Documentación: Carrera de Coches con Interfaz

Andrea Sofía Pais Dos Santos

November 19, 2025

Índice

1 Descripción del problema

Tenemos que crear una aplicación que simula una carrera de coches, que necesita gestionar varios coches compitiendo al mismo tiempo en una pista. Para ello, creamos un programa que gestione el movimiento de los coches y muestre visualmente su progreso en la carrera. La aplicación tiene que ser capaz de:

1. Poder crear los nombres de los coches que participan.
2. Simular el movimiento aleatorio de cada coche con diferentes velocidades máximas.
3. Mostrar visualmente la posición de cada coche en la pista en tiempo real.
4. Gestionar la llegada a la meta y determinar el orden de llegada.
5. Reiniciar la carrera para nuevas pruebas.

2 Requisitos funcionales

Los requisitos funcionales que va a tener el sistema son:

RF01: Cada coche debe avanzar de forma aleatoria dentro de su velocidad máxima.
RF02: Los coches deben mostrar su posición actual en la pista visualmente.
RF03: El sistema debe clasificar correctamente el orden de llegada a la meta.
RF04: La aplicación debe permitir reiniciar la carrera.
RF05: Mostrar el puesto de llegada de cada coche al llegar a la meta.

3 Requisitos no funcionales

Los requisitos no funcionales que va a tener el sistema son:

RNF01: Interfaz intuitiva y visualmente atractiva.
RNF02: Tiempo de respuesta rápido en la actualización de posiciones.
RNF03: Los hilos deben finalizar correctamente al terminar la carrera.

4 Casos de uso

Descripción de casos de uso principales de la aplicación:

Caso de Uso	Descripción
Iniciar Carrera	El usuario inicia la simulación y los coches comienzan a moverse por la pista de forma concurrente.
Visualizar Progreso	El sistema muestra en tiempo real la posición de cada coche en la pista mediante una representación visual.
Finalizar Carrera	Cuando un coche alcanza los 800 metros, el sistema registra la llegada y muestra la posición final.
Reiniciar Simulación	El usuario puede limpiar la pista y comenzar una nueva carrera con los mismos coches.

5 Historias de usuario

Identificación	Nombre	Tarea	Objetivo
HU2	Visualización Carrera	El usuario quiere ver el progreso de los coches en tiempo real en una pista visual y reiniciar la carrera.	Seguir el desarrollo de la carrera y ver las posiciones de los coches.

6 Tecnologías Utilizadas

Java	Lenguaje de programación principal para la lógica de la aplicación	
JavaFX	Framework para crear la interfaz gráfica de usuario	
IntelliJ	IDE utilizado para el desarrollo del proyecto	

7 Clases e Interfaces

Explicación detallada de cada clase y los componentes de la interfaz gráfica desarrollados para la aplicación de carreras de coches.

7.1 Interfaz gráfica con JavaFX

7.1.1 HelloController

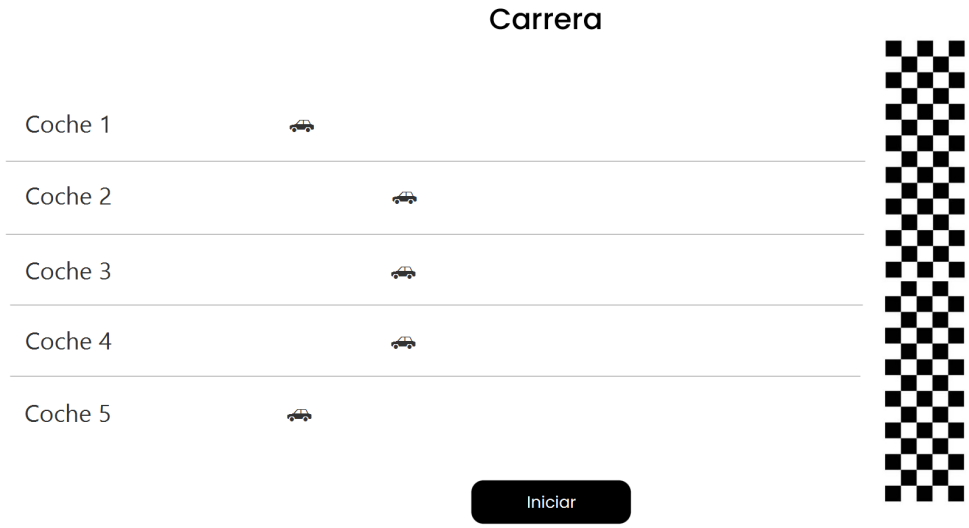
- Declaración de variables:

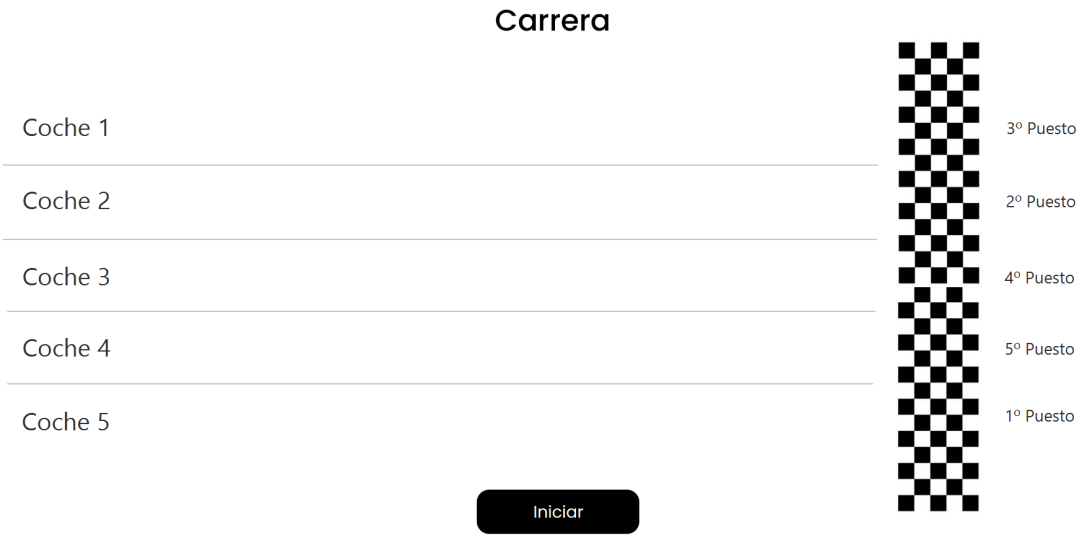
  1. Campos de texto para nombres (nombreCoche1...nombreCoche5) - Para ingresar o mostrar los nombres de los coches que participan en la carrera.
  2. Lista nombreCoches - ArrayList que contiene todos los campos de texto de nombres.

3. Campos de texto para pistas (pista01...pista48) - Representan las posiciones en la pista para cada coche.
4. Lista pistaCoches - ArrayList que contiene arrays de TextField representando cada fila de la pista.
5. Lista coches - ArrayList que contiene los objetos Coche que se crean.

- Función initialize():
  1. Inicializa la lista de nombres de coches y deshabilita la edición de los TextView que no me interesa.
  2. Asigna nombres por defecto ("Coche 1", "Coche 2", etc.).
  3. Llama a crearPista() para organizar los campos de la pista en filas para cada coche.
- Función crearPista():
  1. Organiza los campos de texto en 5 filas (una por coche) con 8 posiciones cada una.
  2. Cada posición representa 100 metros (meta total: 800 metros).
- Función iniciar():
  1. Reinicia la carrera y limpia las pistas.
  2. Crea nuevos objetos Coche con nombres y velocidades específicas.
  3. Inicia los hilos de cada coche con start().
- Función posicionCoche(): Actualiza visualmente la posición de cada coche en la pista.
- Función ordenLlegada(): Es la función que al llamarla nos enseña la posición de los coches (1^{o} Puesto, 2^{o} Puesto, etc.).

Imágenes de la carrera y del final de la carrera.





7.2 Clases en Java

7.2.1 Clase Coche

- Atributos:
  1. nombre - Identificador del coche.
  2. distanciaRecorrida - seguimiento del progreso del coche en metros.
  3. velocidadMaxima - límite máximo del coche.
  4. meta - distancia total de la carrera (800 metros).
  5. fila - índice que identifica la fila/pista del coche.
  6. llegada - controla el orden de llegada.
  7. controller - referencia al controlador para actualizar la interfaz.
  8. posicionllegada - posición final en la carrera.
- Constructor: Inicializa todos los atributos del coche.
- Función reiniciarCarrera(): Método para resetear el contador de llegadas.
- Función run():
  1. Bucle principal mientras no se alcance la meta.
  2. Actualiza distancia recorrida y posición visual.
  3. Controla la llegada a meta con sincronización para determinar orden.
  4. Tiene ausas aleatorias para simular tiempo real.
- La clase utiliza Platform.runLater() para actualizaciones de la interfaz.

7.2.2 Hello View (XML)

Estructura de la interfaz gráfica en FXML:

- AnchorPane como contenedor principal.
- Campos de texto para nombres de coches (5 campos).
- TextViews para representar la pista (5 filas $\times$ 8 columnas).
- Botón para iniciar/reiniciar la carrera.
- Organización visual clara que muestra coches como emojis de un coche y posiciones finales con texto.
- Vinculación de todos los elementos con sus respectivos IDs en el controlador.