

Ejercicios de la UD09



1. **Cuestiones generales**
2. **Ejercicios**
3. **Actividades**
4. **Más ejercicios (Sin `SceneBuilder`)**
5. **Fuentes de información**

1. Cuestiones generales

Implementa las aplicaciones gráficas siguiendo las especificaciones de diseño y funcionalidad descritas en cada ejercicio. Ten en cuenta lo siguiente:

- Los aspectos del diseño que no se describan quedan a tu elección. En todo caso, intenta que la interfaz sea sencilla de utilizar para el usuario.
- Renombra todos los componentes para que se entienda su uso. En lugar de `TextField1`, `Button1` y `Label1` es mejor utilizar abreviaturas: `txtEdad`, `btnSuma` y `lblRes`.
- Valida la información introducida por el usuario y avísale en caso de error.
- Maneja las posibles excepciones que puedan producirse.
- Puedes implementar clases adicionales si lo consideras oportuno.

2. Ejercicios

1. Crea con JavaFX la siguiente pantalla.



The image shows a JavaFX window with a dark gray background. Inside the window is a white rectangular area containing a simple calculator interface. The interface consists of four labels and three text input fields. The labels are 'Operando 1', 'Operando 2', 'Sumar', and 'Resultado'. The 'Operando 1' and 'Operando 2' labels are positioned to the left of their respective input fields. The 'Sumar' label is centered above a button. The 'Resultado' label is positioned to the left of its input field. The button has a light gray gradient and the text 'Sumar' in the center.

Cuando el usuario escriba números (controlarlo con excepciones) y pulse en sumar, debe mostrar en el `txtResultado` el resultado final.

2. Crea con `JavaFX` la siguiente pantalla.

Cuando el usuario escriba números (controlarlo con excepciones) y según que operación elija, deberá hacer la operación y mostrarlo en el txtResultado. Tener en cuenta que la división, debemos controlar que el segundo operando no debe ser 0.

3. Crea la siguiente pantalla con `JavaFX`.

Nombre	Apellidos	Edad
Tabla sin contenido		

Al pulsar en el botón de agregar persona, creara un objeto `Persona` y lo insertara en la tabla.

4. Dado el ejercicio anterior, agrega lo siguiente.

The image shows a web application interface for managing a list of people. On the left, there are three input fields labeled "Nombre", "Apellidos", and "Edad". Below these fields is a button labeled "Agregar persona". To the right of the input fields is a table with three columns: "Nombre", "Apellidos", and "Edad". The table is currently empty, displaying the text "Tabla sin contenido". Below the table, there are two buttons: "Modificar" and "Eliminar".

Al seleccionar en un registro, mostrara los datos en los txt.

Al pulsar el botón Modificar, el registro seleccionado se modificara con los datos que hay en los txt (editando el objeto).

Al pulsar el botón Eliminar, el registro seleccionado se eliminara de la tabla.

5. Dado el ejercicio anterior, modifica la ventana.

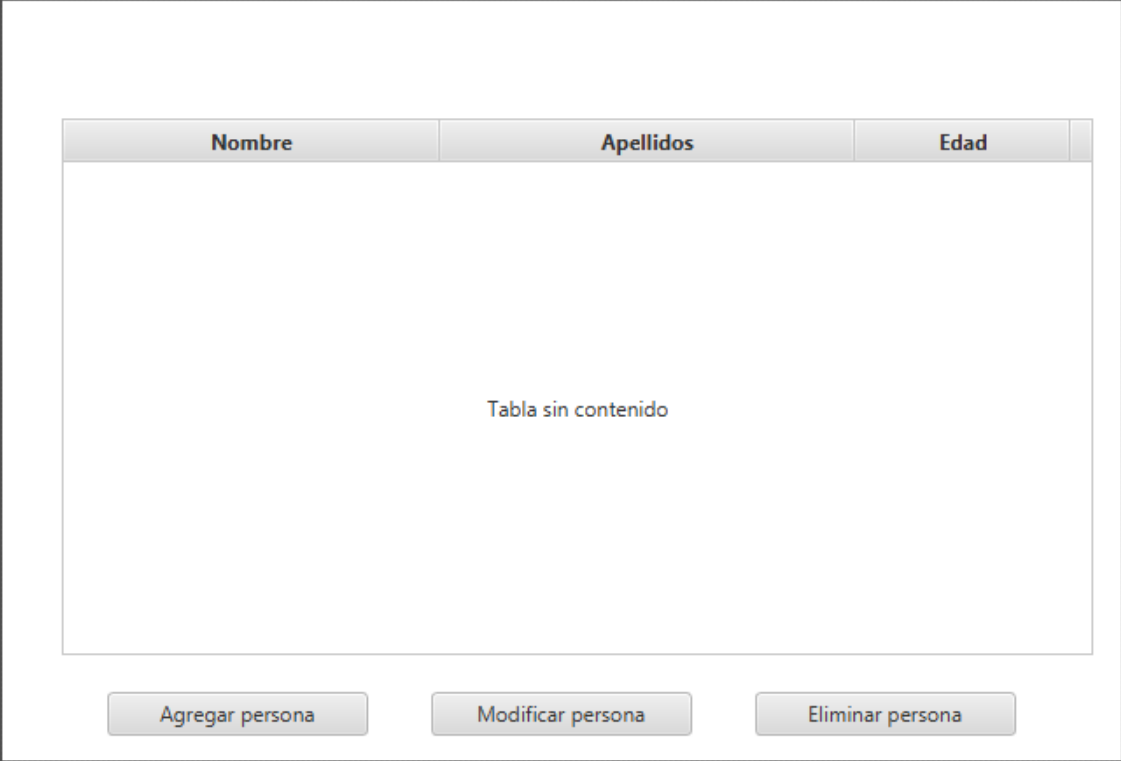
A screenshot of a web application interface. It features a table with three columns: 'Nombre', 'Apellidos', and 'Edad'. The table is currently empty, displaying the text 'Tabla sin contenido' in the center. Below the table, there is a button labeled 'Agregar persona'.

Al pulsar el botón de agregar Persona, se mostrara la siguiente ventana.

A screenshot of a form for adding a new person. The form contains three input fields labeled 'Nombre', 'Apellidos', and 'Edad'. Below the input fields, there are two buttons: 'Guardar' and 'Salir'.

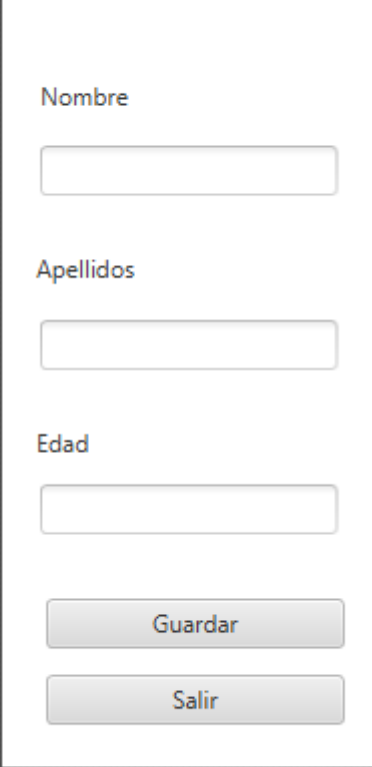
Cuando rellenamos los datos, cerrara la ventana y añadirá la persona a la tabla.

6. Dado el ejercicio anterior, modifica la ventana.



Nombre	Apellidos	Edad
Tabla sin contenido		

Al pulsar el botón de agregar Persona, se mostrara la siguiente ventana.



Nombre

Apellidos

Edad

Guardar

Salir

Cuando rellenamos los datos, cerrara la ventana y añadirá la persona a la tabla.

Cuando pulsemos modificar, se abrirá la ventana anterior, pero con los datos cargados y al guardar editara el objeto seleccionado.

Cuando pulsemos eliminar, eliminar el objeto de la tabla.

7. Dado el ejercicio anterior, modifica la ventana (se mantiene el dialogo de persona).

The screenshot shows a web application interface for managing a list of people. At the top, there is a search filter labeled "Filtrar por nombre:" followed by a text input field. Below this is a table with three columns: "Nombre", "Apellidos", and "Edad". The table is currently empty, displaying the message "Tabla sin contenido". At the bottom of the interface, there are three buttons: "Agregar persona", "Modificar persona", and "Eliminar persona".

Cuando escribamos en el txt de arriba, filtrara los nombre de las personas que contengan el texto que tu indiques.

3. Actividades

1. ParOImpar

Aplicación gráfica que permita introducir un número entero y luego saber si dicho número es par o impar. Utiliza un `TextField` para introducir el valor, un `Button` con el texto “¿par o impar?” y un `Label` para mostrar “PAR” o “IMPAR” según el caso.

2. Minicalculadora1

Aplicación gráfica que permita introducir dos números reales y calcular el resultado de su suma, resta, multiplicación o división. Utiliza un `Button` distinto para cada operación y un único `Label` para mostrar el resultado.

3. Minicalculadora2

Aplicación gráfica que permite introducir dos números enteros (A y B) y permita realizar tres cálculos distintos: suma de A y B, producto de A x B y exponencial A^B (A elevado a B). Utiliza tres `RadioButton` (uno para cada cálculo) y un botón “¡Calcula!”.

4. Factorial

Aplicación de uso de JavaFX con FXML y controlador para calcular el factorial de un número.

5. ValidarLetraNIF

Aplicación gráfica que permita introducir un NIF (8 números y una letra) e indique si la letra de dicho NIF es válida.

6. DatosDeRol

Aplicación gráfica que permita al usuario simular que lanza un dado de juegos de rol. Podrá elegir entre dado de 6 caras (de 1 a 6), dado de 8 caras (de 1 a 8), dado de 10 caras (de 1 a 10), dado de 12 caras (de 1 a 12) y dado de 20 caras (de 1 a 20). Utiliza un botón distinto para cada tipo de dado. Muestra en cada botón una imagen de cada dado. Puedes encontrar las imágenes fácilmente haciendo una búsqueda en Internet.

7. InicioDeSesion

Aplicación gráfica que simule una ventana de inicio de sesión y registro de usuarios. El usuario podrá introducir su nombre de usuario, contraseña (`PasswordField`) y hacer click en un botón de “Iniciar sesión”. Muestra el resultado del intento de inicio de sesión en un `Label`.

Los usuarios registrados y sus contraseñas estarán en el archivo ‘users.txt’. Crea unos pocos usuarios de ejemplo para probar la aplicación. No está permitido utilizar espacios ni en los nombres de usuario ni en las contraseñas.

Extra: Añade un botón “Crear usuario” que registre un nuevo usuario.

8. BurgerMenApp

Implementa una aplicación gráfica que simule una aplicación de pedidos de menús de hamburguesas. La aplicación permitirá configurar un solo menú. El usuario podrá elegir el tipo de hamburguesa, pan, patatas y bebida. También hay elementos opcionales/extra.

El precio de un menú básico es de 8 euros pero algunas opciones tienen un coste adicional. Se mostrará el precio total del menú con las opciones elegidas, el IVA en € (21% adicional) y el precio de venta al público (tras sumar el IVA).

Las opciones obligatorias a elegir son:

- Hamburguesa a elegir: pollo, cerdo, ternera (+1€) o vegana (+1€).
- Pan a elegir: normal, integral o centeno.
- Patatas a elegir: fritas, gajo y caseras (+1 €).
- Bebida a elegir: refresco de cola, refresco de naranja, refresco de limón, agua y cerveza.

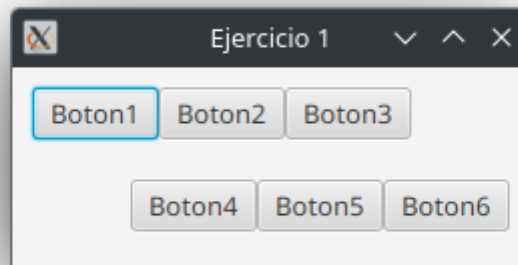
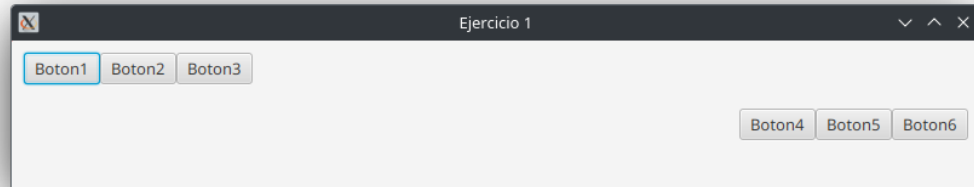
Las opciones extra/adicionales son:

- Hamburguesa doble (+2 €).
- Extra de queso (+0,50 €).
- Extra de patatas (+1 €).
- Salsas: ketchup, barbacoa, mostaza y thai. Pueden pedirse varias de cada (+0,50 cada una).
- Reparto a domicilio (precio estándar) o recogida en el local (-20% sobre el precio final).

4. Más ejercicios (Sin SceneBuilder)

1. Ejercicio 1

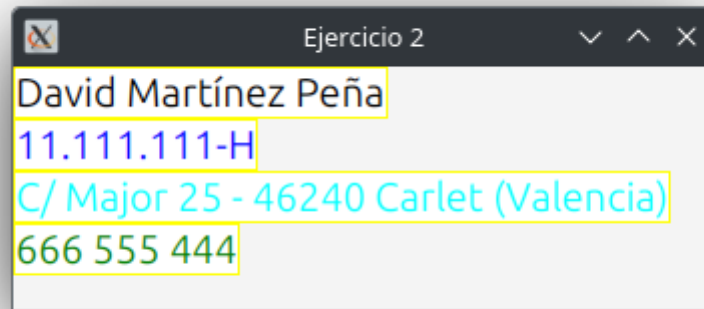
Escriba un programa que muestre la siguiente `stage` usando más de un contenedor. Los tres botones superiores se alinearan a la parte izquierda de la ventana, y los tres inferiores a la parte derecha.



1. Ejercicio 2

Escriba un programa que muestre cuatro líneas de texto con cuatro `Label`:

- Escribe tu nombre, DNI, dirección, teléfono en cada una de las etiquetas.
- Establece el fondo de las etiquetas en blanco.
- Establece el color del texto de las etiquetas en negro, azul, cian, verde respectivamente.
- Establece la fuente de cada etiqueta en TimesNewRoman/Ubuntu, negrita y 20 píxeles.
- Establece el borde de cada etiqueta en un borde de línea con color amarillo.



1. Ejercicio 3

Muestre un `stage` que contenga nueve etiquetas. Una etiqueta puede mostrar un icono de imagen para X, un icono de imagen para O, o nada. Lo que se muestra se decide al azar. Utilizar el Método `Math.random()` para generar un número entero 0, 1 o 2, que corresponde a mostrando un icono de imagen X, un icono O o nada (una imagen vacía).



5. Fuentes de información

- [Wikipedia](#)
- [Programación \(Grado Superior\) - Juan Carlos Moreno Pérez \(Ed. Rama\)](#)
- Apuntes IES Henri Matisse (Javi García Jimenez?)
- Apuntes AulaCampus
- [Apuntes José Luis Comesaña](#)
- [Apuntes IOC Programació bàsica \(Joan Arnedo Moreno\)](#)
- [Apuntes IOC Programació Orientada a Objectes \(Joan Arnedo Moreno\)](#)