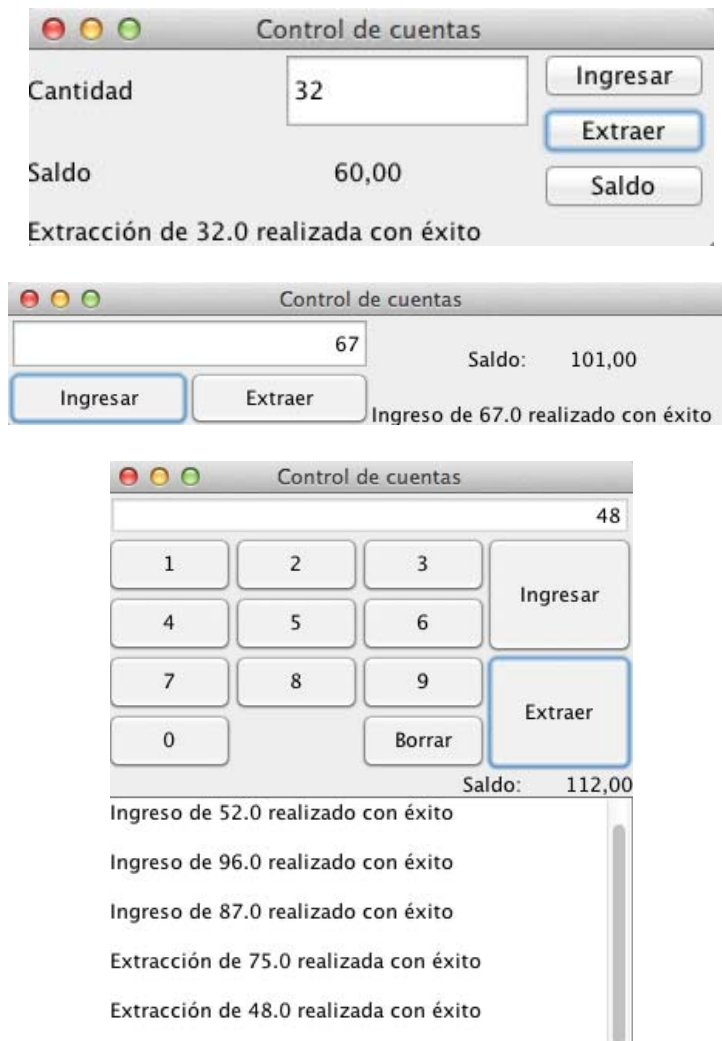


**Ejercicio 1. (proyecto prCuentasGUI)**

Se desea desarrollar una aplicación gráfica que permita manipular una cuenta bancaria (el modelo es la clase Cuenta definida en las transparencias del tema de Interfaces Gráficas de Usuario). La aplicación tendrá tres posibles vistas:



Las clases que implementan los diferentes paneles gráficos son PanelCuenta1, PanelCuenta2 y PanelCuenta3, respectivamente. Las tres implementan la interfaz VistaCuenta.

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

public interface VistaCuenta {
    String INGRESO = "INGRESO";
    String GASTO = "GASTO";
    String SALDO = "SALDO";
    /** Obtenemos la cantidad a ingresar o extraer
     * @return double con la cantidad a ingresar
     */
    double obtenerCantidad();
    /** Establecemos el saldo de la cuenta.
     * @param saldo double con el saldo de la cuenta.
     */
    void saldo(double saldo);
    /** Mostramos un mensaje de información.
```

```
        *   @param msg String con el mensaje a mostrar.  
        */  
void mensaje(String msg);  
/** Pasamos el controlador.  
    */  
void controlador(ActionListener ctr);  
}
```

Para controlar los botones de ingresar, extraer y saldo se debe implementar la clase `CtrCuenta`.

La clase `CuentaDemo` será la clase principal de la aplicación. Debemos poder poner o quitar comentarios al panel que queremos utilizar y el resto debe funcionar.

```
VistaCuenta vistaCuenta = new PanelCuenta1();  
// VistaCuenta vistaCuenta = new PanelCuenta2();  
//VistaCuenta vistaCuenta = new PanelCuenta3();
```

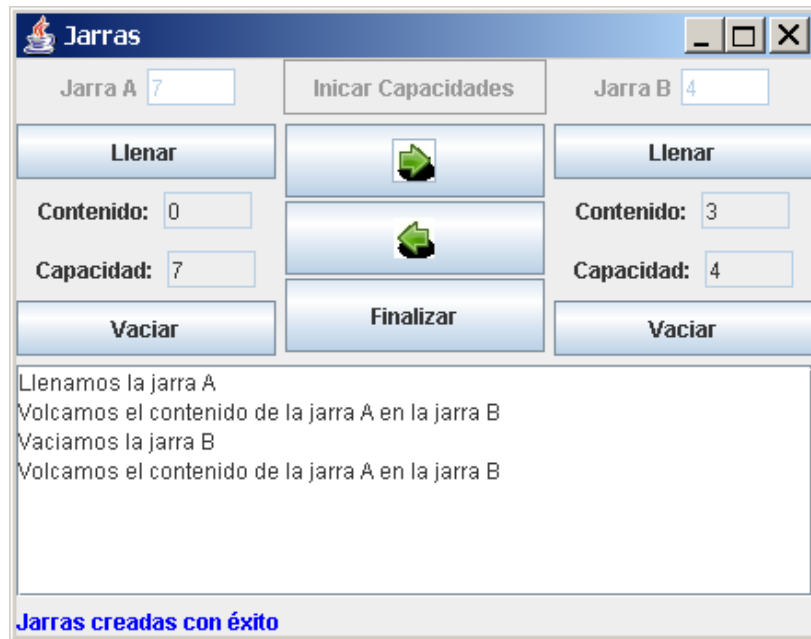
Observa que la segunda y la tercera versión no proporcionan un botón para consultar el saldo. Su valor se actualiza tras ingresar o extraer.

En la tercera versión, el campo de texto en el que introducir la cantidad es no editable, forzando a que la cantidad se introduzca utilizando la botonera.

La interfaz `VistaCuenta` y las clases `Cuenta`, `PanelCuenta1`, `PanelCuenta2`, `PanelCuenta3` y `CuentaDemo` se proporcionan, por lo que el alumno debe desarrollar la clase `CtrCuenta`.

**Ejercicio 2. (proyecto prJarrasGUI)**

Se desea desarrollar una aplicación gráfica que permita manipular dos jarras. La aplicación tendrá una interfaz gráfica como la de la figura.



La clase que implementa el panel gráfico es `PanelJarras` que implementa la interfaz `VistaJarras`. La clase `ControladorJarras` será la clase controladora a implementar y `AplicacionJarras` la clase principal de la aplicación. El modelo es la clase `Jarra`.

La interfaz `VistaJarras` y las clases `Jarra`, `PanelJarras` y `AplicacionJarras` se proporcionan. También se proporcionan las imágenes de las flechas verdes.

NOTA: Mientras las jarras no hayan sido correctamente inicializadas, no tiene sentido hacer trasvases sobre ellas. Por eso la aplicación tiene dos modos. Modo de inicialización de jarras en las que sólo es posible actuar sobre el botón de iniciar capacidades, y modo trasvase en las que ya son posibles los trasvases pero no la inicialización. El botón `finalizar` nos lleva de nuevo al modo de inicialización de jarras.

```

public interface VistaJarras {
    public static final String INICIAR = "INICIAR";
    public static final String LLENAR_A = "LLENAR_A";
    public static final String LLENAR_B = "LLENAR_B";
    public static final String VACIAR_A = "VACIAR_A";
    public static final String VACIAR_B = "VACIAR_B";
    public static final String VOLCAR_A_EN_B = "VOLCAR_A_EN_B";
    public static final String VOLCAR_B_EN_A = "VOLCAR_B_EN_A";
    public static final String FINALIZAR = "FINALIZAR";
    /**
     * Pasamos el controlador.
     */
    public void controlador(ActionListener ctr);
    /**
     * Obtenemos la capacidad inicial de la jarra A.
     * @return int con la capacidad inicial de la jarra A.
     */
    public int capacidadInicialA();
    /**
     * Obtenemos la capacidad inicial de la jarra B
     * @return int con la capacidad inicial de la jarra B.
     */
    public int capacidadInicialB();
    /**
     * Mostramos un mensaje de error.
     * @param mensaje
     *      String con el mensaje a mostrar.
     */
    public void error(String mensaje);
    /**
     * Mostramos un mensaje de información.
     * @param mensaje
     *      String con el mensaje a mostrar.
     */
    public void ok(String mensaje);
    /**
     * Habilitamos o deshabilitamos el modo inicialización de jarras o trasvases
     * @param b
     *      true para habilitar el modo inicialización; false para el modo trasvases
     */
    public void habilitarInicio(boolean b);
    /**
     * Establecemos la capacidad de la jarra A.
     * @param c
     *      int con la capacidad de la jarra A
     */
    public void capacidadA(int c);
    /**
     * Establecemos la capacidad de la jarra B.
     * @param c
     *      int con la capacidad de la jarra B.
     */
    public void capacidadB(int c);
    /**
     * Establecemos el contenido de la jarra A.
     * @param c
     *      int con el contenido de la jarra A.
     */
    public void contenidoA(int c);
    /**
     * Establecemos el contenido de la jarra B.
     * @param c
     *      int con el contenido de la jarra B.
     */
    public void contenidoB(int c);
    /**
     * Añadimos un mensaje al histórico.
     * @param mensaje
     *      String con el mensaje a añadir.
     */
    public void agregarAHistorico(String mensaje);
    /**
     * Limpiamos el histórico.
     */
    public void limpiar();
}

```