



## **Ejercicio de nivel 5 cupiMuseo Consideraciones adicionales de diseño**

### **Consejos prácticos para la construcción de la interfaz gráfica de la aplicación**

1. Recomendación para la distribución de los paneles en la ventana principal.



Cada cuadro en rojo es un panel como se describe a continuación:

- ✓ Panel con la imagen del banner.
- ✓ Panel con la información de una obra de arte.
- ✓ Panel con las opciones de calificación.
- ✓ Panel con las opciones.

2. Para incluir una imagen en un panel o una ventana, se puede utilizar un JLabel de la siguiente manera:

```
JLabel lblImagen = new JLabel( );  
ImageIcon icono = new ImageIcon( rutaImagen );  
lblImagen.setIcon( icono );
```

Donde rutaImagen es la ruta donde se encuentra ubicada la imagen. Por ejemplo, la imagen del banner se encuentra en `"/data/imagenes/banner.png"`.

3. Para el manejo de distintos tipos de fuentes en JLabel se usa la clase Font de java.awt. Para crear una fuente personalizada y asignarla a un JLabel se pueden seguir los siguientes pasos:

```
JLabel lblEjemplo = new JLabel( textoLabel );  
Font fuenteNueva = new Font("Arial", Font.BOLD, 20);  
lblEjemplo.setFont(fuenteNueva);
```

En este ejemplo, se asignó una fuente Arial de tamaño 20 a la etiqueta con nombre lblEjemplo. El constructor de la clase Font recibe 3 parámetros: el tipo de la fuente, el estilo de la fuente y el tamaño de la fuente.

A continuación se muestran dos ejemplos de cómo crear una fuente personalizada:

- a) Una etiqueta creada con los siguientes comandos:

```
JLabel lblEjemplo = new JLabel( "Texto ejemplo" );  
Font fuenteNueva = new Font("Arial", Font.BOLD, 15);  
lblEjemplo.setFont(fuenteNueva);
```

Se verá así:

**Texto ejemplo**

- b) La misma etiqueta pero con formato distinto de fuente tiene los siguientes comandos:

```
JLabel lblEjemplo = new JLabel( "Texto ejemplo" );  
Font fuenteNueva = new Font("Comic Sans MS", Font.ITALIC, 30);  
lblEjemplo.setFont(fuenteNueva);
```

Se verá así:

*Texto ejemplo*

Si desea consultar la documentación de la clase Font, para conocer los estilos y tipos de letra disponible, lo puede hacer en el siguiente enlace:

<http://docs.oracle.com/javase/6/docs/api/java/awt/Font.html>



4. En Mac se debe tener en cuenta que para poder pintar un botón deben activar las siguientes propiedades del mismo:
  - ✓ `botonX.setOpaque( true );`
  - ✓ `botonX.setBorderPainted( true );`
  - ✓ `botonX.setBorder( BorderFactory.createLineBorder( Color.white ) );`

### Explicación de algunos aspectos del mundo

1. El constructor de la clase principal, `CupiMuseo`, se encarga de crear todos los elementos cargando la información de un archivo.
2. La clase `CupiMuseo` maneja el método `darObraActual`, que devuelve la obra que está actualmente seleccionada.
3. La clase `CupiMuseo` maneja los métodos `darObraAnterior`, `darObraSiguiente`, `darPrimeraObra` y `darUltimaObra`, que se desplazan entre las obras y devuelven la obra requerida en cada caso.

