



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



ALUMNO: Meza Vargas Brandon David.

PRÁCTICA: Práctica No. 2.

TEMA: Arreglos de objetos.

OPCIÓN: Opción 3, mosaico de imágenes.

FECHA: 16-oct-2020

GRUPO: 2CM1

MATERIA: Programación Orientada a Objetos

INTRODUCCIÓN

Los arreglos de objetos nos permiten almacenar un conjunto de datos del mismo tipo. Con un único nombre definimos nuestro arreglo y por medio de un subíndice hacemos referencia a cada elemento del mismo.

Los arreglos de objetos en Java se numeran desde el elemento cero, que sería el primer elemento, hasta el tamaño-1 que sería el último elemento. Es decir, si tenemos un array de diez elementos, el primer elemento sería el cero y el último elemento sería el nueve.

La ventaja principal de un arreglo es que organiza los datos de tal manera que puede ser manipulado fácilmente. Por ejemplo, si tiene un array que contiene los ingresos de un grupo seleccionado de hogares, es fácil calcular el ingreso promedio haciendo un ciclo a través del array. Además, los arreglos organizan los datos de tal manera que se pueden ordenar fácilmente.

Los objetos en Java nos ahorran tiempo, código y funcionalidad cuando queremos manejar una gran cantidad de elementos en un mismo programa.

DESARROLLO

Para esta práctica se escogió la opción número 3, en la cual tenemos que colocar 40 imágenes en botones, es decir, tenemos que hacer un arreglo de 40 botones en donde pondremos una imagen a cada botón, en este caso, cada botón tendrá una imagen de un perrito, al momento de hacer clic en una imagen esta tendrá que aparecer en una etiqueta.

Para resolver este problema, usaremos un Applet, primeramente importamos las librerías necesarias para nuestro applet, posteriormente creamos nuestro método `init()` el cual se encargara de inicializar nuestro applet.

Para colocar los botones en nuestro applet hacemos uso de un panel, además de un `GridLayout` de 4 por 10 para que se acomoden de una mejor manera nuestros botones.

Usamos un ciclo `for` para ir colocando nuestros botones uno por uno hasta que llegue al elemento 39, es decir, nuestro último botón, la línea de código que se encarga de colocar las imágenes en el botón la podemos ver en la imagen 1;

```
elementos[i] = new JButton(new ImageIcon(i+".jpg"));
```

Imagen 1. Línea de código que agrega imágenes a los botones

En donde la letra `i` representa el índice donde se ira colocando cada imagen, como nuestra variable `i` va del 0 hasta el 39, las imágenes fueron guardadas de esta forma, es decir; 0.jpg, 1.jpg, 2.jpg, etc. Algo muy importante aquí es que las imágenes de nuestro applet deben estar guardadas en la misma carpeta que nuestro archivo java,

de otra forma se generan los botones pero sin imágenes por qué no se encontraron. Dentro de este ciclo también añadimos los botones al panel y su respectivo ActionListener, posteriormente añadimos dos etiquetas más, una con el texto tu selección y la otra etiqueta es donde se mostrará la imagen a la que hagamos clic.

Nuestro método init() lo vemos en la imagen 2 que se muestra a continuación;

```
public void init(){
    c=getContentPane();
    c.setLayout(new GridLayout(4,10));
    elementos = new JButton[40];
    for(i=0;i<40;i++){
        elementos[i] = new JButton(new ImageIcon(i+".jpg"));
        c.add(elementos[i]);
        elementos[i].addActionListener(this);
    }

    l=new JLabel("Tu seleccion: ");
    c.add(l);
    selec = new JLabel();
    c.add(selec);
}
```

Imagen 2. Método init().

Posteriormente dentro del método actionPerformed(), haremos que cada vez que se haga clic sobre un botón extraiga la imagen con ayuda de getIcon() y la coloque en nuestra etiqueta con un setIcon(), esto lo vemos en la imagen 3;

```
public void actionPerformed (ActionEvent e){
    JButton b=(JButton)e.getSource();
    selec.setIcon(b.getIcon());
}
```

Imagen 3. Método actionPerformed.

Finalmente en nuestro main creamos un objeto de tipo mosaico para así manda a llamar a init() y start(), además de crear un objeto tipo Frame para visualizar nuestro applet, esta parte del código lo vemos en la imagen 3;

```

public static void main(String[] args) {
    mosaico m=new mosaico();
    m.init(); m.start();
    Frame f=new Frame("Mosaico de imágenes");
    f.add("Center",m);
    f.setSize(900,300);

    f.setVisible(true);
}

```

Imagen 4. Nuestro main

Teniendo como resultado lo siguiente:

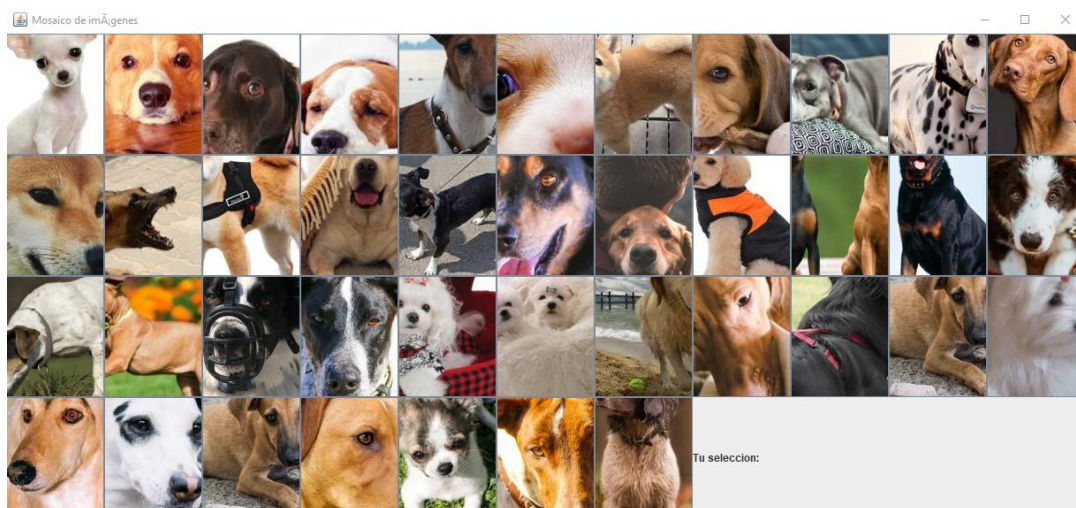


Imagen 5. Ejecución del applet.

Y, si seleccionamos una imagen nos lo muestra en la etiqueta como lo vemos en la imagen 6;

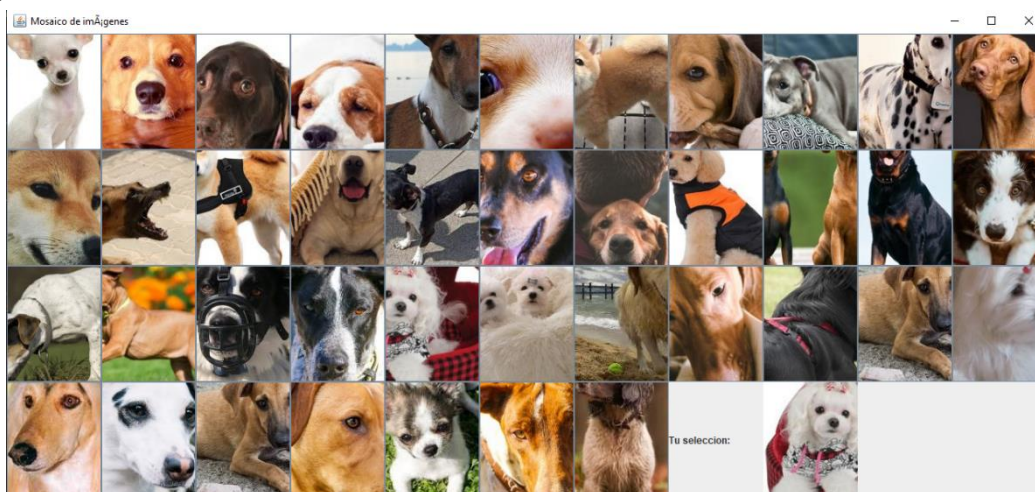


Imagen 6

CONCLUSIONES

Como vimos en la introducción y desarrollo de este reporte, los arreglos de objetos son una forma sencilla de manipular gran cantidad de información, en este caso, botones, además que permiten que nuestra Interfaz Gráfica de Usuario se vea agradable, ordenada y sea fácil de manipular, de igual forma es muy importante considerar un layout adecuado al momento de manejar arreglos o elementos en nuestra GUI para que estos tengan una distribución adecuada en ella.

Con esta práctica adquirí conocimientos sobre arreglos de objetos y como implementarlos en una GUI, además de como colocar una imagen en un botón.