// End of variables declaration  
}  
  
9) Archivo: Window.route_selection_TS.properties  
  
route=C:\\Users\\Diego\\Documents\\TESIS\\Seleccion_TS\\ganador.fixed.ts  
  
10) Clase: Main  
  
package sms;  
  
import Window.SMSWINDOW;  
import java.sql.SQLException;  
  
public class Main {  
  
    public static void main(String[] args) throws SQLException {  
  
        SMSWINDOW smswindow = new SMSWINDOW();  
        smswindow.main(args);  
    }  
}
```

ANEXO N°10 - Aplicación votación

1) votación.ncl :

```

<ncl id="myNCLDocID" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <connectorBase id="connBaseId">
      <importBase alias="conn" documentURI="defaultConnBase.ncl">
        </importBase>
      </connectorBase>
      <regionBase id="rgbase1">
        <region height="100.00%" id="rgvideo" left="0.00%" top="0.00%"
width="100.00%">
          </region>
          <region height="19.00%" id="rgmensaje" left="70.00%" top="10.00%"
width="28.00%" zIndex="2">
            </region>
            <region height="25.00%" id="rginteractividad" left="80.00%"
top="10.00%" width="18.75%" zIndex="2">
              </region>
              <region height="8.00%" id="rgmenu1" left="60%" top="25.00%"
width="35.00%" zIndex="2">
                </region>
                <region height="8.00%" id="rgmenu2" left="60.00%" top="40.00%"
width="35.00%" zIndex="2">
                  </region>
                  <region height="8.00%" id="rgmenu3" left="60%" top="55%"
width="35.00%" zIndex="2">
                    </region>
                    <region height="8.00%" id="rgmenu4" left="60%" top="70%"
width="35.00%" zIndex="2">
                      </region>
                      <region height="20.00%" id="rgopcionA" left="15%" top="75.00%"
width="17.44%" zIndex="2">
                        </region>
                        <region height="20.00%" id="rgopcionB" left="45%" top="75.00%"
width="17.44%" zIndex="2">
                          </region>
                          <region height="5.00%" id="rgvolverMenu" left="75%" top="90.00%"
width="15%" zIndex="2">
                            </region>
                            <region height="31.46%" id="rglua" left="0.00%" top="0.00%"
width="100.00%">
                              </region>
                              <region height="78.50%" id="rgcanal" left="49.88%" top="10.62%"
width="50.00%" zIndex="2">
                                </region>
                                <region height="7.08%" id="rgregresarmenu" left="82.79%"
top="77.92%" width="14.64%" zIndex="3">
                                  </region>

```



```

        <region height="30.00%" id="rgexito" left="62.06%" top="70.00%"
width="30.00%" zIndex="3">
        </region>
        <region height="25.00%" id="rgprograma" left="5%" top="75%"
width="90.00%">
        </region>
        <region height="10.83%" id="rgenviarSMS" left="59.48%"
top="83.33%" width="19.60%" zIndex="2">
        </region>
        <region height="10.83%" id="rgsalir_envio" left="84.54%"
top="83.33%" width="7.50%" zIndex="2">
        </region>
    </regionBase>
    <descriptorBase id="descriptorBase1">
        <descriptor id="descregion" region="rgvideo">
        </descriptor>
        <descriptor id="descmensaje" region="rgmensaje">
        </descriptor>
        <descriptor id="descinteractividad" region="rginteractividad">
        </descriptor>
        <descriptor focusBorderColor="green" focusBorderWidth="10"
focusIndex="1" id="descmenu1" moveDown="2" moveUp="4" region="rgmenu1">
        </descriptor>
        <descriptor focusBorderColor="green" focusBorderWidth="10"
focusIndex="2" id="descmenu2" moveDown="3" moveUp="1" region="rgmenu2">
        </descriptor>
        <descriptor focusBorderColor="green" focusBorderWidth="10"
focusIndex="3" id="descmenu3" moveDown="4" moveUp="2" region="rgmenu3">
        </descriptor>
        <descriptor focusBorderColor="green" focusBorderWidth="10"
focusIndex="4" id="descmenu4" moveDown="1" moveUp="3" region="rgmenu4">
        </descriptor>
        <descriptor focusBorderColor="yellow" focusBorderWidth="10"
focusIndex="5" id="descopcionA" moveLeft="7" moveRight="6" region="rgopcionA">
        </descriptor>
        <descriptor focusBorderColor="yellow" focusBorderWidth="10"
focusIndex="6" id="descopcionB" moveLeft="5" moveRight="7" region="rgopcionB">
        </descriptor>
        <descriptor focusBorderColor="yellow" focusBorderWidth="10"
focusIndex="7" id="descregresarmenu" moveLeft="6" moveRight="5"
region="rgvolverMenu">
        </descriptor>
        <descriptor id="desclua" region="rglua">
        </descriptor>
        <descriptor id="desccanal" region="rgcanal">
        </descriptor>
        <descriptor focusBorderColor="red" focusBorderWidth="10"
focusIndex="8" id="descregresar_menu" moveDown="8" moveLeft="8" moveRight="8"
moveUp="8" region="rgregresarmenu" selBorderColor="green">
        </descriptor>
        <descriptor explicitDur="2s" id="desexito" region="rgexito">

```

```

    </descriptor>
    <descriptor id="descprogramaA" region="rgprograma">
    </descriptor>
    <descriptor id="descprogramaB" region="rgprograma">
    </descriptor>
    <descriptor focusBorderColor="green" focusBorderWidth="8"
focusIndex="9" id="descenviarA" moveLeft="10" moveRight="10" region="rgenviarSMS">
    </descriptor>
    <descriptor focusBorderColor="green" focusBorderWidth="8"
focusIndex="11" id="descenviarB" moveLeft="12" moveRight="12" region="rgenviarSMS">
    </descriptor>
    <descriptor focusBorderColor="green" focusBorderWidth="8"
focusIndex="10" id="descsalir_envioA" moveLeft="9" moveRight="9" region="rgsalir_envio">
    </descriptor>
    <descriptor focusBorderColor="green" focusBorderWidth="8"
focusIndex="12" id="descsalir_envioB" moveLeft="11" moveRight="11"
region="rgsalir_envio">
    </descriptor>
  </descriptorBase>
</head>
<body id="myBodyID">
  <media descriptor="descregion" id="comercial" src="sbtvd-ts://0">
    <area begin="3s" end="8s" id="area1">
    </area>
    <area begin="10s" id="area2">
    </area>
    <property name="bounds">
    </property>
  </media>
  <media descriptor="descinteractividad" id="icono_interactividad"
src="media/imagen/icono_interactividad.png">
  </media>
  <media descriptor="descmensaje" id="mensaje"
src="media/imagen/mensaje.png">
  </media>
  <port component="comercial" id="p1">
  </port>
  <link id="link1" xconnector="conn#onBeginStart">
    <bind component="comercial" interface="area1" role="onBegin">
    </bind>
    <bind component="mensaje" role="start">
    </bind>
  </link>
  <link id="link2" xconnector="conn#onEndStop">
    <bind component="comercial" interface="area1" role="onEnd">
    </bind>
    <bind component="mensaje" role="stop">
    </bind>
  </link>
  <link id="link3" xconnector="conn#onBeginStart">
    <bind component="comercial" interface="area2" role="onBegin">

```

```

    </bind>
    <bind component="icono_interactividad" role="start">
    </bind>
</link>
<media descriptor="descmenu4" id="menu4" src="media/imagen/menu4.png">
</media>
<media descriptor="descmenu1" id="menu1" src="media/imagen/menu1.png">
</media>
<media descriptor="descmenu2" id="menu2" src="media/imagen/menu2.png">
</media>
<media descriptor="descmenu3" id="menu3" src="media/imagen/menu3.png">
</media>
<link id="link4" xconnector="conn#onKeySelectionStopStart">
    <bind component="icono_interactividad" role="onSelection">
        <bindParam name="keyCode" value="RED">
        </bindParam>
    </bind>
    <bind component="icono_interactividad" role="stop">
    </bind>
    <bind component="menu1" role="start">
    </bind>
    <bind component="menu2" role="start">
    </bind>
    <bind component="menu3" role="start">
    </bind>
    <bind component="menu4" role="start">
    </bind>
</link>
<link id="link15" xconnector="conn#onSelectionStartStop">
    <bind component="menu4" role="onSelection">
    </bind>
    <bind component="menu1" role="stop">
    </bind>
    <bind component="menu2" role="stop">
    </bind>
    <bind component="menu3" role="stop">
    </bind>
    <bind component="menu4" role="stop">
    </bind>
    <bind component="icono_interactividad" role="start">
    </bind>
</link>
    <media descriptor="descopcionA" id="opcionA"
src="media/imagen/opcionA.png">
    </media>
    <media descriptor="descopcionB" id="opcionB"
src="media/imagen/opcionB.png">
    </media>
    <media descriptor="descregresarmenu" id="regresar_menu"
src="media/imagen/regresar_menu.png">
    </media>

```

```

<link id="link27" xconnector="conn#onSelectionStartStop">
  <bind component="menu3" role="onSelection">
  </bind>
  <bind component="menu1" role="stop">
  </bind>
  <bind component="menu2" role="stop">
  </bind>
  <bind component="menu3" role="stop">
  </bind>
  <bind component="menu4" role="stop">
  </bind>
  <bind component="opcionA" role="start">
  </bind>
  <bind component="opcionB" role="start">
  </bind>
  <bind component="regresar_menu" role="start">
  </bind>
</link>
<link id="link29" xconnector="conn#onSelectionStopStart">
  <bind component="regresar_menu" role="onSelection">
  </bind>
  <bind component="opcionA" role="stop">
  </bind>
  <bind component="opcionB" role="stop">
  </bind>
  <bind component="regresar_menu" role="stop">
  </bind>
  <bind component="menu1" role="start">
  </bind>
  <bind component="menu2" role="start">
  </bind>
  <bind component="menu3" role="start">
  </bind>
  <bind component="menu4" role="start">
  </bind>
</link>
<link id="link30" xconnector="conn#onSelectionSet_var">
  <bind component="menu3" role="onSelection">
  </bind>
  <bind component="comercial" interface="bounds" role="set">
    <bindParam name="var" value="20%,10%,60%,60%">
    </bindParam>
  </bind>
</link>
<link id="link31" xconnector="conn#onSelectionSet_var">
  <bind component="regresar_menu" role="onSelection">
  </bind>
  <bind component="comercial" interface="bounds" role="set">
    <bindParam name="var"
value="0%,0%,166.666666%,166.666666%">
    </bindParam>

```

```

        </bind>
    </link>
    <media descriptor="desclua" id="opcion1" src="media/lua/opcion1.lua"
type="application/x-ginga-NCLua">
    </media>
    <media descriptor="desc canal" id="canal" src="media/imagen/canal.png">
    </media>
    <media descriptor="desc regresar_menu" id="regresar"
src="media/imagen/regresar.png">
    </media>
    <link id="link44" xconnector="conn#onSelectionSetStartStop">
        <bind component="menu1" role="onSelection">
        </bind>
        <bind component="comercial" interface="bounds" role="set">
            <bindParam name="var" value="0%,25%,50%,50%">
            </bindParam>
        </bind>
        <bind component="menu1" role="stop">
        </bind>
        <bind component="menu2" role="stop">
        </bind>
        <bind component="menu3" role="stop">
        </bind>
        <bind component="menu4" role="stop">
        </bind>
        <bind component="canal" role="start">
        </bind>
        <bind component="regresar" role="start">
        </bind>
    </link>
    <link id="link45" xconnector="conn#onSelectionSetStartStop">
        <bind component="regresar" role="onSelection">
        </bind>
        <bind component="canal" role="stop">
        </bind>
        <bind component="regresar" role="stop">
        </bind>
        <bind component="comercial" interface="bounds" role="set">
            <bindParam name="var" value="0%,0%,200%,200%">
            </bindParam>
        </bind>
        <bind component="menu1" role="start">
        </bind>
        <bind component="menu2" role="start">
        </bind>
        <bind component="menu3" role="start">
        </bind>
        <bind component="menu4" role="start">
        </bind>
    </link>

```



```

    <media descriptor="descprogramaA" id="programaA"
src="media/imagen/programaA.png">
    </media>
    <link id="link59" xconnector="conn#onSelectionStopStart">
        <bind component="opcionA" role="onSelection">
        </bind>
        <bind component="opcionA" role="stop">
        </bind>
        <bind component="opcionB" role="stop">
        </bind>
        <bind component="regresar_menu" role="stop">
        </bind>
        <bind component="programaA" role="start">
        </bind>
        <bind component="enviarsmsA" role="start">
        </bind>
        <bind component="salir_envioA" role="start">
        </bind>
    </link>
    <media descriptor="descprogramaB" id="programaB"
src="media/imagen/programaB.png">
    </media>
    <link id="link61" xconnector="conn#onSelectionStopStart">
        <bind component="opcionB" role="onSelection">
        </bind>
        <bind component="opcionA" role="stop">
        </bind>
        <bind component="opcionB" role="stop">
        </bind>
        <bind component="regresar_menu" role="stop">
        </bind>
        <bind component="programaB" role="start">
        </bind>
        <bind component="enviarsmsB" role="start">
        </bind>
        <bind component="salir_envioB" role="start">
        </bind>
    </link>
    <media descriptor="descenviarA" id="enviarsmsA"
src="media/imagen/enviarsms.png">
    </media>
    <media descriptor="descenviarB" id="enviarsmsB"
src="media/imagen/enviarsms.png">
    </media>
    <media descriptor="descsalir_envioA" id="salir_envioA"
src="media/imagen/salir_envio.png">
    </media>
    <media descriptor="descsalir_envioB" id="salir_envioB"
src="media/imagen/salir_envio.png">
    </media>
    <link id="link62" xconnector="conn#onSelectionStopStart">

```

```

    <bind component="salir_envioB" role="onSelection">
    </bind>
    <bind component="programaB" role="stop">
    </bind>
    <bind component="enviarsmsB" role="stop">
    </bind>
    <bind component="salir_envioB" role="stop">
    </bind>
    <bind component="opcionA" role="start">
    </bind>
    <bind component="opcionB" role="start">
    </bind>
    <bind component="regresar_menu" role="start">
    </bind>
  </link>
  <link id="link63" xconnector="conn#onSelectionStartStop">
    <bind component="salir_envioA" role="onSelection">
    </bind>
    <bind component="programaA" role="stop">
    </bind>
    <bind component="enviarsmsA" role="stop">
    </bind>
    <bind component="salir_envioA" role="stop">
    </bind>
    <bind component="opcionA" role="start">
    </bind>
    <bind component="opcionB" role="start">
    </bind>
    <bind component="regresar_menu" role="start">
    </bind>
  </link>
  <media descriptor="desclua" id="opcion2" src="media/lua/opcion2.lua"
type="application/x-ginga-NCLua">
  </media>
  <link id="link64" xconnector="conn#onSelectionStart">
    <bind component="enviarsmsA" role="onSelection">
    </bind>
    <bind component="opcion1" role="start">
    </bind>
  </link>
  <media descriptor="desexito" id="exito_envio"
src="media/imagen/exito_envio.png">
  </media>
  <link id="link65" xconnector="conn#onEndStart">
    <bind component="opcion1" role="onEnd">
    </bind>
    <bind component="exito_envio" role="start">
    </bind>
  </link>
  <link id="link66" xconnector="conn#onSelectionStart">
    <bind component="enviarsmsB" role="onSelection">

```

```

        </bind>
        <bind component="opcion2" role="start">
        </bind>
    </link>
    <media descriptor="desexito" id="exito_envioB"
src="media/imagen/exito_envioB.png">
    </media>
    <link id="link67" xconnector="conn#onEndStart">
        <bind component="opcion2" role="onEnd">
        </bind>
        <bind component="exito_envioB" role="start">
        </bind>
    </link>
</body>
</ncl>

```

2) Opción.lua:

Archivo Lua encargado del envío de sms:

```

function send_vote (evt)
    if (evt.class == 'ncl') and (evt.type == 'presentation') and (evt.action == 'start')
then
    option = 1
    msg = string.format("#ginga#id:12345,option:%d",option)
    numero = 51962253918
    ret = event.post('out',{class='sms',to=numero,value=msg})
    if(ret==false) then

    else

    end
    event.post{
        class = 'ncl',
        type = 'presentation',
        action = 'stop'}
    end
end

event.register(send_vote)

```

ANEXO N°11 - Generación de Transport Stream

En el anexo se detalla el script correspondiente a cada tabla presente en el canal de televisión PUCPTV.

1. Tabla NIT:

```
#
# Network Information Table
# this is a basic NIT with the minimum descriptors, OpenCaster has a big library ready to use
#

nit = network_information_section(
    network_id = tvd_orig_network_id,
    network_descriptor_loop = [
        network_descriptor(network_name = "PUCP"),
        system_management_descriptor(
            broadcasting_flag = 0,
            broadcasting_identifier = 3,
            additional_broadcasting_identification = 0x01,
            additional_identification_bytes = [],
        )
    ],
    transport_stream_loop = [
        transport_stream_loop_item(
            transport_stream_id = tvd_ts_id,
            original_network_id = tvd_orig_network_id,
            transport_descriptor_loop = [
                service_list_descriptor(
                    dvb_service_descriptor_loop = [
                        service_descriptor_loop_item (
                            service_ID = tvd_service_id_hd,
                            service_type = 1,
                        ),
                        service_descriptor_loop_item (
                            service_ID = tvd_service_id_sd,
                            service_type = 2,
                        ),
                        service_descriptor_loop_item (
                            service_ID = tvd_service_id_ld,
                            service_type = 3,
                        ),
                    ],
                ),
                terrestrial_delivery_system_descriptor(
                    area_code = 1341,
                    guard_interval = 0x01,
                    transmission_mode = 0x02,
                    frequencies = [
                        tds_frequency_item( freq=ts_freq )
                    ],
                ),
            ],
        ),
    ],
)
```

```

partial_reception_descriptor (
    service_ids = []
),
transport_stream_information_descriptor (
    remote_control_key_id = ts_remote_control_key,
    ts_name = "PUCPTV",
    transmission_type_loop = [
        transmission_type_loop_item(
            transmission_type_info = 0x0F,#15
            service_id_loop = [
                service_id_loop_item(
                    service_id = tvd_service_id_sd
                ),
                service_id_loop_item(
                    service_id = tvd_service_id_hd
                ),
            ]
        ),
        transmission_type_loop_item(
            transmission_type_info = 0xAF,#175
            service_id_loop = [
                service_id_loop_item(
                    service_id = tvd_service_id_ld
                ),
            ],
        ),
    ],
),
],
),
],
version_number = 1,
section_number = 0,
last_section_number = 0,
)

```

2. Tabla PAT:

```

#
# Program Association Table (ISO/IEC 13818-1 2.4.4.3)
#

pat = program_association_section(
    transport_stream_id = tvd_ts_id,
    program_loop = [
        program_loop_item(
            program_number = tvd_service_id_hd,
            PID = tvd_pmt_pid_hd,
        ),
        program_loop_item(
            program_number = tvd_service_id_sd,
            PID = tvd_pmt_pid_sd,
        ),
    ],
)

```

```

    ),
    program_loop_item(
        program_number = tvd_service_id_ld,
        PID = tvd_pmt_pid_ld,
    ),
    program_loop_item(
        program_number = 0,
        PID = 16,
    ),
],
version_number = 1,
section_number = 0,
last_section_number = 0,
)

```

3. Tabla SDT

```

#
# Service Description Table (ETSI EN 300 468 5.2.3)
# this is a basic SDT with the minimum descriptors, OpenCaster has a big library ready to use
#

```

```

sdt = service_description_section(
    transport_stream_id = tvd_ts_id,
    original_network_id = tvd_orig_network_id,
    service_loop = [
        service_loop_item(
            service_ID = tvd_service_id_hd,
            EIT_schedule_flag = 0,
            EIT_present_following_flag = 0,
            running_status = 4,
            free_CA_mode = 0,
            service_descriptor_loop = [
                service_descriptor(
                    service_type = 1,
                    service_provider_name = "",
                    service_name = "PUCPHD",
                ),
            ],
        ),
    ],
),
service_loop_item(
    service_ID = tvd_service_id_sd,
    EIT_schedule_flag = 0,
    EIT_present_following_flag = 0,
    running_status = 4,
    free_CA_mode = 0,
    service_descriptor_loop = [
        service_descriptor(
            service_type = 2,
            service_provider_name = "",
            service_name = "PUCPSD",
        ),
    ],
),

```

```

    ],
  ),
  service_loop_item(
    service_ID = tvd_service_id_ld,
    EIT_schedule_flag = 0,
    EIT_present_following_flag = 0,
    running_status = 4,
    free_CA_mode = 0,
    service_descriptor_loop = [
      service_descriptor(
        service_type = 3,
        service_provider_name = "",
        service_name = "PUCP oneseg",
      ),
    ],
  ),
  ),
  ],
  version_number = 1,
  section_number = 0,
  last_section_number = 0,
)

```

4. Tabla PMT para el servicio PUCP HD

```

#
# Program Map Table (ISO/IEC 13818-1 2.4.4.8)
# this is a basic PMT the the minimum descriptors, OpenCaster has a big library ready to use
#

pmt_hd = program_map_section(
  program_number = tvd_service_id_hd,
  PCR_PID = 2064,
  program_info_descriptor_loop = [],
  stream_loop = [
    stream_loop_item(
      stream_type = 2, # mpeg2 video stream type
      elementary_PID = 2064,
      element_info_descriptor_loop = []
    ),
    stream_loop_item(
      stream_type = 3, # mpeg2 audio stream type
      elementary_PID = 2068,
      element_info_descriptor_loop = []
    ),
    stream_loop_item(
      stream_type = 5, # AIT stream type
      elementary_PID = 2001,
      element_info_descriptor_loop = [
        data_component_descriptor (
          data_component_id = 0xA3, # sistema AIT
          additional_data_component_info = ait_identifier_info(

```



```

        application_type =
GINGA_NCL_application_type,
        ait_version = 0
    ).bytes(),
    ),
    application_signalling_descriptor(
        application_type = 9, # 9 GINGA-NCL
        AIT_version = 1, # current ait version
    ),
    ]
),
stream_loop_item(
    stream_type = 0x0B, # DSMCC stream type
    elementary_PID = 2004,
    element_info_descriptor_loop = [
        association_tag_descriptor(
            association_tag = 0x0C,
            use = 0,
            selector_lenght = 0,
            transaction_id = 0x80000000,
            timeout = 0xFFFFFFFF,
            private_data = "",
        ),
        stream_identifier_descriptor(
            component_tag = 0x0C,
        ),
        carousel_identifier_descriptor(
            carousel_ID = 2,
            format_ID = 0,
            private_data = "",
        ),
        data_component_descriptor (
            data_component_id = 0xA0, # sistema GINGA
            additional_data_component_info =
additional_ginga_j_info(
                transmission_format = 0x2,
                document_resolution = 0x5,
                organization_id = 0x0000000A,
                application_id = 0x0064,
                carousel_id = 2,
            ).bytes(),
        ),
    ]
)
),
    ],
    version_number = 1,
    section_number = 0,
    last_section_number = 0,
)

```

5. Tabla PMT para el servicio PUCP SD

```

#
# Program Map Table (ISO/IEC 13818-1 2.4.4.8)
# this is a basic PMT the the minimum descriptors, OpenCaster has a big library ready to use
#
pmt_sd = program_map_section(
    program_number = tvd_service_id_sd,
    PCR_PID = 2064,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # mpeg2 video stream type
            elementary_PID = 2064,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 3, # mpeg2 audio stream type
            elementary_PID = 2068,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5, # AIT stream type
            elementary_PID = 2001,
            element_info_descriptor_loop = [
                data_component_descriptor (
                    data_component_id = 0xA3, # sistema AIT
                    additional_data_component_info = ait_identifier_info(
                        application_type =
GINGA_NCL_application_type,
                        ait_version = 0
                    ).bytes(),
                ),
                application_signalling_descriptor(
                    application_type = 9, # 9 GINGA-NCL
                    AIT_version = 1, # current ait version
                ),
            ]
        ),
        stream_loop_item(
            stream_type = 0x0B, # DSMCC stream type
            elementary_PID = 2004,
            element_info_descriptor_loop = [
                association_tag_descriptor(
                    association_tag = 0x0C,
                    use = 0,
                    selector_lenght = 0,
                    transaction_id = 0x80000000,
                    timeout = 0xFFFFFFFF,
                    private_data = "",
                ),
                stream_identifier_descriptor(

```

```

        component_tag = 0x0C,
    ),
    carousel_identifier_descriptor(
        carousel_ID = 2,
        format_ID = 0,
        private_data = "",
    ),
    data_component_descriptor (
        data_component_id = 0xA0, # sistema GINGA
        additional_data_component_info =
additional_ginga_j_info(
        transmission_format = 0x2,
        document_resolution = 0x5,
        organization_id = 0x0000000A,
        application_id = 0x0064,
        carousel_id = 2,
    ).bytes(),
    ),
    ],
    ),
    ],
    version_number = 1,
    section_number = 0,
    last_section_number = 0,
)

```

6. Tabla PMT para el servicio PUCP one-seg

```

#
# Program Map Table (ISO/IEC 13818-1 2.4.4.8)
# this is a basic PMT the the minimum descriptors, OpenCaster has a big library ready to use
#
pmt_id = program_map_section(
    program_number = tvd_service_id_ld,
    PCR_PID = 2064,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # mpeg2 video stream type
            elementary_PID = 2064,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 3, # mpeg2 audio stream type
            elementary_PID = 2068,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5, # AIT stream type
            elementary_PID = 2001,
            element_info_descriptor_loop = [
                data_component_descriptor (

```

```

data_component_id = 0xA3, # sistema AIT
additional_data_component_info = ait_identifier_info(
    application_type =
GINGA_NCL_application_type,
    ait_version = 0
).bytes(),
),
application_signalling_descriptor(
    application_type = 9, # 9 GINGA-NCL
    AIT_version = 1, # current ait version
),
]
),
stream_loop_item(
    stream_type = 0x0B, # DSMCC stream type
    elementary_PID = 2004,
    element_info_descriptor_loop = [
        association_tag_descriptor(
            association_tag = 0x0C,
            use = 0,
            selector_lenght = 0,
            transaction_id = 0x80000000,
            timeout = 0xFFFFFFFF,
            private_data = "",
        ),
        stream_identifier_descriptor(
            component_tag = 0x0C,
        ),
        carousel_identifier_descriptor(
            carousel_ID = 2,
            format_ID = 0,
            private_data = "",
        ),
        data_component_descriptor (
            data_component_id = 0xA0, # sistema GINGA
            additional_data_component_info =
additional_ginga_j_info(
                transmission_format = 0x2,
                document_resolution = 0x5,
                organization_id = 0x0000000A,
                application_id = 0x0064,
                carousel_id = 2,
            ).bytes(),
        ),
    ]
)
),
version_number = 1,
section_number = 0,
last_section_number = 0,
)

```

7. Tabla AIT

```

ait = application_information_section(
    application_type = 0x0009, # GINGA-NCL
    common_descriptor_loop = [],
    application_loop = [
        application_loop_item(
            organisation_id = 0x0000000A,
            application_id = 0x64,
            application_control_code = 0x01, # AUTOSTART

            application_descriptors_loop = [
                transport_protocol_descriptor(
                    protocol_id = 0x0001,
                    transport_protocol_label = 0,
                    remote_connection = 0,
                    component_tag = 0x0C, # association_tag
                ),
                application_descriptor(
                    application_profile = 0x0001,
                    version_major = 1,
                    version_minor = 0,
                    version_micro = 0,
                    service_bound_flag = 1,
                    visibility = 3,
                    application_priority = 1,
                    transport_protocol_labels = [ 0 ],
                ),
                application_name_descriptor(
                    application_name = "app_ginga"
                ),
                ginga_ncl_application_descriptor(
                    parameters = [ ]
                ),
                ginga_ncl_application_location_descriptor (
                    base_directory = "/",
                    class_path_extension = "",
                    initial_class = "inicio.ncl", # nombre del archivo NCL
                                                    # a ser ejecutado.
                ),
            ],
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)

```