UD03 – PE – Practica Escrita (Supuesto Práctico)

Ejercicio 1 – Aplicación de Lista de Contactos (3,5 puntos)

Se quiere crear una pequeña aplicación en **Java** que permita gestionar una **lista de contactos** utilizando **Swing**. Cada contacto tiene un **nombre**, un **teléfono** y un **id**. La información de los contactos se mostrará en un componente **JTextArea**.

Cuando el usuario pulse el botón **"Agregar"**, se creará un nuevo contacto y se mostrará su información en el área de texto.
Cuando se pulse el botón **"Borrar"**, se eliminará todo el contenido del área de texto (es decir, la lista de contactos visualizada). **Se muestra imagen del ejercicio**.

a) Clase Contacto (1,5 puntos)

Crea la clase **Contacto** que contenga los siguientes elementos:

- Atributos: id, nombre, telefono.
- Tendrás que crear un id distinto cada vez que se crea un contacto.
- Constructor que reciba los tres valores.
- Métodos **get** y **set** para el nombre y teléfono.
- Método **getId()** para obtener el identificador del contacto.
- Método **mostrarInfo(id)** que devuelva un texto con toda la información del contacto en formato:
  ID: 1 | Nombre: Ana | Teléfono: 600123456

b) Botón "Agregar" (1,5 puntos)

Crea el manejador de evento del botón **"Agregar"** que:

- Tome el nombre y el teléfono escritos en los campos de texto.
- Cree un objeto de la clase **Contacto** asignándole un id automático.
- Muestre la información del contacto en el **JTextArea** (uno por línea).
- Limpie los campos de texto después de agregar.

c) Botón "Borrar" (0,5 puntos)

Crea el manejador de evento del botón **"Borrar"** que elimine el contenido del **JTextArea**, dejando la lista vacía.

Responder la pregunta dentro del proyecto Java entregado, en el paquete Pregunta01.

Ejercicio 2: Selección de Método de Pago y Bolsa (3 puntos)

Se quiere crear una aplicación gráfica en **Java** que permita determinar el **método de pago** y si el cliente desea **bolsa o no**.

Para ello se tendrán varios componentes **JRadioButton**:

- Para el **método de pago**: “Efectivo”, “Tarjeta”, “Transferencia” y “Bizum”.
- Para la **opción de bolsa**: “Sí” y “No”.

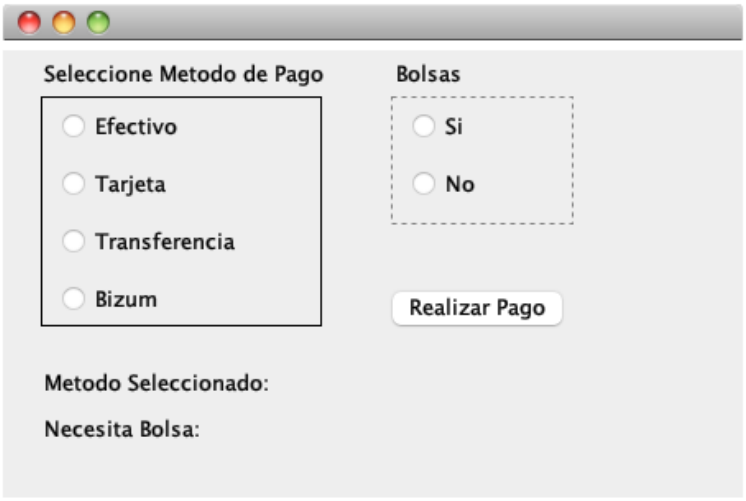
Se crearán dos arrays de tipo String:

- arrTxtOpcionesPago → contendrá las opciones de pago.
- arrTxtOpcionesBolsa → contendrá las opciones para la bolsa.

Habrá dos componentes **JLabel** que mostrarán:

- El método de pago seleccionado.
- Si el cliente desea bolsa o no.

Además, habrá un botón **“Realizar Pago”**, que al hacer clic mostrará en los JLabel las opciones seleccionadas por el cliente. Se adjunta imagen del ejercicio.



a) Agrupación de botones y creación de arrays (1 puntos)

Crea el código necesario para:

- Agrupar los botones de radio de los métodos de pago y de las opciones de bolsa en sus respectivos **ButtonGroup**.
- Crear los arrays arrTxtOpcionesPago y arrTxtOpcionesBolsa con las opciones correspondientes.

b) Evento del botón “Realizar Pago” (2,0 puntos)

Crea el **manejador de evento** del botón **“Realizar Pago”** que:

- Obtenga el texto de los **JRadioButton** seleccionados (tanto del método de pago como de la bolsa).
- Muestre los textos seleccionados en los **JLabel** correspondientes (modificando su texto).

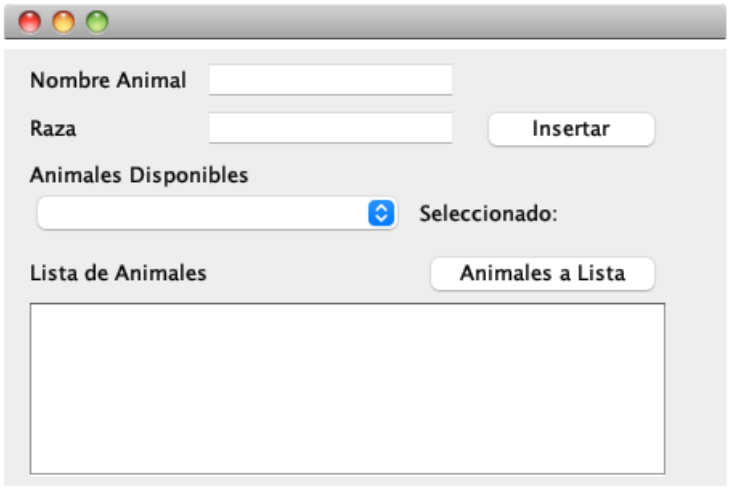
Responder la pregunta dentro del proyecto Java entregado, en el paquete Pregunta02.

Ejercicio 3: Lista de Animales y Razas (3,5 puntos)

Se quiere crear una pequeña aplicación en **Java** que muestre una **lista de animales y sus razas**, utilizando un **JComboBox**, un **JList** y varios botones. Se muestra imagen del ejercicio.

La aplicación debe permitir:

- Mostrar animales iniciales (nombre y raza) en un JComboBox.
- Insertar nuevos animales desde un formulario.
- Mostrar el animal seleccionado.
- Pasar todos los animales del JComboBox a una lista (JList) al pulsar un botón.



a) Clase Animal y carga inicial (1,5 puntos)

Crea la clase **Animal** con los siguientes elementos:

- Atributos: nombre y raza.
- Completa la clase Animal con el código necesario.
- Constructor que reciba ambos valores.
- Métodos **get** y **set** para los dos atributos.
- Método **toString()** que devuelva un texto en el formato:
  "Perro - Pastor Alemán"

Crea también dos arrays con los datos iniciales:

String[] petNames = {"Perro", "Gato", "Conejo", "Pájaro"};
String[] petRaza = {"Pastor Alemán", "Siames", "Belier", "Canario"};

y agrégalos al JComboBox como animales disponibles.

b) Botón "Insertar" (1,0 punto)

Crea el **manejador de evento** del botón **"Insertar"** que:

- Tome el nombre y la raza escritos en los campos de texto.
- Cree un objeto Animal con esos datos.
- Lo agregue al JComboBox (comboBox) de animales disponibles.
- Limpie los campos de texto después de insertarlo.

c) Mostrar el animal seleccionado (0,5 puntos)

Crea el manejador de evento **ItemChange** del JComboBox (comboBox) que muestre el **nombre** del animal seleccionado en un componente JTextField llamado **lblSelectedAnimal**.

d) Botón "Animales a Lista" (0,5 puntos)

Crea el manejador de evento del botón **"Animales a Lista"** que agregue todos los animales disponibles en el JComboBox a la JList **listAnimales**.
Cada elemento de la lista debe mostrarse como texto con el **nombre y raza del animal**.

Responder la pregunta dentro del proyecto Java entregado, en el paquete Pregunta03.

Nota: (para todos los ejercicios)
Responder exactamente a lo que se pregunta y como se pregunta.
No usar IA. No usar Internet.
Ejercicio obligatorio e individual de cada alumno.
No se puede usar ningún material no suministrado por el profesor.
El código debe de estar libre de errores.
Realizar comentarios y documentar tu solución.
Formatear el código para lectura clara e indentación correcta que facilite la lectura del código.
Generar y justificar tu solución aportada.
Prohibido el uso del móvil en el aula y durante el examen.
Permanecer en silencio durante el examen (no se habla con el compañero).
Entregar dicho ejercicio en la plataforma siguiendo instrucciones del profesor.
Leer atentamente todas las preguntas.