UNIVERSIDAD ORT
Uruguay

OBLIGATORIO FINAL PROGRAMACIÓN II

Autores:

Ignacio Lanzani (143289)

Alejandro Rossi (195243)

TABLA DE CONTENIDO

INTRODUCCIÓN

Introducción:

Para este obligatorio creamos un software que gestiona elementos y dinámicas propias de los juegos para incorporar al aprendizaje en base a los temas que que sean subidos al al programa.
El sistem posee dos formas de jugar basadas en la gamificación: Flash y Memory

Decisiones de diseño:

Con base en lo trabajado en clase diseñamos el programa para que sea lo más sencillo e intuitivo posible, intentando que las opciones sean claras y a su vez dirijan al usuario hacia lo que este desee realizar.

En la ventana principal se irán habilitando las opciones a medida que el usuario cumpla con las condiciones establecidas por el sistema y necesarias para poder realizar las tareas. Luego de cargar temas en el programa, ya sea, manualmente, desde un archivo previamente guardado o por la opción de generarlos automáticamente. El usuario podrá utilizar las opciones que determine convenientes para jugar al modo “Flash” o modo “Memory”.

Estas opciones se encuentran distribuidas en los menús de la barra superior y las opciones que corresponden a la elección de temas para los juegos se encuentran disponibles en la ventana principal separando así, la organización del sistema de los juegos.

Modo Flash:

El modo fomenta la memoria, se seleccionan preguntas al azar de los temas elegidos en la ventana principal para crear tarjetas. Cliqueando en las tarjeta alternamos entre su pregunta y su respuesta, podemos avanzar hacia la siguiente pregunta y volver a la pregunta anterior. En cualquier momento tenemos la opción de volver hacia el menú principal.

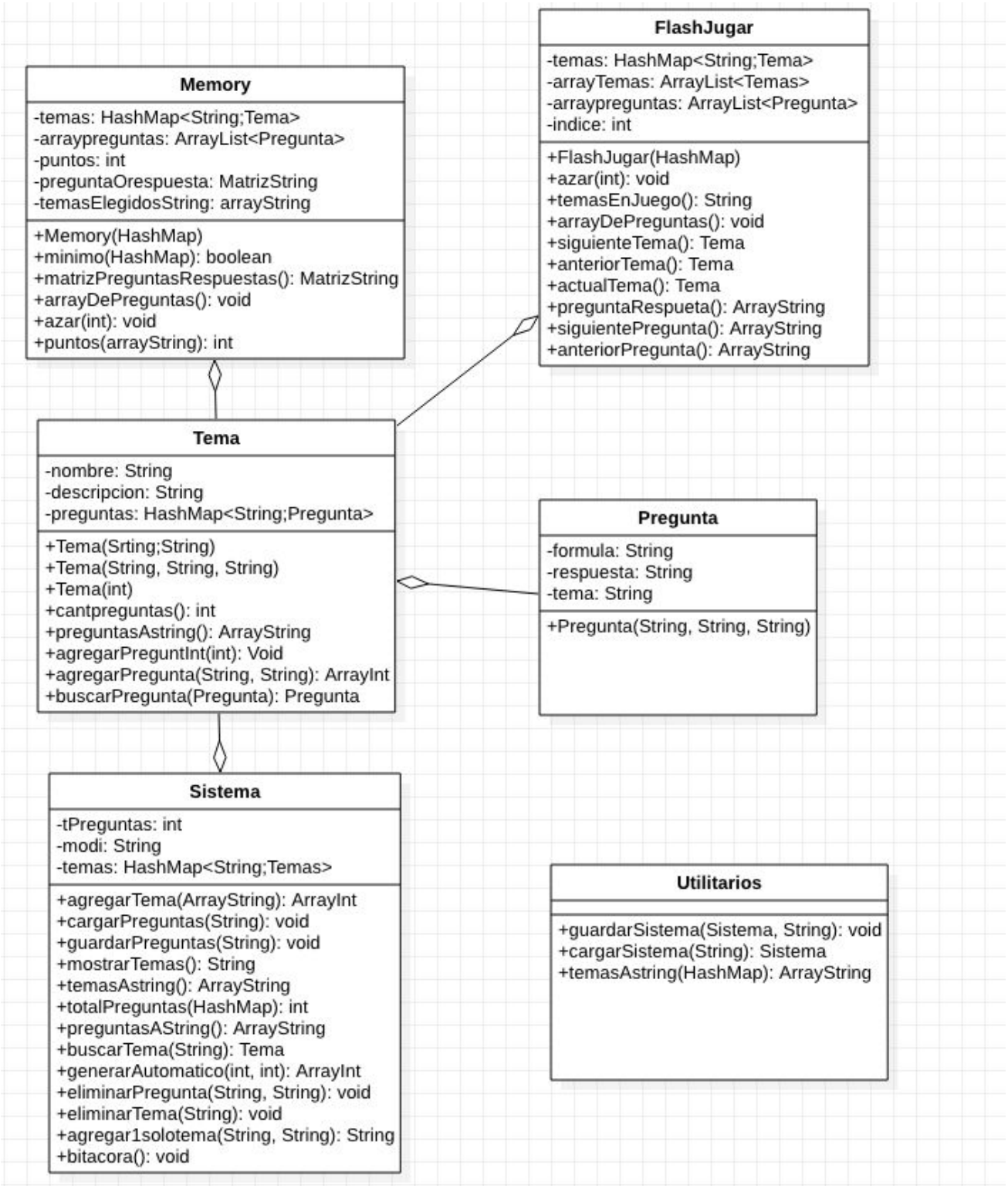
Modo Memory:

Si en el menú principal los temas seleccionados tienen al menos seis preguntas, se activará la opción de jugar Memory este juego contribuye a la atención así como a la memoria. El sistema selecciona al azar 6 preguntas y respuestas y las distribuye en el tablero generando doce tarjetas. El usuario debe unir las Preguntas con su respuesta para así sumar puntos. Dentro de este modo generamos botones de ayuda para el usuario.

Dentro del menú principal el sistema posee un sencillo pero poderoso sistema para gestionar los temas y las preguntas, el programa tiene una interfaz para agregar Temas o preguntas a los temas ya existentes, modificar la descripción de los temas creados o las respuestas de las preguntas ya cargadas en el sistema y por último, también eliminar preguntas y temas de forma sencilla y cómoda.

Todos los temas cargados en el sistema se guardarán automáticamente al cerrar el programa, teniendo la posibilidad de exportarlos del programa para abrirlos en otro ordenador.

UNIFIED MODELING LANGUAGE (U.M.L)



Link Video

[Video Muestra en YouTube]

Dominio: Clase FlashJugar

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

public class FlashJugar {

   private HashMap<String, Tema> temas = new HashMap<>();

   private ArrayList<Tema> arraytemas = new ArrayList<Tema>();

   private ArrayList<Pregunta> arraypreguntas = new ArrayList<Pregunta>();

   private int indice=0;


   public FlashJugar( HashMap<String, Tema> unTemas){

        this.temas=unTemas;

        this.azar(2);

        this.arrayDePreguntas();

   }

   public FlashJugar(){

   }

   //Set´s y Get´s

    public HashMap<String, Tema> getTemas() {

        return temas;

    }

    public void setTemas(HashMap<String, Tema> temas) {

        this.temas = temas;
```

```
    }


    public void setArraytemas(ArrayList<Tema> arraytemas) {
        this.arraytemas = arraytemas;
    }
    public ArrayList<Pregunta> getArraypreguntas() {
        return arraypreguntas;
    }
    public void setArraypreguntas(ArrayList<Pregunta> arraypreguntas) {
        this.arraypreguntas = arraypreguntas;
    }
    public int getIndice() {
        return indice;
    }
    public void setIndice(int indice) {
        this.indice = indice;
    }
    public ArrayList<Tema> getArraytemas() {
        return arraytemas;
    }
    public void setAux(ArrayList<Tema> aux) {
        this.setArraytemas(aux);
    }
     public void azar(int unaCantidad){//quita esta cantidad de pregutnas y selecciona al
azar
        int count=unaCantidad;

        ArrayList<Integer> arrayrandom = new ArrayList<>();//lleno el array con enteros
        for (int x=0;x<5;x++){
```

```
            arrayrandom.add(x);
            Collections.shuffle( arrayrandom);
        }
        ArrayList<String>aux = new ArrayList<String>();
        for (String i : this.getTemas().keySet()) {
            for (String b : this.getTemas().get(i).getPreguntas().keySet()) {
                                                    if   (arrayrandom.get(0)%2==0   &&
(this.getTemas().get(i).getPreguntas().size())-count>0 && count>0){
                    aux.add(this.getTemas().get(i).getPreguntas().get(b).getFormula());
                    Collections.shuffle( arrayrandom);
                }
             count--;
            }//endfor b
            for(int x=0;x<aux.size();x++){
                this.getTemas().get(i).getPreguntas().remove(aux.get(0));
            }
            count=0;
        }//endfor


    }
    //temas elegidos dentro del hashmap
    public String temasEnJuego(){
        String salida="";
        for (String i: this.getTemas().keySet()){
                                            salida+=this.getTemas().get(i).getNombre()+"
"+this.getTemas().get(i).getDescripcion()+"  ";


        }
    return salida;
```

```
    }
    //Recorre el hasmap para peder reccorerlo en forma numerica dentro del array list
    public  void arrayDePreguntas(){
        int y=0;//indice para el array
        this.getTemas();
            for (String i : this.getTemas().keySet()) {//reccorre los temas
                  for (String b : this.getTemas().get(i).getPreguntas().keySet()){// las
preguntas

this.getArraypreguntas().add(y,this.getTemas().get(i).getPreguntas().get(b));
                    System.out.println(this.getArraypreguntas().get(y).getFormula());
                    y++;
                }
            }
            Collections.shuffle(this.getArraypreguntas());
    }
    //metodo que permite que permite moverse asia la derecha
    public Tema siguienteTema(){
        if (getIndice()<this.getArraytemas().size()){
            this.setIndice(this.getIndice() + 1);
        }
        return this.getArraytemas().get(getIndice());
    }

    //metodo que permite que permite moverse asia la izquierada
    public Tema anteriorTema(){
        if (getIndice()>0){
            this.setIndice(this.getIndice() - 1);
        }
```

```
        return this.getArraytemas().get(getIndice());
    }

    //tema acual en el array
    public Tema actualTema(){

        return this.getArraytemas().get(getIndice());
    }

    //devuelve un string de pregunta y respuesta
    public String[] preguntaRespuesta(){
        String[] salida=new String[2];
        salida[0]=this.getArraypreguntas().get(this.getIndice()).getFormula();
        salida[1]=this.getArraypreguntas().get(this.getIndice()).getRespuesta();
        return salida;
    }//endmetodo

    //nos permite avanzar en el array de pregutnas
     //si llegamos al final para o cambia de tema si hay uno
    public String[] siguientePregunta(){
        if (this.getIndice()<this.getArraypreguntas().size()-1) {
            this.setIndice(this.getIndice()+1);
        }
        return this.preguntaRespuesta();
    }

    //nos permite ir Hacia atras en el array de preguntas
    //si llegamos al final para o cambia de tema si hay uno
```

```
    public String[] anteriorPregunta(){
        if (this.getIndice()>=1) {
            this.setIndice(this.getIndice()-1);
        }
        return this.preguntaRespuesta();
    }
}
```

Dominio: Clase Memory

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
public class Memory {
    private HashMap<String, Tema> temas = new HashMap<>();
    private ArrayList<Pregunta> arraypreguntas = new ArrayList<Pregunta>();
    private int puntos=0;
    private String[][] preguntaOresputa= new String[3][4]; //matriz
    private String temasElegidosString="";

    public Memory(HashMap<String, Tema> unMapa){
      this.temas=unMapa;
      //this.azar(2);
      if (this.minimo(unMapa)){ //pedis que halla un minimo
        this.arrayDePreguntas();
      }
    }
    public boolean minimo(HashMap<String, Tema> unMapa){
        int count=0;
```

```
        for (String i : unMapa.keySet()) {
            count+=unMapa.get(i).Cantpreguntas();
        }
        return (count>=6);
    }
    //Set´s y Get´s
    public HashMap<String, Tema> getTemas() {
        return temas;
    }
    public void setTemas(HashMap<String, Tema> temas) {
        this.temas = temas;
    }
    public ArrayList<Pregunta> getArraypreguntas() {
        return arraypreguntas;
    }
    public void setArraypreguntas(ArrayList<Pregunta> arraypreguntas) {
        this.arraypreguntas = arraypreguntas;
    }
    public int getPuntos() {
        return puntos;
    }
    public void setPuntos(int puntos) {
        this.puntos = puntos;
    }
    public String[][] getPreguntaOresputa() {
        return preguntaOresputa;
    }
    public void setPreguntaOresputa(String[][] preguntaOresputa) {
```

```
        this.setPreguntaOresputa(preguntaOresputa);
    }

    public String getTemasElegidosString() {
        return temasElegidosString;
    }
    public void setTemasElegidosString(String temasElegidosString) {
        this.temasElegidosString = temasElegidosString;
    }
      public String[][] matrizPregutnasRespuestas(){//este metodo garantiza preguntas y
respuestas en lugares siempre disintos
         String[][] matriz= new String[3][4];          //devuelve y guarda las respuestas
y formaulas en lugares aleatorios
        ArrayList<Integer> random = new ArrayList<>();//lleno el array con enteros
        for (int x=0;x<12;x++){
            random.add(x);
        }
        Collections.shuffle(random);             //ahora quedaron en deshorden
        for(int i=0;i<6;i++){                        //numeros al azar sin repeticion
                                                   //guardas la respusta y la pregutna al
mismo tienmpo en lugares distintos

matriz[random.get(i)/4][random.get(i)%4]=this.getArraypreguntas().get(i).getFormula();

matriz[(random.get(i+6))/4][(random.get(i+6))%4]=this.getArraypreguntas().get(i).getResp
uesta();
            this.getPreguntaOresputa()[random.get(i)/4][random.get(i)%4]="PREG";
            this.getPreguntaOresputa()[(random.get(i+6))/4][(random.get(i+6))%4]="RES";
        }
        return matriz;
    }
```

```
   public  void arrayDePreguntas(){
        int y=0;//indice para el array
        this.getTemas();
        for (String i : this.getTemas().keySet()) {//reccorre los temas

this.setTemasElegidosString(this.getTemasElegidosString()+this.getTemas().get(i).getNomb
re()+" "+this.getTemas().get(i).getDescripcion()+"; ");
                 for (String b : this.getTemas().get(i).getPreguntas().keySet()){// las
preguntas

this.getArraypreguntas().add(y,this.getTemas().get(i).getPreguntas().get(b));
                System.out.println(this.getArraypreguntas().get(y).getFormula());
                y++;
            }
        }
        Collections.shuffle(this.getArraypreguntas());
    }


    public void azar(int unaCantidad){//quita esta cantidad de pregutnas
        int count=unaCantidad;
        int random=4;
        ArrayList<String>aux = new ArrayList<String>();
        for (String i : this.getTemas().keySet()) {
            for (String b : this.getTemas().get(i).getPreguntas().keySet()) {
                                                                    if    (random%2==0    &&
(this.getTemas().get(i).getPreguntas().size())-count>0 && count>0){
                    aux.add(this.getTemas().get(i).getPreguntas().get(b).getFormula());
                }
                count--;
```

```
            }//for
            for(int x=0;x<aux.size();x++){
                this.getTemas().get(i).getPreguntas().remove(aux.get(0));
            }
            count=0;
        }//endfor
     }

    public int puntos(String unArray[]){
        int salida=-10;
        for (int i=0;i<this.getArraypreguntas().size();i++){
            if (this.getArraypreguntas().get(i).getFormula().compareTo(unArray[0])==0){

if(this.getArraypreguntas().get(i).getRespuesta().compareTo(unArray[1])==0){
                    salida=50;
                }
                                                                    }else      if
(this.getArraypreguntas().get(i).getFormula().compareTo(unArray[1])==0){

if(this.getArraypreguntas().get(i).getRespuesta().compareTo(unArray[0])==0){
                    salida=50;
                }
            }
        }

        salida=this.getPuntos()+salida;
        if (salida<0){
          this.setPuntos(0);
        }else
```

```
            {this.setPuntos(salida);}
        return  (this.getPuntos());
    }
}
```

Dominio: Clase Pregunta

```
import java.io.Serializable;
public class Pregunta implements Serializable{
    private String formula="";
    private String respuesta="";
    private String tema="";

    public Pregunta( String unTema, String unaFormula,String unaRespuesta){
        this.tema=unTema;
        this.formula=unaFormula;
        this.respuesta=unaRespuesta;
    }
    //Set´s y Get´s
    public String getFormula() {
        return formula;
    }
    public void setFormula(String unaFormula) {
        this.formula = unaFormula;
    }
    public String getRespuesta() {
```

```
        return respuesta;
    }



    public void setRespuesta(String unaRespuesta) {
        this.respuesta = unaRespuesta;
    }
    public String getTema() {
        return tema;
    }
    public void setTema(String unTema) {
        this.tema = unTema;
    }
}
```

Dominio: Clase Sistema

```
import claseformater.ArchivoGrabacion;
import java.io.Serializable;
import java.util.HashMap;
import claseformater.ArchivoLectura;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
public class Sistema implements Serializable{
    private int tPreguntas=0;
    private String modi="";
     private HashMap<String, Tema> temas = new HashMap<>(); //transient si no quiero
serializar
    //Set´s y Get´s
```

```
    public HashMap<String, Tema> getTemas() {
        return temas;
    }

    public void setTemas(HashMap<String, Tema> unaLista) {
        this.temas = unaLista;
    }
    public int getTpreguntas() {
        return tPreguntas;
    }
    public void setTpreguntas(int tpreguntas) {
        this.tPreguntas = tpreguntas;
    }
    public String getModi() {
        return modi;
    }
    public void setModi(String modi) {
        this.modi = modi;
    }
   public int[] agregarTema(String[] unArray){
                                    // chequea que las lineas no esten vacias
                                    //si no estan vacias procede a crear o modificar
                                        //arrar[0] es tema array 1 es pregunta y array 3
respuesta
        int[] aux= new int[3];

        if (unArray[0].length()!=0 && unArray[1].length()!=0 && unArray[2].length()!=0){
            Tema t1;
            //busco, no esta agrego
```

```
            //busco y esta llamo al que agrega
            if (this.getTemas().get(unArray[0])!=null){
            //encontre el tema, busco la pregutna
                  int[] aux2=this.getTemas().get(unArray[0]).agregarPregunta( unArray[1],
unArray[2]);
                aux[1]=aux2[1];
                aux[2]=aux2[2];
            } else {
                //no encontre el tema, lo agrego
                t1=new Tema(unArray[0],unArray[1],unArray[2]);
                this.getTemas().put(t1.getNombre(), t1);
                aux[1]=1;//nueva
           }
        }else {
            aux[0]=1;//descartada
        }
        return aux;
    }
   public void cargarPreguntas(String unString){
        String salida="";
        ArchivoLectura lec= new ArchivoLectura(unString);
        String[] s1= new String[3];
        int[] aux=new int[3];
        int[] aux2;
        int count=0;
        while (lec.hayMasLineas()){
            s1[count]=lec.linea();
            count++;
```

```
            if (count==3){
                count=0;
                aux2= this.agregarTema(s1);
                aux[0]+=aux2[0];
                aux[1]+=aux2[1];
                aux[2]+=aux2[2];
            }
            System.out.println(lec.linea());
        }
        if(count>0){
            aux[0]++;
        }
        this.setModi("Descartadas "+aux[0]+ " Nuevas "+aux[1]+" Modificadas "+aux[2]);
    }
   public void guardarPreguntas(String unString){
        ArchivoGrabacion sal= new ArchivoGrabacion (unString);
         for(String i:this.getTemas().keySet()){                              //busca los
TEmas
             for(String b:this.getTemas().get(i).getPreguntas().keySet()){//se para en el
tema, y buscas sus pregutnas
                sal.grabarLinea(this.getTemas().get(i).getNombre());

sal.grabarLinea(this.getTemas().get(i).getPreguntas().get(b).getFormula());

sal.grabarLinea(this.getTemas().get(i).getPreguntas().get(b).getRespuesta());
            }
       }
       sal.cerrar();
       }
```

```
    public String mostrarTemas(){//simplemente para mostrar en formato string todas las preguntas y respuestas delsistema

        String salida="";

        for (String i : this.getTemas().keySet()) {

                            System.out.println("Name:  "  +  i  +  "  Descripcion:  "  +  this.getTemas().get(i).getDescripcion());

                                    salida+=("Name:  "  +  i  +  "  Descripcion:  "  +  this.getTemas().get(i).getDescripcion())+("\n");

            for (String b : this.getTemas().get(i).getPreguntas().keySet()) {

                                System.out.println("esto:  "  +  b  +  "  Descripcion:  "  +  this.getTemas().get(i).getPreguntas().get(b).getRespuesta());

                                        salida+=("esto:  "  +  b  +  "  Descripcion:  "  +  this.getTemas().get(i).getPreguntas().get(b).getRespuesta())+("\n");

            }

        }

        return salida;

    }

    public String[] temasAstring(){

                                                return(this.getTemas().keySet().toArray(new String[this.getTemas().size()]));//combierte a arrray de string

    }

    public int totalPreguntas(HashMap<String, Tema> unHashMap){

        int aux=0;

        for (String i:unHashMap.keySet()){

           aux+=unHashMap.get(i).getPreguntas().size();

        }

        return aux;

    }

    public String[] preguntasAstring(){

       String[] salida=new String[this.totalPreguntas(this.getTemas())];

       int indice=0;
```

```
       for(String i: this.getTemas().keySet()){

               String[]  aux=  this.getTemas().get(i).getPreguntas().keySet().toArray(new
String[this.getTemas().get(i).getPreguntas().size()]);

          System.arraycopy(aux, 0, salida, indice, aux.length);

          indice+=this.getTemas().get(i).getPreguntas().size();

        }

        return salida;

    }

    public Tema buscarTema(String unTema){

        return this.getTemas().get(unTema);

    }

public int[] generarAutomatico(int unaCantidadTemas, int catidadPreguntas  ){


        int[] retorno=new int [3];

        int[] auxiliar=new int [3];

        for(int i=0;i<unaCantidadTemas;i++){

           String[] aux= new String[3];

           aux[0]="T:"+(i+1);

           for (int y=0;y< catidadPreguntas;y++){

               aux[1]="T:"+(i+1)+ " P:"+(y+1)+" Texto de P:"+(y+1);

               aux[2]="T:"+(i+1)+ " P:"+ (y+1) +" Respuesta de P:"+(y+1);

               auxiliar=this.agregarTema(aux);

               retorno[0]+=auxiliar[0];                                    //descartada

               retorno[1]+=auxiliar[1];                                    //nueva

               retorno[2]+=auxiliar[2];                                    //modificada

                          this.getTemas().get("T:"+(i+1)).setDescripcion("Descripcion  de
T:"+(i+1));

            }
```

```
            if (catidadPreguntas==0){
                this.agregar1solotema("T:"+(i+1), "Descripcion de T:"+(i+1));


            }
        }
            this.setModi("Descartadas "+retorno[0]+ " Nuevas "+retorno[1]+" Modificadas
"+retorno[2]);
        System.out.println(getModi());
     return retorno;
     }


   public void eliminarPregunta (String unTema, String unaPregunta){
        this.getTemas().get(unTema).getPreguntas().remove(unaPregunta);
     }
   public boolean elimarTema (String unTema){
        boolean salida=this.getTemas().get(unTema).Cantpreguntas()==0;
        if (this.getTemas().get(unTema).Cantpreguntas()==0){
            this.getTemas().remove(unTema);
        }
        return !salida;
     }
   public String agregar1solotema(String unNombre, String unaDescripcion){
        String salida="";
        if (this.getTemas().get(unNombre)!=null){
            salida="El tema ya existe";
        }else {
            Tema t= new Tema(unNombre, unaDescripcion);
            this.getTemas().put(t.getNombre(), t);
```

```
            salida="Tema creado correcamente";
        }
       return salida;
   }

   public void bitacora(){
        String aGrabar=getModi();
        ArchivoGrabacion grab= new ArchivoGrabacion("Bitacora.txt",true);
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
        LocalDateTime now = LocalDateTime.now();
        grab.grabarLinea(aGrabar+" "+ dtf.format(now));
        grab.cerrar();
    }
}
```

Dominio: Clase Tema

```
import java.io.Serializable;
import java.util.HashMap;
public class Tema implements Serializable{
    private String nombre="";
    private String descripcion="";
    private HashMap<String, Pregunta> preguntas = new HashMap<>();
    public Tema(String unNombre, String unaDescripcion){
        this.nombre=unNombre;
        this.descripcion=unaDescripcion;
    }
    //Get´s  Set´s
    public String getNombre() {
```

```
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public int Cantpreguntas() {
        return getPreguntas().size();
    }
    public String[] preguntasAstring(){
                                        return(this.getPreguntas().keySet().toArray(new
String[this.getPreguntas().size()]));
    }
    public void setPreguntas(HashMap<String, Pregunta> preguntas) {
        this.preguntas = preguntas;
    }
    public HashMap<String, Pregunta> getPreguntas() {
        return preguntas;
    }
    public Tema(int unNumero){
        this.nombre="T:"+unNumero;
        this.descripcion="Descripcion de T"+unNumero;
    }
```

```
    public void agregarPreguntInt(int unNumero){
          Pregunta p= new Pregunta (this.getNombre(),this.getNombre()+ " P:"+unNumero+" Texto de P "+unNumero," "+this.getNombre()+" P:"+unNumero+" Respuesta de P"+unNumero);
        this.preguntas.put(p.getFormula(), p);
    }
    public Tema(String unNombre, String unaPregunta, String unaRespuesta){
        this.nombre=unNombre;
        Pregunta p= new Pregunta (unNombre,unaPregunta,unaRespuesta);
        this.preguntas.put(p.getFormula(), p);
        this.descripcion="a completar";
    }
    //si la pregunta existe, modifica la respuesta
    //si no existe la crea
    public int[] agregarPregunta(String unaPregunta, String unaRespuesta){
        int[] aux=new int[3];
        //si la pregunta existe , se modifica
        Pregunta p= new Pregunta(this.getNombre(),unaPregunta,unaRespuesta);
        if (this.getPreguntas().get(unaPregunta)!=null){
            //encontre la pregunte
            this.getPreguntas().get(unaPregunta).setRespuesta(p.getRespuesta());
            aux[2]=1;//modificado
        }else{
            //agrega la nueva pregunta al tema
            this.getPreguntas().put(unaPregunta,p);
            aux[1]=1;
        }
        return aux;
```

```
    }
    public Pregunta buscarPregunta(Pregunta unaPregutna){
        return getPreguntas().get(unaPregutna.getFormula());
    }
}
```

Dominio: Clase Utilitarios

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.HashMap;
public abstract class Utilitarios {
       public static void guardarSistema(Sistema unContenido, String unaDir) throws IOException{
        FileOutputStream archivo =new FileOutputStream (unaDir);
        try (ObjectOutputStream datos = new  ObjectOutputStream(archivo)) {
             datos.writeObject(unContenido);
        } catch (IOException e){
            System.out.println("e");
        }
    }
          public static Sistema cargarSistema(String unaDir) throws IOException, ClassNotFoundException{
        FileInputStream archivo =new FileInputStream (unaDir);//cambiar a contenido
        Sistema s1 = null;
        try (ObjectInputStream datos = new  ObjectInputStream(archivo)) {
```

```
             s1 = (Sistema) datos.readObject();
        }  catch (IOException e){
            System.out.println("e");
        }catch(ClassNotFoundException a){
            System.out.println("a");
        }
        return s1;
    }
    public static String[] temasAstring( HashMap<String, Tema> unosTemas){
        return(unosTemas.keySet().toArray(new String[unosTemas.size()]));
   }
}
```