

Interfaces

Ejercicio 0

Crea una ventana (Frame) utilizando el Window Builder. Utiliza la interfaz de diseño para añadirla un **Absolute Layout**. A este layout, añádele un **JButton** y un **JTextField**. Inicialmente, el **JTextField** tiene que mostrar el texto “¡Hey!”. Cuando pulsemos el botón, se disparará un **evento** que cambiará el texto “¡Hey!” por el texto “¡Hola mundo!”, y viceversa.

Solución: Ejercicio0.java

Ejercicio 1

Crea una ventana (Frame) utilizando el Window Builder. Utiliza la interfaz de diseño para añadirla un **Absolute Layout**. La ventana tiene que tener dos **JTextField** para introducir el usuario y el password; dos **JLabel** para etiquetar los campos de texto; y un **JButton**.

Cuando el botón es pulsado, se cogerá el valor de los **JTextField**. Si ambos valores están rellenos y son “admin”, muestra un mensaje pop-up con el texto “Bienvenido”. En cualquier otro caso, muestra un mensaje pop-up con el texto “User o password desconocido”

El pop-up se hace con la instrucción **JOptionPane.showMessageDialog (...)**. Consulta en Google ejemplos de cómo se utiliza.

Solución: Ejercicio1.java

Ejercicio 2

Crea una ventana (Frame) utilizando el Window Builder. Utiliza la interfaz de diseño para añadirla un **Absolute Layout**. La ventana tiene que preguntar al usuario por el mejor videojuego. Se le ofrecerán cuatro opciones (las que quieras), cada una en un **JRadioButton** diferente. Cada vez que el usuario escoja una, debe de mostrarse en un **JTextField** y se deben de deseleccionar todos los demás **JRadioButton**.

Solución: Ejercicio2.java

Ejercicio 3

Crea una ventana. Esta ventana tiene un formulario de registro de usuarios con unos campos para introducir el nombre, apellido 1, apellido 2, DNI/NIE y edad. Un **JComboBox** para dar la opción de escoger DNI o NIE. Si el usuario escoge DNI, se tiene que comprobar que en el campo hay tiene de verdad un DNI al darle al botón 'Aceptar'. Si no, se mostrará un error. Si todo está bien, muestra en un campo de texto los datos introducidos. Otro botón 'Limpiar' borrará todos los campos.

Solución: Ejercicio3. java

Ejercicio 4

Carrusel de fotos. Crea una ventana que muestre tan solo una foto de una colección de seis. Añade cuatro botones: un botón muestra la primera foto; otro la anterior; otro la siguiente; otro la última foto de la lista. Las fotos las guardas en un Array de **ImageIcon** (mira el Ejercicio 2) ¡Cuidado con el tamaño del Array al recorrerlo! Deshabilita los botones cuando el usuario no deba pulsarlos.

Ayuda: Para que la imagen aparezca a escala del JPanel, hay que hacer esto:

```
ImageIcon imageIcon1 = new ImageIcon(this.getClass().getResource("Imagen1.jpg"));
Image image = imageIcon1.getImage().getScaledInstance(JPanelImage.getWidth(), JPanelImage.getHeight(), Image.SCALE_SMOOTH);
imageIcon1.setImage(image);
```

Y ahora el ImageIcon ya está listo para meterlo en el Array.

Solución: Ejercicio4. java

Ejercicio 5

Crea una Agenda Visual. El usuario debe poder introducir el nombre, apellidos, DNI y teléfono de un contacto. A continuación, al darle a un botón 'Añadir' el contacto se introduce en una **JTable**. Deben de poder añadirse máximo 5 contactos.

Antes de añadir un contacto, se hacen las siguientes validaciones:

- Nombre: Sólo permite caracteres en mayúscula o minúscula, mínimo 1, máximo 20
- Apellidos: Sólo permite mayúsculas o minúscula, mínimo 1, máximo 30
- DNI: Sólo permite ocho números del 0 al 9, y una letra mayúscula o minúscula
- Teléfono: Sólo permite nueve números del 0 al 9

Nota: Los **JTable** son un poco raros. Para poder ‘llenar’ un **JTable** necesitas usar un Array que contenga Arrays de Strings. El primer array [] son las filas, el segundo array [] [] son las columnas de la fila. Ejemplo para una tabla de dos filas y tres columnas.

```
String datosTabla [][]={ {"101","Amit","670000"}, {"102","Jai","780000"} };
```

Para añadir una fila nueva a la tabla hay que hacer trampa. Primero, usas **remove ()** en el **JPanel** que contiene la tabla para quitarla. Luego, vuelves a hacer la tabla poniéndola el array con los datos nuevos. Finalmente, añades la tabla de vuelta al **JPanel** y le das la orden de repintado con **updateUI ()**.

Solución: Ejercicio5. java

Ejercicio 6

Vamos a manejarnos con varios paneles en una sola ventana. Crea una interfaz que tenga tres paneles. Cada panel es de un color diferente: rojo, azul y verde. Inicialmente, todos los paneles están ocultos, excepto el primero.

- Cuando se pulsa el botón de Panel 1, se muestra el panel rojo y se ocultan los demás.
- Cuando se pulsa el botón de Panel 2, se muestra el panel azul y se ocultan los demás.
- Cuando se pulsa el botón de Panel 3, se muestra el panel verde y se ocultan los demás.

Nota: Este ‘truco’ permite diseñar interfaces diferentes y darle al usuario la sensación de que está cambiando de ventana.

Solución: Ejercicio6. java