

### Reglas del juego

- Tal como se le dijo antes, durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- Puede consultar sus notas, el libro o apuntes; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo.
- No se resolverá ninguna duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone. Tenga presente que un supuesto no puede contradecir lo que se le está diciendo.
- Debe escribir con letra clara.

La forma de evaluar este parcial es la siguiente:

Ítem	Valor
<b>Primer punto</b>	<b>40%</b>
La interfaz dibujada contiene todos los componentes que se encuentran en el código respetando sus propiedades.	35%
La interfaz dibujada respeta las restricciones y ubicaciones de todos los componentes al incluirse en la interfaz.	65%
<b>Segundo punto</b>	<b>40%</b>
El diagrama de clases cumple con el código entregado.	30%
El diagrama de clases cumple con el patrón de diseño MVC.	10%
Se identifican los métodos que intervienen en el registro de los personajes en cada una de las clases.	20%
Se describen de forma completa los métodos que intervienen en el proceso de registro.	40%
<b>Tercer punto</b>	<b>20%</b>
Se completa el método +cambiarAnimación():void de la clase HilolImagen.	50%
Se completa el método +eventoAnimación():void de la clase Ejecutable.	50%

### Punto 1 (40%)

J.K Rowling contrató a un Ingeniero para que le desarrollara una aplicación que le permitiera almacenar alguna información básica de los personajes de su heptalogía de novelas basadas en la vida de Harry Potter. El Ingeniero se fue del país, para realizar algunos estudios de postgrado, y le dejó solamente el código de la ventana y de la clase que instancia esta ventana. Ella está muy preocupada porque no entiende ni una sola línea del archivo que él le dejó y lo ha contratado a usted para que entienda el código y haga un gráfico que muestre la ventana que generará dicho código.

A continuación se presenta el código que el Ingeniero dejó. Usted debe dibujar **todos** los componentes que están en el código de acuerdo a las propiedades y restricciones que el Ingeniero incluyó para cada uno de ellos.

Recuerde: Si tiene algún supuesto o algún comentario respecto a los componentes de la ventana, escríbalos.

```
package control;

import java.awt.event.*;
import vista.*;
import modelo.*;
```

```

public class Ejecutable {

    private static Ventana ventana;
    private static Novela novela;

    public static void main(String[] args) {
        ventana = new Ventana();
        novela = new Novela();
        ventana.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
        ventana.setTitle("Personajes de Harry Potter");
        ventana.setSize(600, 400);
        ventana.setVisible(true);
    }

    public static void eventoAnimación() {
        //Me falta este método
    }
}

package vista;
import java.awt.*;
import javax.swing.*;

public class Ventana extends JFrame {
    private JSplitPane panelDivisorio;
    private JPanel panelArriba;
    private JLabel labelTipoPersonaje;
    private ButtonGroup grupoBotones;
    private JLabel labelImagen;
    private JTextArea áreaDescripción;
    private JLabel labelDescripción;
    private JComboBox comboBoxClasificación;
    private JLabel labelClasificación;
    private JRadioButton opciónSecundario;
    private JRadioButton opciónPrincipal;
    private JTextField campoNombre;
    private JLabel labelNombre;
    private JPanel panelAbajo;
    private GridBagLayout panelArribaLayout;
    private GridBagConstraints restricciones;
    private FlowLayout panelAbajoLayout;
    private JButton botónGuardar;
    private JButton botónAnimación;

    public Ventana() {
        super();
        inicializarComponentes();
        agregarComponentes();
    }

    private void inicializarComponentes() {
        // =====//

        panelDivisorio = new JSplitPane();
        panelDivisorio.setOrientation(JSplitPane.VERTICAL_SPLIT);
        panelDivisorio.setDividerLocation(250);
        panelDivisorio.setEnabled(false);
        // =====//

        panelArriba = new JPanel();
        panelArriba.setPreferredSize(new java.awt.Dimension(250, 600));
    }
}

```

```

panelArribaLayout = new GridBagLayout();
restricciones = new GridBagConstraints();

labelNombre = new JLabel("Nombre");
campoNombre = new JTextField();
campoNombre.setPreferredSize(new java.awt.Dimension(200, 21));

labelTipoPersonaje = new JLabel("Tipo de personaje");
opciónPrincipal = new JRadioButton("Principal");
opciónPrincipal.setSelected(true);
opciónSecundario = new JRadioButton("Secundario");
grupoBotones = new ButtonGroup();
grupoBotones.add(opciónPrincipal);
grupoBotones.add(opciónSecundario);

labelClasificación = new JLabel("Clasificación del personaje");
comboBoxClasificación = new JComboBox(new String[] {
    "Alumnos de Hogwarts", "Empleados de Hogwarts (otros)",
    "Trabajadores del Ministerio de Magia", "Mortífagos",
    "Miembros de la Orden del Fénix",
    "Miembros del Ejército de Dumbledore",
    "Miembros del Club de las Eminencias", "Brigada Inquisitorial",
    "Art français", "Animagos", "Hombres Lobo" });
comboBoxClasificación.setPreferredSize(new java.awt.Dimension(200, 21));

labelDescripción = new JLabel("Descripción del personaje");
áreaDescripción = new JTextArea();
áreaDescripción.setPreferredSize(new java.awt.Dimension(350, 63));
// =====//

panelAbajo = new JPanel();
panelAbajo.setPreferredSize(new java.awt.Dimension(600, 150));

panelAbajoLayout = new FlowLayout();

labelImagen = new JLabel();
labelImagen.setIcon(new ImageIcon("HarryPotter.jpg"));
botónGuardar = new JButton("Guardar");
botónAnimación = new JButton("Animación");

// =====//
}
private void agregarComponentes() {

// =====//

panelArriba.setLayout(panelArribaLayout);

restricciones.weightx = 1;
restricciones.weighty = 0.0;
restricciones.anchor = GridBagConstraints.WEST;
Insets insets = new Insets(10, 10, 10, 10);
restricciones.insets = insets;

restricciones.gridx = 0;
restricciones.gridy = 0;
panelArriba.add(labelNombre, restricciones);

restricciones.gridx = 1;
restricciones.gridy = 0;
panelArriba.add(campoNombre, restricciones);

```

```

    restricciones.gridx = 0;
    restricciones.gridy = 1;
    panelArriba.add(labelTipoPersonaje, restricciones);

    restricciones.gridx = 1;
    restricciones.gridy = 1;
    panelArriba.add(opciónPrincipal, restricciones);

    restricciones.gridx = 2;
    restricciones.gridy = 1;
    panelArriba.add(opciónSecundario, restricciones);

    restricciones.gridx = 0;
    restricciones.gridy = 2;
    panelArriba.add(labelClasificación, restricciones);

    restricciones.gridx = 1;
    restricciones.gridy = 2;
    panelArriba.add(comboBoxClasificación, restricciones);

    restricciones.gridx = 0;
    restricciones.gridy = 3;
    panelArriba.add(labelDescripción, restricciones);

    restricciones.gridx = 1;
    restricciones.gridy = 3;
    restricciones.gridwidth = 2;
    panelArriba.add(áreaDescripción, restricciones);

    // =====//

    panelAbajo.setLayout(panelAbajoLayout);
    panelAbajo.add(labelImagen);
    panelAbajo.add(botónGuardar);
    panelAbajo.add(botónAnimación);

    // =====//

    panelDivisorio.add(panelArriba, JSplitPane.TOP);
    panelDivisorio.add(panelAbajo, JSplitPane.BOTTOM);
    this.getContentPane().add(panelDivisorio, BorderLayout.CENTER);
}

public JTextArea getÁreaDescripción() {
    return áreaDescripción;
}
public JButton getBotónGuardar() {
    return botónGuardar;
}
public JButton getBotónAnimación() {
    return botónAnimación;
}
public JTextField getCampoNombre() {
    return campoNombre;
}
public JComboBox getComboBoxClasificación() {
    return comboBoxClasificación;
}
public JRadioButton getOpciónPrincipal() {
    return opciónPrincipal;
}
}

```

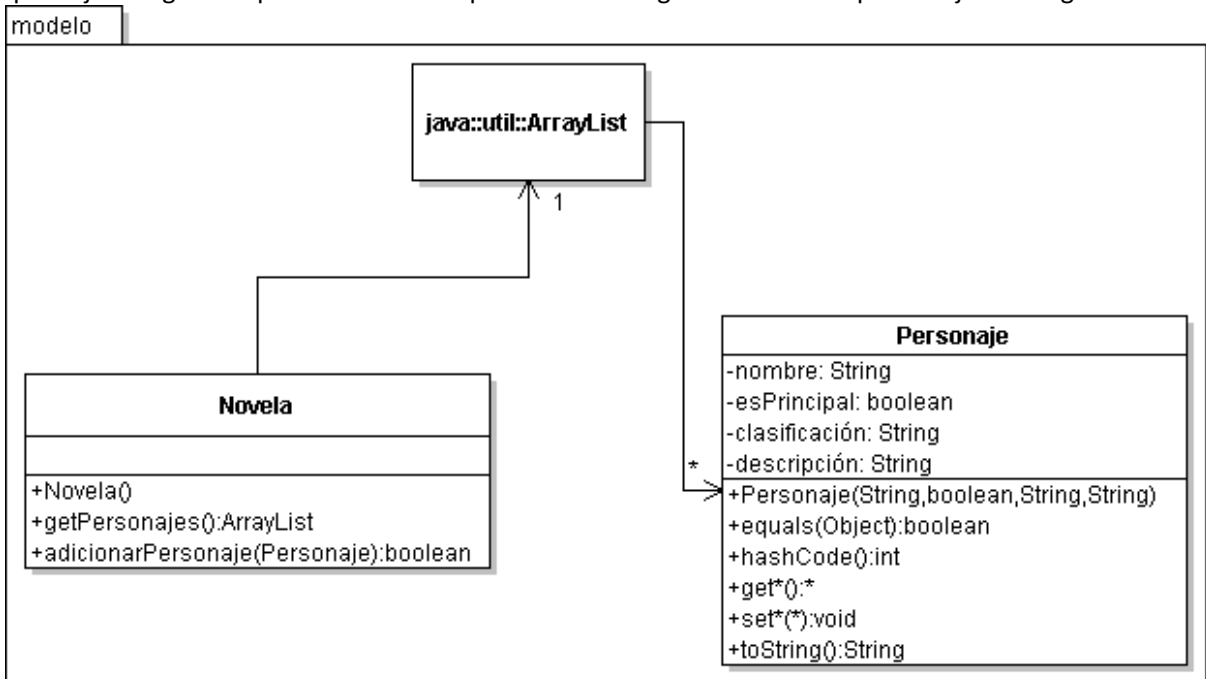
```

public JRadioButton getOpciónSecundario() {
    return opciónSecundario;
}
public void limpiar(){
    campoNombre.setText("");
    opciónPrincipal.setSelected(true);
    comboBoxClasificación.setSelectedIndex(0);
    áreaDescripción.setText("");
}
public void setImagen(String imagen) {
    this.labelImagen.setIcon(new ImageIcon(imagen));
}
}
}

```

## Punto 2 (40%)

Dentro de la documentación que recibió J.K Rowling del Ingeniero que contrató, se encontraba la parte correspondiente al modelo del diagrama de clases. La famosa escritora, contenta con su trabajo inicial, decide contratarlo y ahora le ha solicitado completar el diagrama de clases con el control y la vista que corresponden al código que dejó el Ingeniero para solucionar el problema. El diagrama de clases que él dejó es el siguiente:



La escritora quiere que dentro del diagrama de clases, incluya en la clase o clases que correspondan, y respetando MVC, él o los métodos que sean necesarios para registrar un personaje en el momento en que se presiona el botón guardar cumpliendo con los siguientes requisitos:

1. No se permiten personajes repetidos
2. Todos los atributos del personaje son obligatorios
3. Cuando se intente registrar un personaje al cual no se le haya ingresado toda la información la aplicación debe mostrar un mensaje que indique el error.
4. Cuando se adicione correctamente un personaje, la aplicación debe mostrar el mensaje exitoso y limpiar los campos para el ingreso de un nuevo personaje.

Adicionalmente, ella ha solicitado a usted que elabore la descripción de los métodos requeridos para cumplir con la funcionalidad mencionada. La descripción de los métodos debe ser **detallada**, es decir, que debe incluir paso a paso lo que se debe hacer para lograr el funcionamiento esperado dentro de cada uno de los métodos.

### **Punto 3 (20%)**

La escritora ha quedado muy contenta con su trabajo, pues gracias a la clara documentación presentada se pudo llevar a cabo la construcción de la aplicación; sin embargo, ahora ella quiere agregar algo que haga más divertida la aplicación. Para esto pretende que una vez se presione el botón de Animación, se pueda crear una especie de animación que consiste en lo siguiente:

Una vez presionado el botón de animación, se debe cambiar cada 10 segundos la imagen de Harry Potter que se encuentra en la ventana (para esto ella ha dispuesto un total de cincuenta imágenes). Cuando el botón vuelva a ser presionado debe dejar de cambiar la imagen y dejar la última que se había mostrado. Así el usuario que ingresa los datos puede disfrutar de las imágenes mientras registra personajes o dejar una imagen fija.

J.K. Rowling intentó contactarlo, pero como no pudo, recurrió a un programador que no terminó el trabajo. Ella recibió el siguiente código

```
package control;

import vista.*;
import java.util.*;

public class HiloImagen extends Thread {
    private boolean animar;
    private Ventana ventana;
    private String[] imágenes;

    public HiloImagen(Ventana ventana, String[] imágenes) {
        animar = false;
        this.ventana = ventana;
        this.imágenes = imágenes;
    }

    public void run() {

        while (true) {
            if (animar) {
                Random generador = new Random();
                ventana.setImagen(imágenes[generador.nextInt(50)]);
                ventana.repaint();
                try {
                    Thread.sleep(10000);
                } catch (InterruptedException e) {
                }
            }
        }

        public void cambiarAnimación() {
            //Me falta este método
        }
    }
}
```

Usted debe realizar el código del método +cambiarAnimación():void de la clase HiloImagen que permita iniciar la animación y parar la animación sucesivamente a medida que se presiona el botón, y realizar el código del método +eventoAnimación():void de la clase Ejecutable para capturar el evento del botón de animación y cambiar el estado de la misma en la medida que es presionado.