



Proyecto JavaFX – “Túnel del Terror → Ruleta TRUCO/TRATO”

Autor: Alberto

Centro: MEDAC Sevilla

Módulo: M5 – Desarrollo de Interfaces



ÍNDICE

1. Introducción
 2. Objetivos del Proyecto
 3. Estructura General
 4. Explicación del Código
 - 4.1. Pantalla de Entrada al Túnel
 - 4.2. Paso de Datos entre Pantallas
 - 4.3. Pantalla de la Ruleta
 - 4.4. Estilo y Diseño (CSS)
 5. Dificultades y Aprendizaje
 6. Conclusión
 7. Capturas del Proyecto
-



1. Introducción

El proyecto “Túnel del Terror → Ruleta TRUCO/TRATO” es una miniaplicación creada con **JavaFX** y **Scene Builder**.

Su objetivo es ofrecer una experiencia divertida e inmersiva inspirada en **Halloween**, combinando animaciones, validaciones y un diseño temático.

2. Objetivos del Proyecto

- Crear una interfaz visual atractiva y temática.
- Validar correctamente los datos de entrada.
- Implementar una animación de ruleta con resultado aleatorio.
- Aprender a pasar información entre distintas pantallas en JavaFX.
- Ofrecer una experiencia fluida y coherente para el usuario.

3. Estructura General

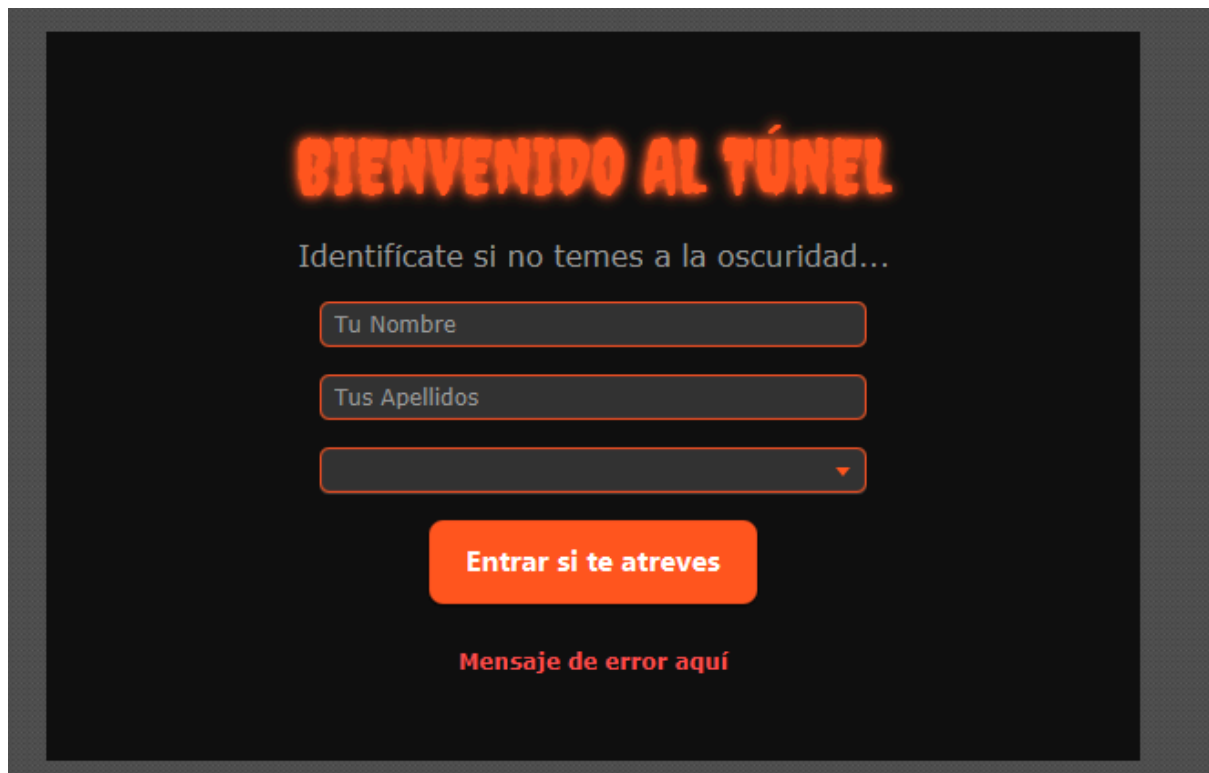
El proyecto se divide en dos pantallas principales:

1. **Entrada al Túnel** (`entrada.fxml` + `EntradaController.java`)
→ Recoge los datos del usuario (nombre, apellidos y curso).
2. **Ruleta TRUCO/TRATO** (`ruleta.fxml` + `RuletaController.java`)
→ Muestra los datos introducidos y lanza la animación de la ruleta.

Ambas pantallas están conectadas mediante **controladores JavaFX** y utilizan **FXML** para la interfaz.

4. Explicación del Código

4.1. Pantalla de Entrada al Túnel



Esta pantalla contiene tres campos:

- **Nombre** (TextField)
- **Apellidos** (TextField)
- **Curso** (ComboBox con valores: DAM1, DAM2, DAW1, DAW2, SMR1, SMR2)

Al pulsar el botón “**Entrar si te atreves**”, el programa:

1. Comprueba que todos los campos están completos.
2. Si falta alguno, muestra un mensaje claro y no deja avanzar.
3. Si todo es correcto, cambia de escena a la pantalla de la ruleta.

👉 Esta validación se implementa con simples condicionales (*if*) y etiquetas (*Label*) que muestran mensajes de error visibles.

🧠 4.2. Paso de Datos entre Pantallas

Para enviar la información del usuario (nombre, apellidos y curso) desde la primera pantalla a la segunda, se utiliza un **método setter** en el segundo controlador:

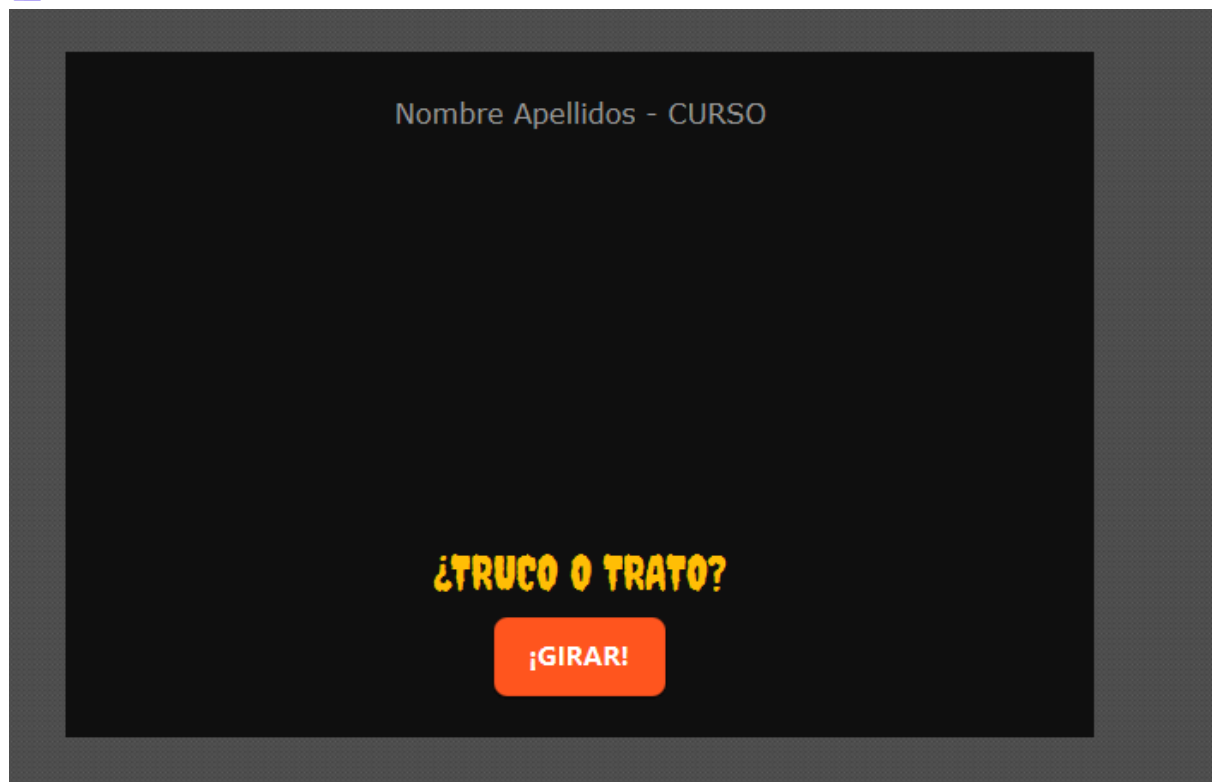
```
public void setDatos(String nombre, String apellidos, String curso)
{
    labelNombre.setText(nombre + " " + apellidos);
    labelCurso.setText(curso);
}
```

El controlador de entrada llama a este método justo antes de cambiar la escena.

💡 *Esto garantiza que los datos se muestren correctamente en la ruleta sin necesidad de variables globales.*



4.3. Pantalla de la Ruleta



En esta vista se muestra:

- El **nombre completo** del usuario.
- El **curso** seleccionado.
- Una **imagen de ruleta** que gira al pulsar el botón **GIRAR**.

La animación se hace con `RotateTransition`, ajustando su duración y ángulo para simular un giro realista.

```
RotateTransition rt = new RotateTransition(Duration.seconds(3),
ruleta);
rt.setByAngle(720 + Math.random() * 360);
rt.setInterpolator(Interpolator.EASE_OUT);
rt.play();
```

Mientras la ruleta gira:

- El botón **GIRAR** se desactiva para evitar múltiples giros.
- Al detenerse, se genera un número aleatorio (0 o 1).
- Si el número es **0** → **TRUCO** 🎩, si es **1** → **TRATO** 🍬.

Se muestra un mensaje final en pantalla con el resultado.

🕯 4.4. Estilo y Diseño (CSS)

El estilo está definido en un archivo `halloween.css`, con:

- Colores **negros y naranjas** típicos de Halloween.
- Tipografía decorativa pero legible.
- Fondos con imágenes o degradados oscuros.
- Botones con efectos hover y bordes redondeados.

Todo el diseño fue ajustado en **Scene Builder**, cuidando las **alineaciones y anclajes** para mantener la coherencia en cualquier tamaño de ventana.

⚙ 5. Dificultades y Aprendizaje

Durante el desarrollo tuve que **repetir el proyecto 3 o 4 veces desde cero**, porque inicialmente la animación y el paso de datos no funcionaban bien.

Cada intento me ayudó a comprender mejor:

- Cómo manejar **controladores JavaFX**.

- Cómo cargar escenas con **FXMLLoader**.
- Cómo animar elementos y controlar los eventos de finalización.

Gracias a esa práctica constante, logré una versión final totalmente funcional.

6. Conclusión

El proyecto “Túnel del Terror → Ruleta TRUCO/TRATO” combina programación, animación y diseño de interfaz de manera equilibrada.

Es un ejercicio excelente para reforzar los conocimientos de:

- **JavaFX,**
- **Interacción entre vistas,**
- **Validaciones, y**
- **Efectos visuales dinámicos.**

El resultado final es una app divertida, visualmente atractiva y completamente funcional, con espíritu de Halloween 🎃.