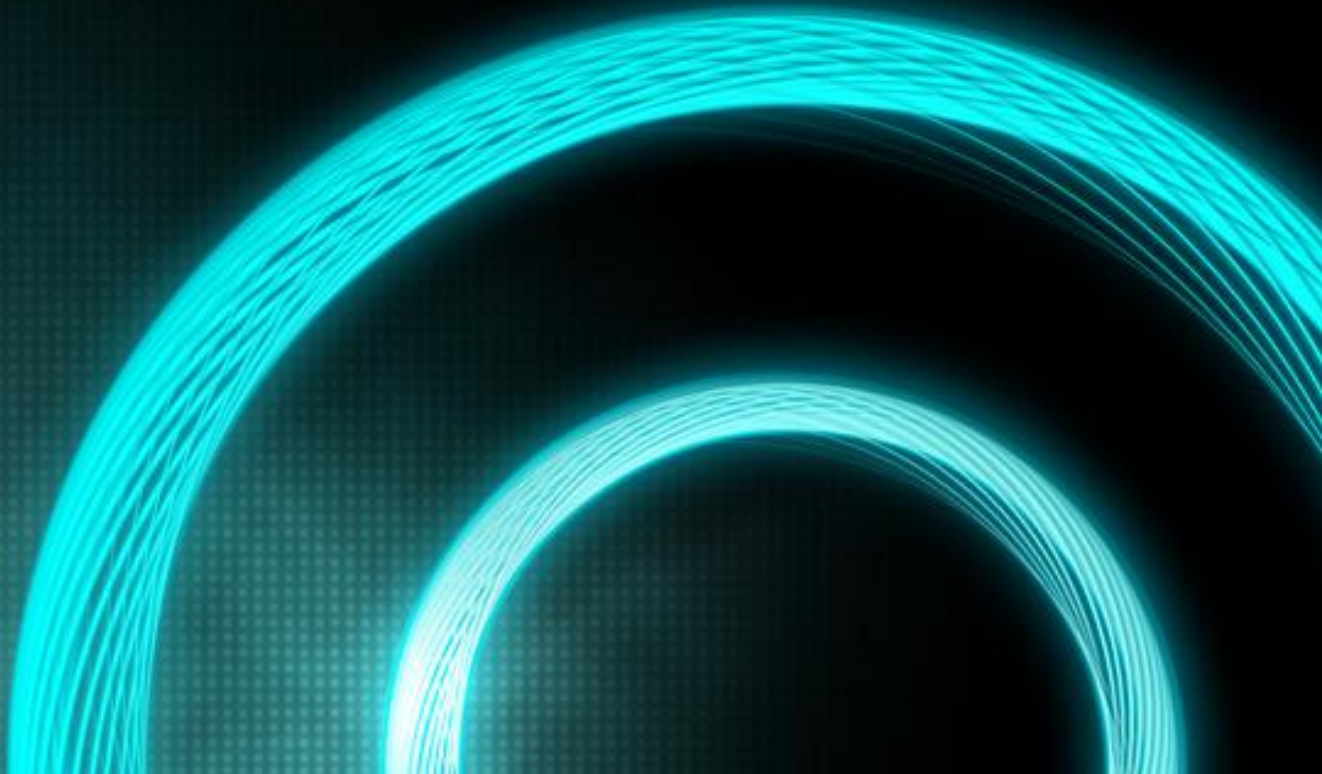


Manual

17-06-2019

Técnico

Programación Práctica 3º Trimestre





Manual Técnico

Carmen Sandiego

ÍNDICE

Introducción	3
Base de Datos	3
Descripción	3
Diagrama Entidad-Relación (ERD)	3
Esquema Relacional (ER)	4
Diagrama de Workbench.....	4
Creación de Base de Datos	5
Creación de Tablas	5
<u>JUGADOR</u>	5
<u>CASOS</u>	6
<u>CIUDADES</u>	6
<u>SOSPECHOSOS</u>	6
<u>VIAJES</u>	7
Librerías	7
Instalación	7
Código Java	8

Introducción

Este es el manual técnico del juego “Carmen Sandiego”. En este manual se describen los aspectos para la gestión de la base de datos y el código utilizado:

Más específicamente las siguientes características:

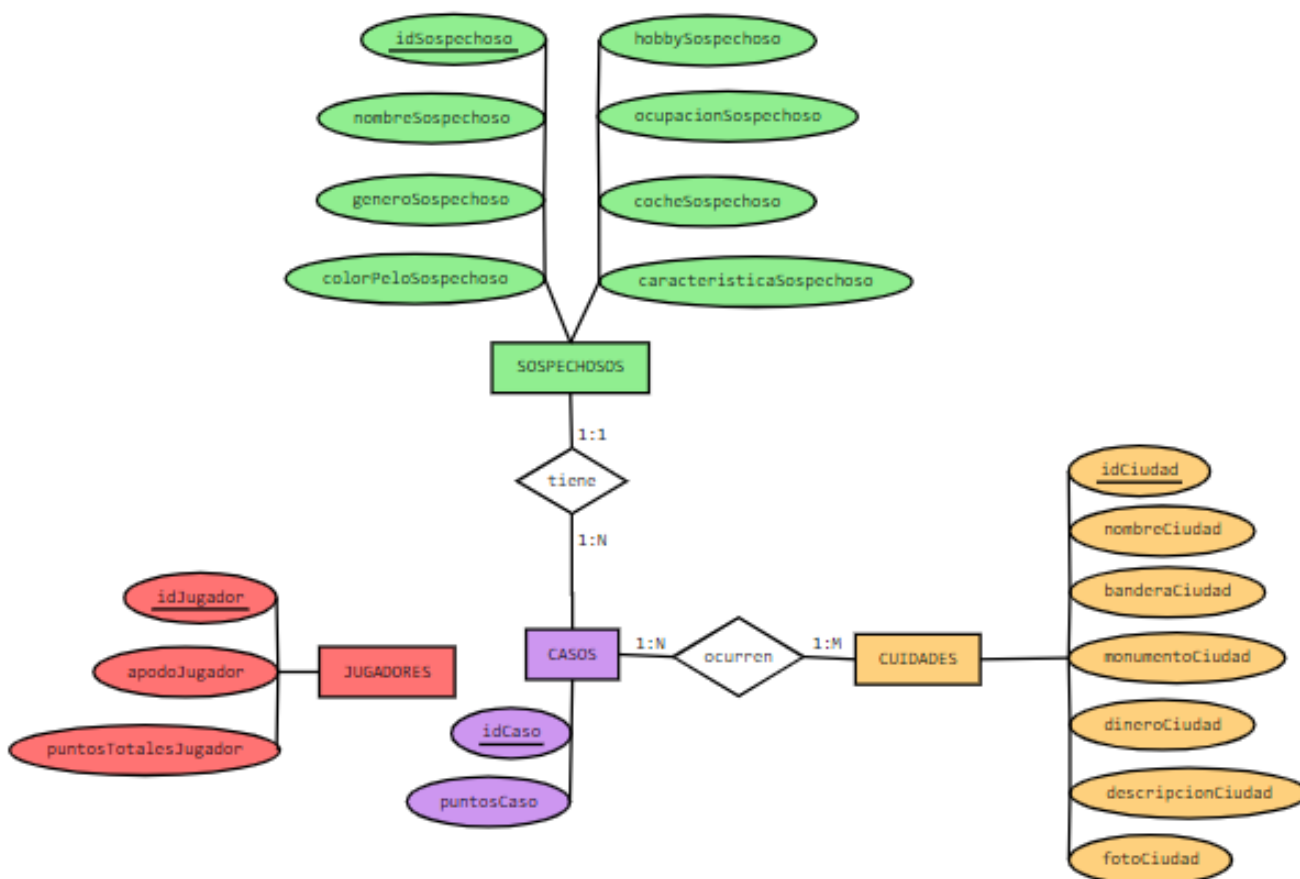
- Modelo de datos (diagrama E-R, esquema relacional y diagrama de Workbench).
- Sentencias SQL de creación de la base de datos y las tablas.
- Librerías externas utilizadas para el desarrollo de la aplicación.
- Las clases JAVA desarrolladas.

Base de Datos

Descripción

El juego de Carmen Sandiego trata sobre un agente de investigación al que se le asignan misiones de perseguir y arrestar a ladrones por todo el mundo. Empiezas en una ciudad y puedes ir a tres lugares diferentes de esa ciudad para averiguar a donde ha escapado mediante el testimonio de una persona. También hay tres opciones de viajes a otras ciudades de otros países. Debes decidir el lugar adecuado al que viajar o podrás perder la pista al ladrón. En los testimonios que vas logrando te dicen datos del ladrón (el color de pelo, su género, etc...), esto facilitará descubrir de la lista de sospechosos quien es el ladrón y así pedir una orden de captura para cuando logres alcanzarlo. Cuanto más rápido lo atrapes más puntos recibirás como investigador.

Diagrama Entidad-Relación (ERD)



Esquema Relacional (ER)

JUGADORES (**idJugador**, apodoJugador, puntosTotalesJugador)

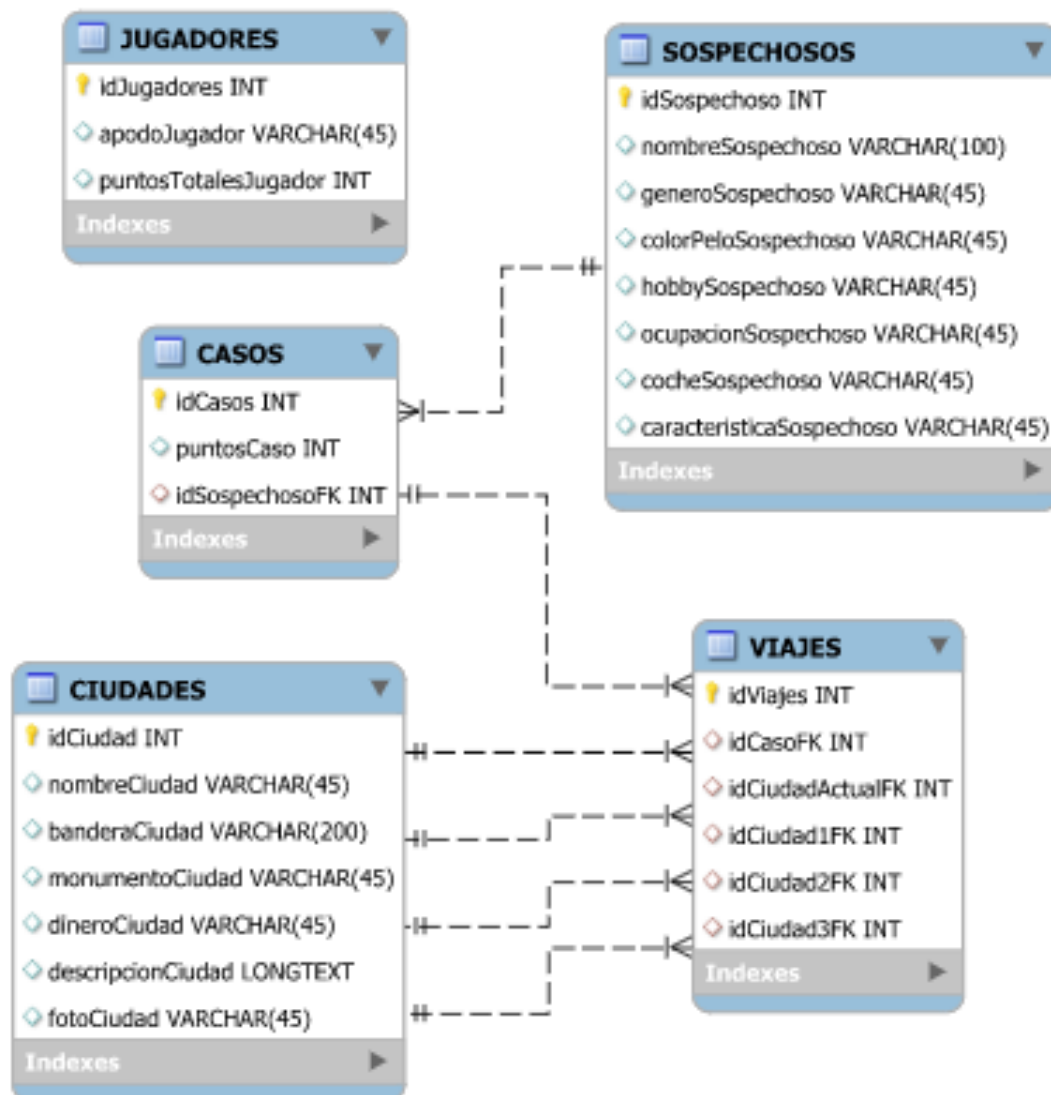
SOSPECHOSOS (**idSospechoso**, nombreSospechoso, generoSospechoso, colorPeloSospechoso, hobbySospechoso, ocupacionSospechoso, cocheSospechoso, caracteristicaSospechoso)

CASOS (**idCaso**, puntosCaso, **idSospechosoFK**)

CIUDADES (**idCiudad**, nombreCiudad, banderaCiudad, monumentoCiudad, dineroCiudad, descripcionCiudad, fotoCiudad)

VIAJES (**idViaje**, **idCasoFK**, **idCiudadActialFK**, **idCiudad1FK**, **idCiudad2FK**, **idCiudad3FK**)

Diagrama de Workbench



Creación de Base de Datos

```
MySQL 8.0 Command Line Client - Unicode
mysql> create database carmenSandiego charset utf8 collate utf8_spanish2_ci;
Query OK, 1 row affected, 2 warnings (0.36 sec)

mysql>
```

```
MySQL 8.0 Comm...
mysql> show databases;
+-----+
| Database |
+-----+
| biblioteca |
| carmensandiego |
| clubdetenis |
| ejercicio1 |
| ejercicio10 |
| ejercicio4 |
| ejercicio5 |
| ejercicio7 |
| ejercicio8 |
| empresa |
| gestion |
| information_schema |
| mysql |
| performance_schema |
| sys |
| tallerimportexport |
| videoclub |
+-----+
17 rows in set (0.04 sec)
```

Creación de Tablas

JUGADORES

```
MySQL 8.0 Command Line Client - Unicode
mysql> create table jugadores(idJugador int auto_increment, apodoJugador varchar(45), puntosTotalesJugador int, primary
key(idJugador));
Query OK, 0 rows affected (1.54 sec)

mysql> describe jugadores;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idJugador | int(11) | NO | PRI | NULL | auto_increment |
| apodoJugador | varchar(45) | YES | | NULL | |
| puntosTotalesJugador | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)

mysql>
```



SOSPECHOSOS

```
mysql> create table sospechosos(idSospechoso int auto_increment, nombreSospechoso varchar(100), generoSospechoso varchar(45), colorPeloSospechoso varchar(45), hobbySospechoso varchar(45), ocupacionSospechoso varchar(45), cocheSospechoso varchar(45), caracteristicaSospechoso varchar(45), primary key(idSospechoso));
Query OK, 0 rows affected (0.69 sec)

mysql> describe sospechosos;
```

Field	Type	Null	Key	Default	Extra
idSospechoso	int(11)	NO	PRI	NULL	auto_increment
nombreSospechoso	varchar(100)	YES		NULL	
generoSospechoso	varchar(45)	YES		NULL	
colorPeloSospechoso	varchar(45)	YES		NULL	
hobbySospechoso	varchar(45)	YES		NULL	
ocupacionSospechoso	varchar(45)	YES		NULL	
cocheSospechoso	varchar(45)	YES		NULL	
caracteristicaSospechoso	varchar(45)	YES		NULL	

```
8 rows in set (0.00 sec)

mysql>
```

CASOS

```
mysql> create table casos(idCaso int auto_increment, puntosCaso int, idSospechosoFK int, primary key(idCaso), foreign key(idSospechosoFK) references sospechosos(idSospechoso));
Query OK, 0 rows affected (1.02 sec)

mysql> describe casos;
```

Field	Type	Null	Key	Default	Extra
idCaso	int(11)	NO	PRI	NULL	auto_increment
puntosCaso	int(11)	YES		NULL	
idSospechosoFK	int(11)	YES	MUL	NULL	

```
3 rows in set (0.00 sec)

mysql>
```

CIUDADES

```
mysql> create table ciudades(idCiudad int auto_increment, nombreCiudad varchar(45), banderaCiudad varchar(200), monumentoCiudad varchar(45), dineroCiudad varchar(45), descripcionCiudad longtext, fotoCiudad varchar(45), primary key(idCiudad));
Query OK, 0 rows affected (1.07 sec)

mysql> describe ciudades;
```

Field	Type	Null	Key	Default	Extra
idCiudad	int(11)	NO	PRI	NULL	auto_increment
nombreCiudad	varchar(45)	YES		NULL	
banderaCiudad	varchar(200)	YES		NULL	
monumentoCiudad	varchar(45)	YES		NULL	
dineroCiudad	varchar(45)	YES		NULL	
descripcionCiudad	longtext	YES		NULL	
fotoCiudad	varchar(45)	YES		NULL	

```
7 rows in set (0.00 sec)

mysql>
```



VIAJES

```

mysql> create table viajes(idViaje int auto_increment, idCasoFK int, idCiudadActualFK int, idCiudad1FK int, idCiudad2FK
int, idCiudad3FK int, primary key(idViaje), foreign key(idCasoFK) references casos(idCaso), foreign key(idCiudadActualFK
) references ciudades(idCiudad), foreign key(idCiudad1FK) references ciudades(idCiudad), foreign key(idCiudad2FK) refere
nces ciudades(idCiudad), foreign key(idCiudad3FK) references ciudades(idCiudad));
Query OK, 0 rows affected (1.46 sec)

mysql> describe viajes;
+-----+-----+-----+-----+-----+-----+
| Field                | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| idViaje              | int(11)| NO   | PRI | NULL    | auto_increment |
| idCasoFK             | int(11)| YES  | MUL | NULL    |                |
| idCiudadActualFK    | int(11)| YES  | MUL | NULL    |                |
| idCiudad1FK         | int(11)| YES  | MUL | NULL    |                |
| idCiudad2FK         | int(11)| YES  | MUL | NULL    |                |
| idCiudad3FK         | int(11)| YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Librerías

Para el desarrollo de la aplicación se han utilizado las siguientes librerías externas:

- **MySQL Connector/J 5.1.35**
- **IText 5.5.5**

Instalación

Para poder usar este programa es necesario los siguientes componentes:

- MySQL Server (debidamente configurado).
- Eclipse IDE for Java Developers (Oxygen.2 Release (4.7.2)).
- Jdk 1.8.0_161
- Comprimido del programa.

Una vez se cumplan estos requisitos, el primer paso es descomprimir la carpeta del programa en el escritorio. Dentro de la carpeta “CarmenSandiego” encontraremos un archivo llamado “Instalar” como administrador. Lo ejecutaremos para que así se instale la base de datos necesaria, entre otras cosas.

Antes de usarlo se debe crear la conexión ‘remoto’ con la contraseña ‘Studium2018;’.

Podremos encontrar la carpeta con el programa y todos sus componentes en “C:\Program Files”.

Una vez que se cumplan estos requisitos, ya que está en una carpeta que necesita permisos de administrador, se puede abrir el programa de 2 maneras:

- Abrir un terminal como administrador y, estando en la ruta pertinente, escribir la siguiente sentencia: **java -jar CarmenSandiego.jar**

- Ir a la carpeta donde se encuentra el programa, buscar el archivo “Iniciar” y, pulsando sobre el con el botón derecho del ratón, seleccionar abrir como administrador.



Código Java

Datos.java

```
package CarmenSandiego;

import java.awt.Color;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class Datos extends JDialog implements ActionListener, MouseListener {
    private static final long serialVersionUID = 1L;
    String arrayGenero []= {"VACIO","femenino","masculino"};
    static int indiceGenero=0;
    String arrayPelo []= {"VACIO","castaño","rubio","rojizo","negro"};
    static int indicePelo=0;
    String arrayHobby []= {"VACIO","tenis","alpinismo","croquet"};
    static int indiceHobby=0;
    String arrayCoche []= {"VACIO","un descapotable","una limusina","un
deportivo","una moto"};
    static int indiceCoche=0;
    String arrayCaracteristica []= {"VACIO","una joya muy cara","un
tatuaje","un anillo muy caro"};
    static int indiceCaracteristica=0;
    JFrame frameJuego = new JFrame();
    static int sospechosoJugador;
    ResultSet respuesta;
    //Paneles y Diálogos
    JPanel pnl1 = new JPanel();
    //Etiquetas
    JLabel lblFondo = new JLabel();
    JLabel lblGenero = new JLabel("<html><h2>Genero: </h2></html>");
    JLabel lblPelo = new JLabel("<html><h2>Pelo: </h2></html>");
    JLabel lblHobby = new JLabel("<html><h2>Hobby: </h2></html>");
    JLabel lblCoche = new JLabel("<html><h2>Coche: </h2></html>");
```




```

JLabel lblSeña = new JLabel("<html><h2>Seña: </h2></html>");
JLabel lblGeneroResp = new
JLabel("<html><h2>"+arrayGenero[indiceGenero]+"</h2></html>");
JLabel lblPeloResp = new
JLabel("<html><h2>"+arrayPelo[indicePelo]+"</h2></html>");
JLabel lblHobbyResp = new
JLabel("<html><h2>"+arrayHobby[indiceHobby]+"</h2></html>");
JLabel lblCocheResp = new
JLabel("<html><h2>"+arrayCoche[indiceCoche]+"</h2></html>");
JLabel lblSeñaResp = new
JLabel("<html><h2>"+arrayCaracteristica[indiceCaracteristica]+"</h2></html>");
//Icono Investigar
ImageIcon imagenFondo= new ImageIcon("./img/ordenador.png");
//Boton
JButton btnAtras = new JButton("Atrás");
JButton btnOrden = new JButton("Pedir orden de arresto");

```

```

Datos(JFrame jf){
    frameJuego=jf;
    frameJuego.setEnabled(false);
    setLayout (new GridLayout(1,1));
    setSize(600, 400);
    setLocationRelativeTo(null);
    setResizable(false);
    setUndecorated(true);
    pnl1.setLayout(null);
    pnl1.setBorder(BorderFactory.createLineBorder(Color.white,4));
    pnl1.setBackground(Color.black);
    pnl1.add(lblGenero);
    lblGenero.setForeground(Color.white);
    lblGenero.setBounds(180, 60, 100, 30);
    pnl1.add(lblGeneroResp);
    lblGeneroResp.setBounds(260, 60, 100, 30);
    lblGeneroResp.setForeground(Color.white);
    pnl1.add(lblPelo);
    lblPelo.setForeground(Color.white);
    lblPelo.setBounds(180, 100, 100, 30);
    pnl1.add(lblPeloResp);
    lblPeloResp.setForeground(Color.white);
    lblPeloResp.setBounds(260, 100, 100, 30);
    pnl1.add(lblHobby);
    lblHobby.setForeground(Color.white);
    lblHobby.setBounds(180, 140, 100, 30);
    pnl1.add(lblHobbyResp);
    lblHobbyResp.setForeground(Color.white);
    lblHobbyResp.setBounds(260, 140, 100, 30);
    pnl1.add(lblCoche);
    lblCoche.setForeground(Color.white);
    lblCoche.setBounds(180, 180, 100, 30);
}

```



```

    pnl1.add(lblCocheResp);
    lblCocheResp.setForeground(Color.white);
    lblCocheResp.setBounds(260, 180, 200, 30);
    pnl1.add(lblSeña);
    lblSeña.setForeground(Color.white);
    lblSeña.setBounds(180, 220, 100, 30);
    pnl1.add(lblSeñaResp);
    lblSeñaResp.setForeground(Color.white);
    lblSeñaResp.setBounds(260, 220, 200, 30);
    pnl1.add(btnOrden);
    btnOrden.setBounds(150, 260, 200, 30);
    pnl1.add(btnAtras);
    btnAtras.setBounds(370, 260, 100, 30);
    lblFondo.setBounds(8, 20, 600, 350);
    lblFondo.setIcon(imagenFondo);
    pnl1.add(lblFondo);
    add(pnl1);
    //Listeners
    btnAtras.addActionListener(this);
    btnOrden.addActionListener(this);
    lblGeneroResp.addMouseListener(this);
    lblPeloResp.addMouseListener(this);
    lblHobbyResp.addMouseListener(this);
    lblCocheResp.addMouseListener(this);
    lblSeñaResp.addMouseListener(this);
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    if(btnAtras.equals(e.getSource())) {
        frameJuego.setEnabled(true);
        dispose();
    }
    if(btnOrden.equals(e.getSource())) {
        try {
            String arrayValores [] = {arrayGenero[indiceGenero],
arrayPelo[indicePelo], arrayHobby[indiceHobby], arrayCoche[indiceCoche],
arrayCaracteristica[indiceCaracteristica]};
            String arrayDatos [] = {"generoSospechoso",
"colorPeloSospechoso", "hobbySospechoso", "cocheSospechoso",
"caracteristicaSospechoso"};

            String sentencia = "SELECT idSospechoso, nombreSospechoso
FROM sospechosos WHERE ";
            boolean metido = false;
            for (int i=0;i<arrayDatos.length;i++) {

                if(!arrayValores[i].equals("VACIO")) {

```



```

        if (!metido) {
            sentencia += arrayDatos[i]+ "="
+arrayValores[i]+"';
            metido=true;
        }
        else {
            sentencia += " AND " + arrayDatos[i]+ "="
+arrayValores[i]+"';
        }
    }
    sentencia += " ";
    ModeloCSD.conectaBase();
    if(metido==true) {
        respuesta=ModeloCSD.buscarDatos(sentencia);
        respuesta.last();
        sospechosoJugador=respuesta.getRow();
        ModeloCSD.reduccionTiempo(3);
    }
    if(sospechosoJugador!=1) {
        JOptionPane.showMessageDialog(this,"<html><p>No se ha
encontrado una coincidencia unica o fiable.</p><CENTER><p>No puede emitirse la
orden de arresto.</p></CENTER></html>", "ORDEN DE
ARRESTO",JOptionPane.ERROR_MESSAGE);
        sospechosoJugador=0;
    }else {
        JOptionPane.showMessageDialog(this,"<html><p>Se ha
emitido una orden de arresto para "+respuesta.getString(2)+"</p></html>", "ORDEN
DE ARRESTO",JOptionPane.INFORMATION_MESSAGE);
        sospechosoJugador=respuesta.getInt(1);
        Juego.ordenCaptura=true;
    }
    ModeloCSD.desconectaBase();
    frameJuego.setEnabled(true);
    dispose();
} catch (SQLException e1) {
    ModeloCSD.finalJuego(frameJuego, this);
}
}

@Override
public void mouseClicked(MouseEvent e) {
    if(e.getComponent().equals(lblGeneroResp)) {
        if(indiceGenero!=arrayGenero.length-1) {
            indiceGenero++;
        }else {
            indiceGenero=0;
        }
    }
}

```



```

lblGeneroResp.setText("<html><h2>" + arrayGenero[indiceGenero] + "</h2></html>"
);
    }
    if(e.getComponent().equals(lblPeloResp)) {
        if(indicePelo != arrayPelo.length - 1) {
            indicePelo++;
        } else {
            indicePelo = 0;
        }

lblPeloResp.setText("<html><h2>" + arrayPelo[indicePelo] + "</h2></html>");
    }
    if(e.getComponent().equals(lblHobbyResp)) {
        if(indiceHobby != arrayHobby.length - 1) {
            indiceHobby++;
        } else {
            indiceHobby = 0;
        }

lblHobbyResp.setText("<html><h2>" + arrayHobby[indiceHobby] + "</h2></html>");
    }
    if(e.getComponent().equals(lblCocheResp)) {
        if(indiceCoche != arrayCoche.length - 1) {
            indiceCoche++;
        } else {
            indiceCoche = 0;
        }

lblCocheResp.setText("<html><h2>" + arrayCoche[indiceCoche] + "</h2></html>");
    }
    if(e.getComponent().equals(lblSeñaResp)) {
        if(indiceCaracteristica != arrayCaracteristica.length - 1) {
            indiceCaracteristica++;
        } else {
            indiceCaracteristica = 0;
        }

lblSeñaResp.setText("<html><h2>" + arrayCaracteristica[indiceCaracteristica] +
"</h2></html>");
    }
}

@Override
public void mouseEntered(MouseEvent e) {
    if(e.getComponent().equals(lblGeneroResp)) {
        lblGeneroResp.setForeground(Color.red);
    }
    if(e.getComponent().equals(lblPeloResp)) {
        lblPeloResp.setForeground(Color.red);
    }
}

```



```
    }
    if(e.getComponent().equals(lblHobbyResp)) {
        lblHobbyResp.setForeground(Color.red);
    }
    if(e.getComponent().equals(lblCocheResp)) {
        lblCocheResp.setForeground(Color.red);
    }
    if(e.getComponent().equals(lblSeñaResp)) {
        lblSeñaResp.setForeground(Color.red);
    }
}

@Override
public void mouseExited(MouseEvent e) {
    if(e.getComponent().equals(lblGeneroResp)) {
        lblGeneroResp.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblPeloResp)) {
        lblPeloResp.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblHobbyResp)) {
        lblHobbyResp.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblCocheResp)) {
        lblCocheResp.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblSeñaResp)) {
        lblSeñaResp.setForeground(Color.white);
    }
}

@Override
public void mousePressed(MouseEvent e) {
}

@Override
public void mouseReleased(MouseEvent e) {
}
}
```



Iniciacion.java

```

package CarmenSandiego;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Iniciacion {

    static String sentenciaLadron = "SELECT * FROM sospechosos ORDER BY rand()
LIMIT 1;";
    static String sentenciaCiudades = "SELECT * FROM ciudades ORDER BY
rand());";
    static ResultSet resultado;
    static String nombreLadron, generoLadron, peloLadron, hobbyLadron,
cocheLadron, caracteristicaLadron;
    static int idLadron;
    static String idPartida;
    static Ciudad ciudadesCaso[] = new Ciudad [15];
    static Ciudad ciudadesRnd[] = new Ciudad [2];

    public static void IniciacionJuego() {
        try {
            //Ladron
            resultado=ModeloCSD.buscarDatos(sentenciaLadron);
            resultado.next();
            idLadron = resultado.getInt(1);
            nombreLadron=resultado.getString(2);
            generoLadron=resultado.getString(3);
            peloLadron=resultado.getString(4);
            hobbyLadron=resultado.getString(5);
            cocheLadron=resultado.getString(7);
            caracteristicaLadron=resultado.getString(8);
            ModeloCSD.datosABM("INSERT INTO casos (puntosCaso,
idSospechosoFK) VALUES (0,'"+resultado.getString(1)+"')");
            resultado=ModeloCSD.buscarDatos("SELECT last_insert_id() FROM
casos LIMIT 1;");
            resultado.next();
            idPartida=resultado.getString(1);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void SelecCiudadesCorrecto() {
        try {
            resultado=ModeloCSD.buscarDatos(sentenciaCiudades);
            int i=0;
            while(resultado.next()) {

```




```

        String id=resultado.getString(1);
        String nombre=resultado.getString(2);
        String bandera= resultado.getString(3);
        String monumento = resultado.getString(4);
        String dinero= resultado.getString(5);
        String descripcion = resultado.getString(6);
        String foto = resultado.getString(7);

        ciudadesCaso[i]=new
Ciudad(id,nombre,bandera,monumento,dinero,descripcion,foto);
        i++;
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

public static void SelecCiudadesExtras() {
    try {
        String sentenciaCiudadesRnd="";
        if(Juego.ciudadActual!=0) {
            sentenciaCiudadesRnd = "SELECT * FROM ciudades WHERE NOT(
nombreCiudad='"+ciudadesCaso[Juego.ciudadActual-1].getNombreCiudad()+"' OR
nombreCiudad='"+ciudadesCaso[Juego.ciudadActual].getNombreCiudad()+"' OR
nombreCiudad='"+ciudadesCaso[Juego.ciudadActual+1].getNombreCiudad()+"') ORDER
BY rand() LIMIT 2;";
        }else {
            sentenciaCiudadesRnd = "SELECT * FROM ciudades WHERE NOT(
nombreCiudad='"+ciudadesCaso[Juego.ciudadActual].getNombreCiudad()+"' OR
nombreCiudad='"+ciudadesCaso[Juego.ciudadActual+1].getNombreCiudad()+"') ORDER
BY rand() LIMIT 2;";
        }
        resultado=ModeloCSD.buscarDatos(sentenciaCiudadesRnd);
        int i=0;
        while(resultado.next()) {
            String id=resultado.getString(1);
            String nombre=resultado.getString(2);
            String bandera= resultado.getString(3);
            String monumento = resultado.getString(4);
            String dinero= resultado.getString(5);
            String descripcion = resultado.getString(6);
            String foto = resultado.getString(7);

            ciudadesRnd[i]=new
Ciudad(id,nombre,bandera,monumento,dinero,descripcion,foto);
            i++;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block

```



```
        e.printStackTrace();
    }
}

class Ciudad{
    private String idCiudad, nombreCiudad, banderaCiudad, monumentoCiudad,
    dineroCiudad, descripcionCiudad, fotoCiudad;

    Ciudad(){
        idCiudad="";
        nombreCiudad="";
        banderaCiudad="";
        monumentoCiudad="";
        dineroCiudad="";
        descripcionCiudad="";
        fotoCiudad="";
    }
    Ciudad(String idCiudad, String nombreCiudad, String banderaCiudad, String
    monumentoCiudad, String dineroCiudad, String descripcionCiudad, String
    fotoCiudad){
        this.idCiudad=idCiudad;
        this.nombreCiudad=nombreCiudad;
        this.banderaCiudad=banderaCiudad;
        this.monumentoCiudad=monumentoCiudad;
        this.dineroCiudad=dineroCiudad;
        this.descripcionCiudad=descripcionCiudad;
        this.fotoCiudad=fotoCiudad;
    }
    public String getIdCiudad() {
        return idCiudad;
    }
    public void setIdCiudad(String idCiudad) {
        this.idCiudad=idCiudad;
    }
    public String getNombreCiudad() {
        return nombreCiudad;
    }
    public void setNombreCiudad(String nombreCiudad) {
        this.nombreCiudad = nombreCiudad;
    }
    public String getBanderaCiudad() {
        return banderaCiudad;
    }
    public void setBanderaCiudad(String banderaCiudad) {
        this.banderaCiudad = banderaCiudad;
    }
    public String getMonumentoCiudad() {
        return monumentoCiudad;
    }
}
```



```
public void setMonumentoCiudad(String monumentoCiudad) {
    this.monumentoCiudad = monumentoCiudad;
}
public String getDineroCiudad() {
    return dineroCiudad;
}
public void setDineroCiudad(String dineroCiudad) {
    this.dineroCiudad = dineroCiudad;
}
public String getDescripcionCiudad() {
    return descripcionCiudad;
}
public void setDescripcionCiudad(String descripcionCiudad) {
    this.descripcionCiudad = descripcionCiudad;
}
public String getFotoCiudad() {
    return fotoCiudad;
}
public void setFotoCiudad(String fotoCiudad) {
    this.fotoCiudad = fotoCiudad;
}
}
```



Investigar.java

```
package CarmenSandiego;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.util.Random;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class Investigar extends JDialog implements MouseListener, ActionListener
{

    private static final long serialVersionUID = 1L;
    JFrame frameJuego = new JFrame();
    Random rnd=new Random();
    int tipoPista;
    //Paneles y Diálogos
    JPanel pnl1 = new JPanel();
    //Etiquetas
    JLabel lblMuseo = new JLabel("Museo");
    JLabel lblImgMuseo = new JLabel();
    JLabel lblBanco= new JLabel("Banco");
    JLabel lblImgBanco= new JLabel();
    JLabel lblAerop = new JLabel("Aeropuerto");
    JLabel lblImgAerop = new JLabel();
    //Boton
    JButton btnAtras = new JButton("Atrás");
    //Fuente
    Font fuente = new Font("Calibri", 3, 20);
    /*Añadir imágenes*/
    //Icono Investigar
    ImageIcon iconoBanco= new ImageIcon("./img/banco.png");
    //Icono Investigar
    ImageIcon iconoAeropuerto= new ImageIcon("./img/aeropuerto.png");
    //Icono Investigar
    ImageIcon iconoMuseo= new ImageIcon("./img/museo.png");

    Investigar(JFrame m){
```



```

frameJuego=m;
frameJuego.setEnabled(false);
setLayout (new GridLayout(1,1));
setSize(400, 250);
setLocationRelativeTo(null);
setResizable(false);
setUndecorated(true);
pn1.setLayout(null);
pn1.setBorder(BorderFactory.createLineBorder(Color.white,4));
pn1.setBackground(Color.black);
pn1.add(lblImgBanco);
lblImgBanco.setIcon(iconoBanco);
lblImgBanco.setBounds(10, 35, 100, 100);
pn1.add(lblBanco);
lblBanco.setFont(fuente);
lblBanco.setForeground(Color.white);
lblBanco.setBackground(Color.black);
lblBanco.setOpaque(true);
lblBanco.setBounds(40, 150, 55, 25);
pn1.add(lblImgAerop);
lblImgAerop.setIcon(iconoAeropuerto);
lblImgAerop.setBounds(150, 35, 100, 100);
pn1.add(lblAerop);
lblAerop.setFont(fuente);
lblAerop.setForeground(Color.white);
lblAerop.setBackground(Color.black);
lblAerop.setOpaque(true);
lblAerop.setBounds(150, 150, 100, 25);
pn1.add(lblImgMuseo);
lblImgMuseo.setIcon(iconoMuseo);
lblImgMuseo.setBounds(280, 35, 100, 100);
pn1.add(lblMuseo);
lblMuseo.setFont(fuente);
lblMuseo.setForeground(Color.white);
lblMuseo.setBackground(Color.black);
lblMuseo.setOpaque(true);
lblMuseo.setBounds(300, 150, 60, 25);
pn1.add(btnAtras);
btnAtras.setBounds(150, 190, 100, 30);
add(pn1);
try {
    if(Juego.opcionInvestigar==-1 &&
Juego.viajeCorrecto.equals("Correcto")) {
        Juego.opcionInvestigar=rnd.nextInt(3);
        switch (Juego.opcionInvestigar) {
            case 0:
                Juego.atributoBanco=ModeloCSD.atributoLadronRnd();
                Juego.atributoAeropuerto=
Iniciacion.ciudadesCaso[Juego.ciudadActual+1].getBanderaCiudad();

```



```

        Juego. atributoMuseo=
Iniciacion. ciudadesCaso[Juego. ciudadActual+1].getMonumentoCiudad();
        break;
    case 1:

        Juego. atributoBanco=Iniciacion. ciudadesCaso[Juego. ciudadActual+1].getDinero
Ciudad();

        Juego. atributoAeropuerto=
ModeloCSD. atributoLadronRnd();
        Juego. atributoMuseo=
Iniciacion. ciudadesCaso[Juego. ciudadActual+1].getMonumentoCiudad();
        break;
    case 2:

        Juego. atributoBanco=Iniciacion. ciudadesCaso[Juego. ciudadActual+1].getDinero
Ciudad();

        Juego. atributoAeropuerto=
Iniciacion. ciudadesCaso[Juego. ciudadActual+1].getBanderaCiudad();
        Juego. atributoMuseo= ModeloCSD. atributoLadronRnd();
        break;
    }
}
} catch (Exception e) {
    ModeloCSD. finalJuego(frameJuego, this);
}
//Listeners
btnAtras.addActionListener(this);
lblMuseo.addMouseListener(this);
lblBanco.addMouseListener(this);
lblAerop.addMouseListener(this);
setVisible(true);
}

@Override
public void mouseClicked(MouseEvent e) {
    if(e.getSource().equals(lblAerop)) {
        try {
            if(Juego. pistaActual!=1 &&
Juego. viajeCorrecto.equals("Correcto")) {
                tipoPista=rnd.nextInt(2);
                Juego. lblImgText.setText("");
                if(Juego. opcionInvestigar!=1) {
                    Juego. lblPista.setText("<html><h4
align='justify'>Lo vi corriendo por el aeropuerto y montarse en un avión que
tenía una bandera "+Juego. atributoAeropuerto+"</h4></html>");
                }else {
                    Juego. lblPista.setText("<html><h4
align='justify'>No pude fijarme bien pero creo que el ladrón
"+Juego. atributoAeropuerto+"</h4></html>");
                }
            }
        }
    }
}

```




```

        if(tipoPista==0) {
            Juego.LblImgText.setIcon(new
ImageIcon("./img/pistas/pistaMAeropuerto.jpg"));
        }else if(tipoPista==1) {
            Juego.LblImgText.setIcon(new
ImageIcon("./img/pistas/pistaHAeropuerto.jpg"));
        }
        Juego.pistaActual=1;
        ModeloCSD.reduccionTiempo(1);
        Juego.LblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");
    }else if(Juego.pistaActual!=1 &&
Juego.viajeCorrecto.equals("Incorrecto")){
        Juego.LblImgText.setText("");
        Juego.LblPista.setText("<html><h4 align='justify'>No
he visto a nadie con esa descripción.</h4></html>");
        Juego.LblImgText.setIcon(new
ImageIcon("./img/pistas/pistaMAeropuerto.jpg"));
        Juego.pistaActual=1;
        ModeloCSD.reduccionTiempo(1);
        Juego.LblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");
    }
    frameJuego.setEnabled(true);
    dispose();
}catch (Exception e1) {
    ModeloCSD.finalJuego(frameJuego, this);
}
}
if(e.getSource().equals(lblBanco)) {
    try {
        if(Juego.pistaActual!=2 &&
Juego.viajeCorrecto.equals("Correcto")) {
            tipoPista=rnd.nextInt(2);
            Juego.LblImgText.setText("");
            if(Juego.opcionInvestigar!=0) {
                Juego.LblPista.setText("<html><h4
align='justify'>Lo vi pedir, muy nervioso, cambiar todo su dinero a
"+Juego.atributoBanco+"</h4></html>");
            }else {
                Juego.LblPista.setText("<html><h4
align='justify'>No pude fijarme bien pero creo que el ladrón
"+Juego.atributoBanco+"</h4></html>");
            }
            if(tipoPista==0) {
                Juego.LblImgText.setIcon(new
ImageIcon("./img/pistas/pistaMBanco.jpg"));
            }else if(tipoPista==1) {
                Juego.LblImgText.setIcon(new
ImageIcon("./img/pistas/pistaHBanco.jpg"));
            }
        }
    }
}

```



```

    }
    Juego.pistaActual=2;
    ModeloCSD.reduccionTiempo(1);
    Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");
    }else if(Juego.pistaActual!=2 &&
Juego.viajeCorrecto.equals("Incorrecto")){
    Juego.lblImgText.setText("");
    Juego.lblPista.setText("<html><h4 align='justify'>No
me suena. Siento no poder ayudarle.</h4></html>");
    Juego.lblImgText.setIcon(new
ImageIcon("./img/pistas/pistaMBanco.jpg"));
    Juego.pistaActual=2;
    ModeloCSD.reduccionTiempo(1);
    Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");
    }
    frameJuego.setEnabled(true);
    dispose();
}catch (Exception e1) {
    ModeloCSD.finalJuego(frameJuego, this);
}
}
if(e.getSource().equals(lblMuseo)) {
    try {
        if(Juego.pistaActual!=3 &&
Juego.viajeCorrecto.equals("Correcto")) {
            tipoPista=rnd.nextInt(2);
            Juego.lblImgText.setText("");
            if(Juego.opcionInvestigar!=2) {
                Juego.lblPista.setText("<html><h4
align='justify'>Cuando reponia los estantes lo vi en la sección de monumentos
del mundo y estaba ojeando un libro donde ponía:
"+Juego.atributoMuseo+"</h4></html>");
            }else {
                Juego.lblPista.setText("<html><h4
align='justify'>No pude fijarme bien pero creo que el ladrón
"+Juego.atributoMuseo+"</h4></html>");
            }
            if(tipoPista==0) {
                Juego.lblImgText.setIcon(new
ImageIcon("./img/pistas/pistaMMuseo.jpg"));
            }else if(tipoPista==1) {
                Juego.lblImgText.setIcon(new
ImageIcon("./img/pistas/pistaHMuseo.jpg"));
            }
            Juego.pistaActual=3;
            ModeloCSD.reduccionTiempo(1);
            Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");

```



```

        }else if(Juego.pistaActual!=3 &&
Juego.viajeCorrecto.equals("Incorrecto")){
            Juego.lblImgText.setText("");
            Juego.lblPista.setText("<html><h4 align='justify'>No
se nada sobre el caso al que se refiere. Disculpe, tengo prisa.</h4></html>");
            Juego.lblImgText.setIcon(new
ImageIcon("./img/pistas/pistaHMuseo.jpg"));
            Juego.pistaActual=3;
            ModeloCSD.reduccionTiempo(1);
            Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
"+Juego.horas+":00");
        }
        frameJuego.setEnabled(true);
        dispose();
    }catch (Exception e1) {
        ModeloCSD.finalJuego(frameJuego, this);
    }
}

@Override
public void mouseEntered(MouseEvent e) {
    if(e.getComponent().equals(lblAerop)) {
        lblAerop.setBackground(Color.white);
        lblAerop.setForeground(Color.black);
    }
    if(e.getComponent().equals(lblBanco)) {
        lblBanco.setBackground(Color.white);
        lblBanco.setForeground(Color.black);
    }
    if(e.getComponent().equals(lblMuseo)) {
        lblMuseo.setBackground(Color.white);
        lblMuseo.setForeground(Color.black);
    }
}

@Override
public void mouseExited(MouseEvent e) {
    if(e.getComponent().equals(lblAerop)) {
        lblAerop.setBackground(Color.black);
        lblAerop.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblBanco)) {
        lblBanco.setBackground(Color.black);
        lblBanco.setForeground(Color.white);
    }
    if(e.getComponent().equals(lblMuseo)) {
        lblMuseo.setBackground(Color.black);
        lblMuseo.setForeground(Color.white);
    }
}

```



```
}

@Override
public void mousePressed(MouseEvent e) {
}
@Override
public void mouseReleased(MouseEvent e) {
}

@Override
public void actionPerformed(ActionEvent e) {
    if(btnAtras.equals(e.getSource())) {
        frameJuego.setEnabled(true);
        dispose();
    }
}
}
```



Juego.java

```
package CarmenSandiego;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class Juego extends JFrame implements WindowListener, MouseListener,
ActionListener {

    private static final long serialVersionUID = 1L;
    //TIEMPO JUEGO
    static String semana [] =
{"Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"};
    static int dia=0;
    static int horas=9;
    //CIUDAD
    static int ciudadActual=0;
    //JUEGO
    static int puntosCaso;
    static boolean ordenCaptura=false;
    static boolean capturaLadron=false;
    static String nick;
    //CONTROL PISTAS
    static int opcionInvestigar=-1;
    static int pistaActual=-1;
    static String atributoBanco, atributoAeropuerto, atributoMuseo;
    //CONTROL VIAJES
    static int viajeActual=-1;
    static String viajeCorrecto="Correcto";
    static String ciudad1, ciudad2, ciudad3;
    //Etiquetas
    static JLabel lblCiudad = new JLabel("CUARTEL GENERAL");
    static JLabel lblTiempo= new JLabel(semana[dia]+", "+horas+":00");
    static JLabel lblImgCiudad = new JLabel();
```



```

static JLabel lblImgText = new JLabel();
static JLabel lblPista= new JLabel();
JLabel lblImgInv = new JLabel();
JLabel lblImgViaj = new JLabel();
JLabel lblImgLadron = new JLabel();
JLabel lblImgOrden = new JLabel();
JLabel lblInicio= new JLabel();

//Boton
JButton btnAceptarCaso = new JButton("Aceptar caso");

//Paneles y Diálogos
JPanel pnl1 = new JPanel();
JPanel pnlLugFech = new JPanel();
JPanel pnlImgCiudad = new JPanel();
JPanel pnlGame = new JPanel();
//Fuente
Font fuente = new Font("Calibri", 3, 20);

/*Añadir imágenes*/
//Logo
ImageIcon imagen= new ImageIcon("./img/icono.png");
Image img = imagen.getImage();
//Icono Investigar
ImageIcon imagenHuella= new ImageIcon("./img/huella.png");
//Icono Investigar
ImageIcon imagenMundo= new ImageIcon("./img/mundo.jpg");
//Icono Investigar
ImageIcon imagenLadron= new ImageIcon("./img/ladron.jpg");
//Icono Orden
ImageIcon imagenOrden= new ImageIcon("./img/orden.png");

public Juego(String nick){
    //Reiniciación juego
    dia=0;
    horas=9;
    ciudadActual=0;
    ordenCaptura=false;
    capturaLadron=false;
    opcionInvestigar=-1;
    pistaActual=-1;
    viajeActual=-1;
    lblCiudad.setText("CUARTEL GENERAL");
    lblTiempo.setText(semana[dia]+", "+horas+":00");
    lblPista.setText("");
    Datos.indiceGenero=0;
    Datos.indiceCaracteristica=0;
    Datos.indiceCoche=0;
    Datos.indiceHobby=0;
}

```




```
Datos.indicePelo=0;
Juego.nick=nick;
//Formacion pantalla
setTitle("Carmen Sandiego");
setLayout (new GridLayout(1,1));
setSize(645, 485);
setLocationRelativeTo(null);
setResizable(false);
setIconImage(img);
pnl1.setLayout(null);
pnl1.setBackground(Color.black);
//Panel Lugar y fecha
pnlLugFech.setLayout(null);
pnlLugFech.setBorder(BorderFactory.createLineBorder(Color.white,4));
pnlLugFech.setBackground(Color.black);
pnlLugFech.setBounds(5, 5, 308, 60);
pnlLugFech.add(lblCiudad);
lblCiudad.setFont(fuentes);
lblCiudad.setForeground(Color.white);
lblCiudad.setBounds(10, 10, 170, 25);
pnlLugFech.add(lblTiempo);
lblTiempo.setFont(fuentes);
lblTiempo.setForeground(Color.white);
lblTiempo.setBounds(10, 30, 150, 25);
pnl1.add(pnlLugFech);
//Panel Ciudad Documentos
pnlImgCiudad.setLayout(null);

pnlImgCiudad.setBorder(BorderFactory.createLineBorder(Color.white,4));
pnlImgCiudad.setBackground(Color.black);
pnlImgCiudad.setBounds(5, 70, 308, 380);
lblImgCiudad.setBounds(4, 4, 308, 372);
pnlImgCiudad.add(lblImgCiudad);
pnl1.add(pnlImgCiudad);
//Panel Juego
pnlGame.setLayout(null);
pnlGame.setBorder(BorderFactory.createLineBorder(Color.white,4));
pnlGame.setBackground(Color.black);
pnlGame.setBounds(320, 5, 313, 366);
lblImgText.setBounds(10, 10, 293, 345);
pnlGame.add(lblImgText);
pnl1.add(lblPista);
lblPista.setBounds(360, -60, 200, 300);
pnl1.add(pnlGame);
//Investigar
lblImgInv.setIcon(imagenHuella);
lblImgInv.setBounds(320, 375, 73, 73);
pnl1.add(lblImgInv);
//Viajar
lblImgViaj.setIcon(imagenMundo);
```



```

lblImgViaj.setBounds(400, 375, 73, 73);
pn11.add(lblImgViaj);
//Ladron
lblImgLadron.setIcon(imagenLadron);
lblImgLadron.setBounds(480, 375, 73, 73);
pn11.add(lblImgLadron);
//Orden
lblImgOrden.setIcon(imagenOrden);
lblImgOrden.setBounds(560, 375, 73, 73);
pn11.add(lblImgOrden);

//INICIO
lblImgText.setVisible(false);
lblImgInv.setVisible(false);
lblImgViaj.setVisible(false);
lblImgLadron.setVisible(false);
lblImgOrden.setVisible(false);
lblImgCiudad.setVisible(false);
ModeloCSD.conectaBase();
Iniciacion.IniciacionJuego();
Iniciacion.SelecCiudadesCorrecto();
lblInicio.setForeground(Color.white);
lblInicio.setBounds(15, 35, 270, 300);
lblInicio.setText("<html><h1 align='center'>**** Nuevo caso ****</h1>
<h4>¡Bienvenido Inspector!</h4><h4 align='justify'>¡Ha ocurrido un gran robo en
"+Iniciacion.ciudadesCaso[0].getNombreCiudad()+"!</h4><h4 align='justify'>Todos
nuestros agentes están destinados en otras misiones, por lo que requerimos de su
servicio urgentemente.</h4><h4 align='justify'>Sobra decir que la recompensa
laboral estará dictaminada según lo eficiente que sea resolviendo el
caso.</h4><h3>¡Contamos con usted!</h3></html>");
pn1ImgCiudad.add(lblInicio);
ModeloCSD.desconectaBase();
btnAceptarCaso.setBounds(400, 390, 150, 40);
pn11.add(btnAceptarCaso);

add(pn11);
//Listeners
lblImgInv.addMouseListener(this);
lblImgViaj.addMouseListener(this);
lblImgLadron.addMouseListener(this);
lblImgOrden.addMouseListener(this);
btnAceptarCaso.addActionListener(this);
addWindowListener(this);
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {

```



```

        if(e.getSource().equals(btnAceptarCaso)) {

            lblCiudad.setText(Iniciacion.ciudadesCaso[ciudadActual].getNombreCiudad());
            lblImgCiudad.setIcon(new
ImageIcon("./img/ciudades/"+Iniciacion.ciudadesCaso[ciudadActual].getFotoCiudad(
)));
            lblImgText.setIcon(null);
            lblImgText.setForeground(Color.white);
            lblImgText.setText("<html><h2
align='justify'>" + Iniciacion.ciudadesCaso[ciudadActual].getDescripcionCiudad() + "
</h2></html>");
            lblImgText.setVisible(true);
            lblImgInv.setVisible(true);
            lblImgViaj.setVisible(true);
            lblImgLadron.setVisible(true);
            lblImgOrden.setVisible(true);
            lblImgCiudad.setVisible(true);
            lblInicio.setVisible(false);
            btnAceptarCaso.setVisible(false);
        }
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        if(e.getSource().equals(lblImgInv)) {
            new Investigar(this);
        }
        if(e.getSource().equals(lblImgViaj)) {
            new Viaje(this);
        }
        if(e.getSource().equals(lblImgLadron)) {
            new Sospechosos(this);
        }
        if(e.getSource().equals(lblImgOrden)) {
            new Datos(this);
        }
    }

    @Override
    public void mouseEntered(MouseEvent e) {
    }
    @Override
    public void mouseExited(MouseEvent e) {
    }
    @Override
    public void mousePressed(MouseEvent e) {
    }
    @Override
    public void mouseReleased(MouseEvent e) {
    }

```



```
}  
@Override  
public void windowActivated(WindowEvent arg0) {  
}  
@Override  
public void windowClosed(WindowEvent arg0) {  
}  
  
@Override  
public void windowClosing(WindowEvent arg0) {  
    dispose();  
}  
  
@Override  
public void windowDeactivated(WindowEvent arg0) {  
}  
@Override  
public void windowDeiconified(WindowEvent arg0) {  
}  
@Override  
public void windowIconified(WindowEvent arg0) {  
}  
@Override  
public void windowOpened(WindowEvent arg0) {  
}  
  
}
```



Menu.java

```
package CarmenSandiego;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.HeadlessException;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

public class Menu extends JFrame implements WindowListener, MouseListener,
ActionListener {

    private static final long serialVersionUID = 1L;
    ResultSet resultado;
    String nick;
    /*Componentes básicos*/
    //Etiquetas
    JLabel lblApodo = new JLabel("Introduce tu nick de jugador:");
    JLabel lblPartida= new JLabel("Partida nueva");
    JLabel lblIcono= new JLabel();
    JLabel lblRanking= new JLabel("Ranking");
    JLabel lblTitulo= new JLabel("CARMEN SANDIEGO");
    JLabel lblAyuda= new JLabel("Ayuda");
    //Textos
    JTextField txtNick = new JTextField();
    //Botón
    JButton btnAcep = new JButton("Aceptar");
    JButton btnCanc = new JButton("Cancelar");
    //Paneles y Diálogos
    JPanel pnlFondo = new JPanel();
```



```

JPanel pnlFnd1= new JPanel();
JPanel pnlFnd2= new JPanel();
JPanel pnlFnd3= new JPanel();
JPanel pnlNick = new JPanel();
JDialog dlgNick = new JDialog(this);
//Definir un color y fuente
Color colorFondo=new Color(230, 176, 170);
Color colorMenu=new Color(192, 57, 43);
Font fuenteTitulo = new Font("Serif", 3, 25);
Font fuenteSubTitulo = new Font("Serif", 3, 15);
/*Añadir imágenes*/
//Logo
ImageIcon imagen= new ImageIcon("./img/icono.png");
Image img = imagen.getImage();
Image newimg = img.getScaledInstance(150, 150,
java.awt.Image.SCALE_SMOOTH);
ImageIcon imagentnf = new ImageIcon(newimg);
//Separador
JSeparator sep1 = new JSeparator();
JSeparator sep2 = new JSeparator();
JSeparator sep3 = new JSeparator();

public Menu(){
    /*PRINCIPAL*/
    setTitle("Carmen Sandiego");
    setLayout (new GridLayout(1,1));
    setSize(290, 400);
    setLocationRelativeTo(null);
    setResizable(false);
    setIconImage(img);
    pnlFondo.setLayout(null);
    pnlFondo.setBackground(colorFondo);
    pnlFnd1.setLayout(null);
    pnlFnd1.setBackground(colorFondo);
    pnlFondo.add(pnlFnd1);
    pnlFnd2.setLayout(null);
    pnlFnd2.setBackground(colorFondo);
    pnlFnd3.setLayout(null);
    pnlFnd3.setBackground(colorFondo);
    lblIcono.setIcon(imagentnf);
    pnlFondo.add(lblIcono);
    lblIcono.setBounds(70,15,150,150);
    lblTitulo.setFont(fuenteTitulo);
    pnlFondo.add(lblTitulo);
    lblTitulo.setBounds(15, 155, 300, 60);
    //Separador
    sep1.setOrientation(SwingConstants.HORIZONTAL);
    pnlFondo.add(sep1);
    sep1.setBounds(0, 220, 290, 8);
    pnlFnd1.add(lblPartida);

```




```
lblPartida.setForeground(Color.BLACK);
lblPartida.setBounds(100,10, 80, 30);
pnlFnd1.setBounds(0, 222, 290, 47);
//Separador
sep2.setOrientation(SwingConstants.HORIZONTAL);
pnlFondo.add(sep2);
sep2.setBounds(0, 270, 290, 8);
pnlFnd2.add(lblRanking);
lblRanking.setForeground(Color.BLACK);
lblRanking.setBounds(115,10, 47, 30);
pnlFondo.add(pnlFnd2);
pnlFnd2.setBounds(0, 272, 290, 48);
//Separador
sep3.setOrientation(SwingConstants.HORIZONTAL);
pnlFondo.add(sep3);
sep3.setBounds(0, 320, 290, 8);
pnlFnd3.add(lblAyuda);
lblAyuda.setForeground(Color.BLACK);
lblAyuda.setBounds(120,10, 47, 30);
pnlFondo.add(pnlFnd3);
pnlFnd3.setBounds(0, 322, 290, 48);
add(pnlFondo);

/*DIÁLOGO NICK*/
dlgNick.setTitle("Juego nuevo");
dlgNick.setIconImage(img);
dlgNick.setLayout(new GridLayout(1,1));
dlgNick.setSize(270,140);
dlgNick.setResizable(false);
dlgNick.setLocationRelativeTo(this);
pnlNick.setLayout(null);
pnlNick.add(lblApodo);
lblApodo.setBounds(5, 5, 200, 25);
pnlNick.add(txtNick);
txtNick.setBounds(5, 30, 250, 25);
pnlNick.add(btnAcep);
btnAcep.setBounds(5, 70, 100, 30);
pnlNick.add(btnCanc);
btnCanc.setBounds(155, 70, 100, 30);
pnlNick.setBackground(colorFondo);
dlgNick.add(pnlNick);

/*LISTENERS*/
pnlFnd1.addMouseListener(this);
pnlFnd2.addMouseListener(this);
pnlFnd3.addMouseListener(this);
btnAcep.addActionListener(this);
btnCanc.addActionListener(this);
addWindowListener(this);
setVisible(true);
```



```
}

public static void main(String[] args) {
    new Menu();
}

@Override
public void mouseClicked(MouseEvent arg0) {
    if(arg0.getSource().equals(pnlFnd1)) {
        dlgNick.setVisible(true);
    }
    if(arg0.getSource().equals(pnlFnd2)) {

        new Ranking();
    }
    if(arg0.getSource().equals(pnlFnd3)) {
        ModeloCSD.ficheroAyuda();
    }
}

@Override
public void mouseEntered(MouseEvent e) {
    if(e.getSource().equals(pnlFnd1)) {
        pnlFnd1.setBackground(colorMenu);
        lblPartida.setForeground(Color.WHITE);
    }
    if(e.getSource().equals(pnlFnd2)) {
        pnlFnd2.setBackground(colorMenu);
        lblRanking.setForeground(Color.WHITE);
    }
    if(e.getSource().equals(pnlFnd3)) {
        pnlFnd3.setBackground(colorMenu);
        lblAyuda.setForeground(Color.WHITE);
    }
}

@Override
public void mouseExited(MouseEvent e) {
    if(e.getSource().equals(pnlFnd1)) {
        pnlFnd1.setBackground(colorFondo);
        lblPartida.setForeground(Color.BLACK);
    }
    if(e.getSource().equals(pnlFnd2)) {
        pnlFnd2.setBackground(colorFondo);
        lblRanking.setForeground(Color.BLACK);
    }
    if(e.getSource().equals(pnlFnd3)) {
        pnlFnd3.setBackground(colorFondo);
        lblAyuda.setForeground(Color.BLACK);
    }
}
```



```

    }

    @Override
    public void mousePressed(MouseEvent e) {
    }
    @Override
    public void mouseReleased(MouseEvent e) {
    }
    @Override
    public void windowActivated(WindowEvent arg0) {
    }
    @Override
    public void windowClosed(WindowEvent arg0) {
    }

    @Override
    public void windowClosing(WindowEvent arg0) {
        System.exit(0);
    }

    @Override
    public void windowDeactivated(WindowEvent arg0) {
    }
    @Override
    public void windowDeiconified(WindowEvent arg0) {
    }
    @Override
    public void windowIconified(WindowEvent e) {
    }
    @Override
    public void windowOpened(WindowEvent e) {
    }

    @Override
    public void actionPerformed(ActionEvent arg0) {
        if(btnAcep.equals(arg0.getSource())) {
            try {
                nick=txtNick.getText();
                String sentNick = "SELECT * FROM jugadores WHERE
apodoJugador = '"+ nick +"'";
                ModeloCSD.conectaBase();
                resultado = ModeloCSD.buscarDatos(sentNick);
                if(nick.equals("")) {
                    JOptionPane.showMessageDialog(dlgNick,"No se ha
introducido ningun nick ", "Mensaje de ERROR",JOptionPane.ERROR_MESSAGE);
                } else

                    if(resultado.next()) {

```



```

        int opcion =
JOptionPane.showConfirmDialog(dlgNick,"Nick ya existente, ¿está seguro de usar
ese nick?","Acceso nuevo",JOptionPane.YES_NO_OPTION);
        if(opcion==JOptionPane.YES_OPTION) {
            JOptionPane.showMessageDialog(dlgNick,"Se
ha accedido correctamente. ¡Bienvenido de nuevo!");
            dlgNick.setVisible(false);
            dispose();
            new Juego(nick);
        }
    }else {
        int opcion =
JOptionPane.showConfirmDialog(dlgNick,"Nick no encontrado, ¿desea registrarse
como nuevo inspector?","Acceso nuevo",JOptionPane.YES_NO_OPTION);
        if(opcion==JOptionPane.YES_OPTION) {
            String insJugador = "INSERT INTO jugadores
(apodoJugador, puntosTotalesJugador) VALUES ('"+ nick +"',0);";
            ModeloCSD.datosABM(insJugador);
            dlgNick.setVisible(false);
            dispose();
            new Juego(nick);
        }
    }
    ModeloCSD.desconectaBase();
} catch (HeadlessException | SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
if(btnCanc.equals(arg0.getSource())) {
    dlgNick.setVisible(false);
}
}
}
}

```



ModeloCSD.java

```

package CarmenSandiego;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Random;

import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

public class ModeloCSD {
    /*#####*/
    /*## VARIABLES ##*/
    /*#####*/

    /*>> BASE DE DATOS <<*/
    static String driver = "com.mysql.jdbc.Driver";
    static String url =
"jdbc:mysql://localhost:3306/carmensandiego?autoReconnect=true&useSSL=false";
    static Connection connection = null;
    static Statement statement = null;
    static ResultSet rs = null;
    //USUARIO BASE DE DATOS
    static String LoginBD= "remoto";
    static String passwordBD= "Stodium2018;";

    /*>> RANDOM <<*/
    static Random rnd=new Random();
    static int numRnd;

    static Ciudad ciudadIncorrecta;
    static int puntosFinal;
    /*#####*/
    /*## METODOS ##*/
    /*#####*/

    /*>> METODOS BASE DE DATOS <<*/
    //CONECTA MYSQL
    public static void conectaBase() {
        try{
            //CARGAR LOS CONTROLADORES PARA EL ACCESO A LA BASE DE DATOS

```



```

        Class.forName(driver);
        //ESTABLECER LA CONEXIÓN CON LA BASE DE DATOS
        connection = DriverManager.getConnection(url, loginBD,
passwordBD);
        //CREAR UNA SENTENCIA
        statement = connection.createStatement();
    }catch (ClassNotFoundException cnfe){
        System.out.println("Error 1: "+cnfe.getMessage());
    }catch (SQLException sqle){
        System.out.println("Error 2: "+sqle.getMessage());
    }
}
//DESCONECTA MYSQL
public static void desconectaBase() {
    try{
        if(connection!=null){
            connection.close();
        }
    }catch (SQLException e){
        System.out.println("Error 3: "+e.getMessage());
    }
}

//BUSCA DATOS EN MYSQL MEDIANTE SELECT
public static ResultSet buscarDatos(String sent) {
    try {
        rs=statement.executeQuery(sent);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return rs;
}

//DA DE ALTA, BAJA y MODIFICA LOS DATOS EN MYSQL
public static void datosABM(String sent) {
    try {
        statement.executeUpdate(sent);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

//TABLA SEGUN MYSQL
public static void rellenarTabla(String sent, JTable tabla) {
    try {
        DefaultTableModel modeloTabla= (DefaultTableModel)
tabla.getModel();
        rs = statement.executeQuery(sent);
        int columnas = rs.getMetaData().getColumnCount();

```



```

        for (int i = 0; i<columnas; i++){
modeloTabla.addColumn(rs.getMetaData().getColumnLabel(i+1));
        }
        while (rs.next()){
            Object datos[] = new Object[columnas];
            for(int i=0; i<columnas; i++) {
                datos[i] = rs.getString(i+1);
            }
            modeloTabla.addRow(datos);
        }
    } catch (SQLException e) {
        System.out.println("Error rellenar tabla");
        e.printStackTrace();
    }
}
//DATOS
public static String atributoLadronRnd() {
    numRnd=rnd.nextInt(5);
    String atributo="";
    switch (numRnd) {
        case 0:
            atributo = "era de genero "+Iniciacion.generoLadron;
            break;
        case 1:
            atributo = "tenía el pelo "+Iniciacion.peloLadron;
            break;
        case 2:
            atributo = "dijo que le gustaba el "+Iniciacion.hobbyLadron;
            break;
        case 3:
            atributo = "conducía "+Iniciacion.cocheLadron;
            break;
        case 4:
            atributo = "tenía "+Iniciacion.caracteristicaLadron;
            break;
    }
    return atributo;
}

public static void reduccionTiempo(int tiempo) {
    int tiempoTotal=Juego.horas+tiempo;
    if(tiempoTotal>=22) {
        Juego.dia++;
        Juego.horas=9;
    }else {
        Juego.horas=Juego.horas+tiempo;
    }
}
}

```




```

    public static void cambioCiudadCorrecta() {
        if(Juego.viajeCorrecto.equals("Correcto")) {
            Juego.ciudadActual++;
        }else{
            Juego.viajeCorrecto="Correcto";
        }

        Juego.lblCiudad.setText(Iniciacion.ciudadesCaso[Juego.ciudadActual].getNombreCiudad());
        Juego.lblImgCiudad.setIcon(new
        ImageIcon("./img/ciudades/"+Iniciacion.ciudadesCaso[Juego.ciudadActual].getFotoCiudad()));
        Juego.lblImgText.setIcon(null);
        Juego.lblImgText.setText("<html><h2
        align='justify'>" +Iniciacion.ciudadesCaso[Juego.ciudadActual].getDescripcionCiudad()+"</h2></html>");
        Juego.lblPista.setText("");
        ModeloCSD.reduccionTiempo(5);
        Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
        "+Juego.horas+":00");
        Juego.opcionInvestigar=-1;
        Juego.pistaActual=-1;
        Juego.viajeActual=-1;
    }

    public static void cambioCiudadIncorrecta(String errorCiudad) {
        if(errorCiudad.equals(Iniciacion.ciudadesRnd[0].getNombreCiudad())){
            ciudadIncorrecta=Iniciacion.ciudadesRnd[0];
        }else {
            ciudadIncorrecta=Iniciacion.ciudadesRnd[1];
        }
        Juego.viajeCorrecto="Incorrecto";
        Juego.lblCiudad.setText(ciudadIncorrecta.getNombreCiudad());
        Juego.lblImgCiudad.setIcon(new
        ImageIcon("./img/ciudades/"+ciudadIncorrecta.getFotoCiudad()));
        Juego.lblImgText.setIcon(null);
        Juego.lblImgText.setText("<html><h2
        align='justify'>" +ciudadIncorrecta.getDescripcionCiudad()+"</h2></html>");
        Juego.lblPista.setText("");
        ModeloCSD.reduccionTiempo(5);
        Juego.lblTiempo.setText(Juego.semana[Juego.dia]+",
        "+Juego.horas+":00");
        Juego.opcionInvestigar=-1;
        Juego.pistaActual=-1;
        Juego.viajeActual=-1;
    }

    public static void calculoPuntos() {

```



```

        if(Juego.ordenCaptura==true &&
Datos.sospechosoJugador==Iniciacion.idLadron) {
    Juego.capturaLadron=true;
    switch (Juego.dia) {
    case 0:
        Juego.puntosCaso=(23-Juego.horas)+350;
        break;
    case 1:
        Juego.puntosCaso=(23-Juego.horas)+400;
        break;
    case 2:
        Juego.puntosCaso=(23-Juego.horas)+350;
        break;
    case 3:
        Juego.puntosCaso=(23-Juego.horas)+300;
        break;
    case 4:
        Juego.puntosCaso=(23-Juego.horas)+250;
        break;
    case 5:
        Juego.puntosCaso=(23-Juego.horas)+200;
        break;
    case 6:
        Juego.puntosCaso=(23-Juego.horas)+150;
        break;
    }
    }else {
        Juego.puntosCaso=0;
    }
}

public static void finalJuego(JFrame frameJuego, JDialog ventanaActual) {
    try {
        calculoPuntos();
        if(Juego.ordenCaptura==false) {
            JOptionPane.showMessageDialog(frameJuego,"<html><p>Aun no
has logrado atrapar al ladron y ya no podemos aplazarlo más.</p><p>Otro de
nuestros agentes se encargará a partir de aquí.</p><CENTER><p>Has conseguido
"+Juego.puntosCaso+" puntos.</p></CENTER></html>", "FIN DE
PARTIDA",JOptionPane.ERROR_MESSAGE);
        }else if (Juego.ordenCaptura==true &&
Datos.sospechosoJugador!=Iniciacion.idLadron) {
            JOptionPane.showMessageDialog(frameJuego,"<html><p>La orden
de arresto esta a nombre de otro ladrón.</p><p>Otro de nuestros agentes se
encargará a partir de aquí.</p><CENTER><p>Has conseguido "+Juego.puntosCaso+"
puntos.</p></CENTER></html>", "FIN DE PARTIDA",JOptionPane.ERROR_MESSAGE);
        }else if (Juego.ordenCaptura==true &&
Juego.capturaLadron==false) {
            JOptionPane.showMessageDialog(frameJuego,"<html><p>Has
estado cerca, pero aun no has logrado atrapar al ladron y ya no podemos

```



```

aplazarlo más.</p><p>Otro de nuestros agentes se encargará a partir de
aquí.</p><CENTER><p>Has conseguido "+Juego.puntosCaso+"
puntos.</p></CENTER></html>", "FIN DE PARTIDA", JOptionPane.ERROR_MESSAGE);
    }else if(Juego.ordenCaptura==true &&
Datos.sospechosoJugador!=Iniciacion.idLadron){
        JOptionPane.showMessageDialog(frameJuego, "<html><p>Te has
equivocado al emitir la orden de arresto y por ello no podemos
encarcelarlo.</p><p>Otro de nuestros agentes se encargará a partir de
aquí.</p><CENTER><p>Has conseguido "+Juego.puntosCaso+"
puntos.</p></CENTER></html>", "FIN DE PARTIDA", JOptionPane.ERROR_MESSAGE);
    }else if(Juego.ordenCaptura==true &&
Datos.sospechosoJugador==Iniciacion.idLadron){

        JOptionPane.showMessageDialog(frameJuego, "<html><p>¡Enhorabuena! Has
logrado atrapar al ladron.</p><p>Esperamos grandes cosas de ti en futuros
casos.</p><CENTER><p>Has conseguido "+Juego.puntosCaso+"
puntos.</p></CENTER></html>", "FIN DE PARTIDA", JOptionPane.INFORMATION_MESSAGE);
    }
    conectaBase();
    datosABM("UPDATE casos SET puntosCaso='"+Juego.puntosCaso+"
WHERE idCaso="+Iniciacion.idPartida+"");
    rs=buscarDatos("SELECT puntosTotalesJugador FROM jugadores WHERE
apodoJugador='"+Juego.nick+"'");
    rs.next();
    puntosFinal=Integer.parseInt(rs.getString(1))+Juego.puntosCaso;
    datosABM("UPDATE jugadores SET
puntosTotalesJugador="+puntosFinal+" WHERE apodoJugador='"+Juego.nick+"'");
    desconectaBase();
    frameJuego.dispose();
    ventanaActual.dispose();
    new Menu();
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
//ABRIR LA AYUDA
public static void ficheroAyuda() {
    try{
        Runtime.getRuntime().exec("hh.exe
./ayudaCarmenSanDiego/ayudaCSD.chm");
    }catch (IOException e){
        e.printStackTrace();
    }
}
}
}

```



Ranking.java

```

package CarmenSandiego;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

public class Ranking extends JDialog implements WindowListener{

    private static final long serialVersionUID = 1L;
    String sentenciaRellenarRank ="SELECT apodoJugador AS 'Nick',
puntosTotalesJugador AS 'Puntuación' FROM jugadores ORDER BY 2 DESC LIMIT 10;";
    /*Componentes básicos*/
    //Etiquetas
    JLabel lblRankingTitulo= new JLabel("Ranking");
    //Tabla
    DefaultTableModel modeloTablaRank =new DefaultTableModel();
    JTable tablaRank = new JTable(modeloTablaRank);
    //Paneles y Diálogos
    JPanel pnl1 = new JPanel();
    JPanel pnl2 = new JPanel();
    //Definir un color y fuente
    Font fuenteTitulo = new Font("Serif", 3, 25);
    Color colorFondo=new Color(230, 176, 170);
    /*Añadir imágenes*/
    //Logo
    ImageIcon imagen= new ImageIcon("./img/icono.png");
    Image img = imagen.getImage();

    Ranking(){
        setTitle("Ranking");
        setLayout (new GridLayout(1,1));
        setSize(350,310);
        setIconImage(img);
        setLocationRelativeTo(null);
        setResizable(false);
    }
}

```



```

    pnl1.setLayout(null);
    pnl2.setLayout(new BorderLayout());
    // COMPONENTES
    lblRankingTitulo.setFont(fuenteTitulo);
    pnl1.add(lblRankingTitulo);
    lblRankingTitulo.setBounds(130, 10, 300, 50);
    // TABLA
    ModeloCSD.conectaBase();
    ModeloCSD.rellenarTabla(sentenciaRellenarRank, tablaRank);
    ModeloCSD.desconectaBase();
    pnl2.add(new JScrollPane(tablaRank), BorderLayout.CENTER);
    pnl1.add(pnl2);
    pnl2.setBounds(20,70, 300, 183);
    pnl1.setBackground(colorFondo);
    add(pnl1);
    /*>> LISTENERS Y VISIBILIDAD <<*/
    addWindowListener(this);
    setVisible(true);
}

@Override
public void windowActivated(WindowEvent e) {
}
@Override
public void windowClosed(WindowEvent e) {
}
@Override
public void windowClosing(WindowEvent e) {
    // TODO Auto-generated method stub
    dispose();
}
@Override
public void windowDeactivated(WindowEvent e) {
}
@Override
public void windowDeiconified(WindowEvent e) {
}
@Override
public void windowIconified(WindowEvent e) {
}
@Override
public void windowOpened(WindowEvent e) {
}
}

```



Sospechosos.java

```

package CarmenSandiego;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

public class Sospechosos extends JDialog implements ActionListener{

    private static final long serialVersionUID = 1L;

    JFrame mn = new JFrame();
    String sentencia="SELECT * FROM sospechosos ORDER BY rand()";
    String lista= "<html><br/><CENTER><h2><b>--- LISTA SOSPECHOSOS ---</b></h2></CENTER><br/><br/>";
    ResultSet resultado;
    //Paneles y Diálogos
    JPanel pnl1 = new JPanel();
    JScrollPane scroll = new JScrollPane(pnl1);
    //Etiquetas
    JLabel lblDts = new JLabel();
    //Boton
    JButton btnAtras = new JButton("Atrás");

    Sospechosos(JFrame jf){
        mn=jf;
        mn.setEnabled(false);
        setLayout (new BorderLayout());
        setSize(430, 370);
        setLocationRelativeTo(null);
        setResizable(false);
        setUndecorated(true);
        pnl1.setLayout(new FlowLayout());
        pnl1.setBorder(BorderFactory.createLineBorder(Color.white,4));
        pnl1.setBackground(Color.black);
        add(btnAtras, BorderLayout.SOUTH);
        lblDts.setForeground(Color.WHITE);
    }
}

```



```

    try {
        ModeloCSD.conectaBase();
        resultado=ModeloCSD.buscarDatos(sentencia);
        while(resultado.next()) {

            lista=lista+"<hr/><hr/><CENTER><h2><b>" + resultado.getString(2)
                + "</b></h2></CENTER><hr/><h3><b>" + resultado.getString(6)
                + "</b></h3><p><b>Genero: </b>" + resultado.getString(3)
                + "</p><p><b>Color de pelo: </b>" + resultado.getString(4)
                + "</p><p><b>Hobby: </b>" + resultado.getString(5)
                + "</p><p><b>Seña identificativa:
</b>" + resultado.getString(8)
                + "</p><p><b>Coche:
</b>" + resultado.getString(7) + "</p><hr/><br/>";
        }
        lista=lista+"</html>";
        ModeloCSD.desconectaBase();
    } catch (SQLException e) {
    }
    lblDts.setText(lista);
    pnl1.add(lblDts);
    add(scroll, BorderLayout.CENTER);
    //Listeners
    btnAtras.addActionListener(this);
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    if(btnAtras.equals(e.getSource())) {
        mn.setEnabled(true);
        setVisible(false);
    }
}
}

```



Viaje.java

```
package CarmenSandiego;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.util.Random;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class Viaje extends JDialog implements MouseListener, ActionListener{
    private static final long serialVersionUID = 1L;
    JFrame frameJuego = new JFrame();
    Random rnd=new Random();

    //Paneles y Diálogos
    JPanel pnl1 = new JPanel();
    //Etiquetas
    JLabel lblCiudad1 = new JLabel();
    JLabel lblCiudad2= new JLabel();
    JLabel lblCiudad3 = new JLabel();
    //Boton
    JButton btnAtras = new JButton("Atrás");
    //Fuente
    Font fuente = new Font("Calibri", 3, 20);

    Viaje(JFrame jf){
        frameJuego=jf;
        frameJuego.setEnabled(false);
        setLayout (new GridLayout(1,1));
        setSize(250, 160);
        setLocationRelativeTo(null);
        setResizable(false);
        setUndecorated(true);
        pnl1.setLayout(null);
        pnl1.setBorder(BorderFactory.createLineBorder(Color.white,4));
        pnl1.setBackground(Color.black);
        pnl1.add(lblCiudad1);
        lblCiudad1.setFont(fuente);
        lblCiudad1.setForeground(Color.white);
    }
}
```



```

lblCiudad1.setBackground(Color.black);
lblCiudad1.setOpaque(true);
lblCiudad1.setBounds(30, 15, 180, 25);
pnl1.add(lblCiudad2);
lblCiudad2.setFont(fuente);
lblCiudad2.setForeground(Color.white);
lblCiudad2.setBackground(Color.black);
lblCiudad2.setOpaque(true);
lblCiudad2.setBounds(30, 45, 180, 25);
pnl1.add(lblCiudad3);
lblCiudad3.setFont(fuente);
lblCiudad3.setForeground(Color.white);
lblCiudad3.setBackground(Color.black);
lblCiudad3.setOpaque(true);
lblCiudad3.setBounds(30, 75, 180, 25);
pnl1.add(btnAtras);
btnAtras.setBounds(70, 115, 100, 30);
//RELLENAR VIAJES
try {
    if(Juego.viajeActual== -1 &&
Juego.viajeCorrecto.equals("Correcto")) {
        ModeloCSD.conectaBase();
        Iniciacion.SelecCiudadesExtras();
        ModeloCSD.desconectaBase();
        Juego.viajeActual=rnd.nextInt(3);
        switch (Juego.viajeActual) {
            case 0:

Juego.ciudad1=Iniciacion.ciudadesCaso[Juego.ciudadActual+1].getNombreCiudad
());

Juego.ciudad2=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad3=Iniciacion.ciudadesRnd[1].getNombreCiudad();
                break;
            case 1:

Juego.ciudad1=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad2=Iniciacion.ciudadesCaso[Juego.ciudadActual+1].getNombreCiudad
());

Juego.ciudad3=Iniciacion.ciudadesRnd[1].getNombreCiudad();
                break;
            case 2:

Juego.ciudad1=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad2=Iniciacion.ciudadesRnd[1].getNombreCiudad();

```



```

Juego.ciudad3=Iniciacion.ciudadesCaso[Juego.ciudadActual+1].getNombreCiudad
();
        break;
    }
    ModeloCSD.conectaBase();
    ModeloCSD.datosABM("INSERT INTO viajes (idCasoFK,
idCiudadActualFK, idCiudad1FK, idCiudad2FK, idCiudad3FK) VALUES
("+Iniciacion.idPartida+", '"+Iniciacion.ciudadesCaso[Juego.ciudadActual].getIdCi
udad()+"', '"+Iniciacion.ciudadesCaso[Juego.ciudadActual+1].getIdCiudad()+"', '"+I
niciacion.ciudadesRnd[0].getIdCiudad()+"', '"+Iniciacion.ciudadesRnd[1].getIdCiud
ad()+"'");
        ModeloCSD.desconectaBase();
    }else if(Juego.viajeActual== -1 &&
Juego.viajeCorrecto.equals("Incorrecto")) {
        ModeloCSD.conectaBase();
        Iniciacion.SelecCiudadesExtras();
        ModeloCSD.desconectaBase();
        Juego.viajeActual=rnd.nextInt(3);
        switch (Juego.viajeActual) {
            case 0:

Juego.ciudad1=Iniciacion.ciudadesCaso[Juego.ciudadActual].getNombreCiudad()
;

Juego.ciudad2=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad3=Iniciacion.ciudadesRnd[1].getNombreCiudad();
                break;
            case 1:

Juego.ciudad1=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad2=Iniciacion.ciudadesCaso[Juego.ciudadActual].getNombreCiudad()
;

Juego.ciudad3=Iniciacion.ciudadesRnd[1].getNombreCiudad();
                break;
            case 2:

Juego.ciudad1=Iniciacion.ciudadesRnd[0].getNombreCiudad();

Juego.ciudad2=Iniciacion.ciudadesRnd[1].getNombreCiudad();

Juego.ciudad3=Iniciacion.ciudadesCaso[Juego.ciudadActual].getNombreCiudad()
;
                break;
        }
        ModeloCSD.conectaBase();

```



```

        ModeloCSD.datosABM("INSERT INTO viajes (idCasoFK,
idCiudadActualFK, idCiudad1FK, idCiudad2FK, idCiudad3FK) VALUES
("+Iniciacion.idPartida+"+', '"+ModeloCSD.ciudadIncorrecta.getIdCiudad()+"', '"+Inic
iacion.ciudadesCaso[Juego.ciudadActual].getIdCiudad()+"', '"+Iniciacion.ciudadesR
nd[0].getIdCiudad()+"', '"+Iniciacion.ciudadesRnd[0].getIdCiudad()+"');");
        ModeloCSD.desconectaBase();
    }

    lblCiudad1.setText(Juego.ciudad1);
    lblCiudad2.setText(Juego.ciudad2);
    lblCiudad3.setText(Juego.ciudad3);
    add(pnl1);
    //Listeners
    btnAtras.addActionListener(this);
    lblCiudad1.addMouseListener(this);
    lblCiudad2.addMouseListener(this);
    lblCiudad3.addMouseListener(this);
    setVisible(true);
} catch (Exception e) {
    ModeloCSD.finalJuego(frameJuego, this);
}
}

@Override
public void actionPerformed(ActionEvent e) {
    if(btnAtras.equals(e.getSource())) {
        frameJuego.setEnabled(true);
        dispose();
    }
}

@Override
public void mouseClicked(MouseEvent e) {
    if(e.getComponent().equals(lblCiudad1)) {
        try {
            if(Juego.viajeCorrecto.equals("Incorrecto")) {

                if(lblCiudad1.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual].
getNombreCiudad())) {
                    ModeloCSD.cambioCiudadCorrecta();
                } else {
                    ModeloCSD.cambioCiudadIncorrecta(lblCiudad1.getText());
                }
            } else {
                if(lblCiudad1.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual+1].get
NombreCiudad())) {
                    if(Juego.ordenCaptura==true) {

```



```

        ModeloCSD.reduccionTiempo(4);
        ModeloCSD.finalJuego(frameJuego, this);
    }else {
        ModeloCSD.cambioCiudadCorrecta();
    }
}else {
    ModeloCSD.cambioCiudadIncorrecta(lblCiudad1.getText());
    }
    frameJuego.setEnabled(true);
    dispose();
}catch (Exception e1) {
    ModeloCSD.finalJuego(frameJuego, this);
}
}

if(e.getComponent().equals(lblCiudad2)) {
    try {
        if(Juego.viajeCorrecto.equals("Incorrecto")) {

            if(lblCiudad2.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual].
getNombreCiudad())) {
                ModeloCSD.cambioCiudadCorrecta();
            }else {

                ModeloCSD.cambioCiudadIncorrecta(lblCiudad2.getText());
            }
        }else
        if(lblCiudad2.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual+1].get
NombreCiudad())) {
            if(Juego.ordenCaptura==true) {
                ModeloCSD.reduccionTiempo(4);
                ModeloCSD.finalJuego(frameJuego, this);
            }else {
                ModeloCSD.cambioCiudadCorrecta();
            }
        }else {

            ModeloCSD.cambioCiudadIncorrecta(lblCiudad2.getText());
            }
            frameJuego.setEnabled(true);
            dispose();
        }catch (Exception e1) {
            ModeloCSD.finalJuego(frameJuego, this);
        }
    }

    if(e.getComponent().equals(lblCiudad3)) {
        try {
            if(Juego.viajeCorrecto.equals("Incorrecto")) {

```



```

        if(lblCiudad3.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual].
getNombreCiudad())) {
            ModeloCSD.cambioCiudadCorrecta();
        }else {

ModeloCSD.cambioCiudadIncorrecta(lblCiudad3.getText());
        }
    }else
if(lblCiudad3.getText().equals(Iniciacion.ciudadesCaso[Juego.ciudadActual+1].get
NombreCiudad())) {
        if(Juego.ordenCaptura==true) {
            ModeloCSD.reduccionTiempo(4);
            ModeloCSD.finalJuego(frameJuego, this);
        }else {
            ModeloCSD.cambioCiudadCorrecta();
        }
    }else {

ModeloCSD.cambioCiudadIncorrecta(lblCiudad3.getText());
        }
        frameJuego.setEnabled(true);
        dispose();
    }catch (Exception e1) {
        ModeloCSD.finalJuego(frameJuego, this);
    }
}

}
@Override
public void mouseEntered(MouseEvent e) {
    if(e.getComponent().equals(lblCiudad3)) {
        lblCiudad3.setBackground(Color.white);
        lblCiudad3.setForeground(Color.black);
    }
    if(e.getComponent().equals(lblCiudad2)) {
        lblCiudad2.setBackground(Color.white);
        lblCiudad2.setForeground(Color.black);
    }
    if(e.getComponent().equals(lblCiudad1)) {
        lblCiudad1.setBackground(Color.white);
        lblCiudad1.setForeground(Color.black);
    }
}
@Override
public void mouseExited(MouseEvent e) {
    if(e.getComponent().equals(lblCiudad3)) {
        lblCiudad3.setBackground(Color.black);
        lblCiudad3.setForeground(Color.white);
    }
}

```



```
        if(e.getComponent().equals(lblCiudad2)) {
            lblCiudad2.setBackground(Color.black);
            lblCiudad2.setForeground(Color.white);
        }
        if(e.getComponent().equals(lblCiudad1)) {
            lblCiudad1.setBackground(Color.black);
            lblCiudad1.setForeground(Color.white);
        }
    }
    @Override
    public void mousePressed(MouseEvent e) {
    }
    @Override
    public void mouseReleased(MouseEvent e) {
    }
}
```

