

22 DE OCTUBRE DE 2023



# GAME DESIGN DOCUMENT

## RALLY TEAM TACTICS

BARRACUDA INTERACTIVE

ADRIÁN RUBIO GARRIDO  
ALEJANDRO ASENSIO PÉREZ  
ÁLVARO MARTÍN HITA  
DANIEL HERNÁNDEZ TAMAYO  
ERIC MARTÍNEZ GAMERO  
SERGIO MONTES VEREDAS



## TABLA DE CONTENIDO

Historial de versiones .....	4
Versión 0.1 (26/09/23) .....	4
Versión 0.2 (28/09/23) .....	4
Versión 0.3 (05/10/23) .....	4
Versión 0.4 (08/10/23) .....	4
Versión 0.6 (12/10/23) .....	5
Versión 0.7 (17/10/23) .....	5
Versión 0.8 (20/10/23) .....	5
Versión 1.0 (22/10/23) .....	6
Índice de figuras .....	7
Índice de ilustraciones.....	7
Índice de tablas .....	10
1 Introducción .....	11
1.1 Introducción al juego y su enfoque al aprendizaje pensamiento computacional .....	11
1.2 Contenido del documento .....	11
2 Referencias y concepto de juego .....	12
2.1 Principales referencias .....	12
2.2 Concepto de juego y bases de la jugabilidad .....	16
2.2.1 Rol del jugador .....	16
2.2.2 Objetivo del jugador, jugabilidad y gestión de recursos .....	16
2.2.3 Gameplay loop .....	17
3 Objetivo del proyecto: enseñanza del Pensamiento Computacional a niños pequeños.....	19
3.1 Los videojuegos en el aprendizaje del Pensamiento Computacional .....	19
3.2 Público objetivo.....	19
3.3 Destrezas del Pensamiento Computacional en el juego .....	19
3.4 Plataformas .....	20
4 Trasfondo .....	21
4.1 Ambientación .....	21
4.2 Narrativa.....	22
5 Mecánicas y elementos de juego .....	24
5.1 Cámara y perspectiva .....	24
5.2 Gameplay loop en detalle .....	25
5.2.1 Fase de reconocimiento en detalle .....	25
5.2.2 Fase de ejecución en detalle .....	27

5.3 Mecánicas de juego en detalle.....	29
5.3.1 Comportamiento del vehículo en cada tipo de sección.....	29
5.3.2 Relieve en el trazado .....	33
5.3.3 Gestión de recursos: combustible y daño sufrido.....	37
5.4 Controles y periféricos .....	39
6 Progresión .....	40
6.1 Objetivos del jugador .....	40
6.2 Estructura del juego .....	40
6.3 Clasificación en línea .....	43
7 Interfaz de usuario .....	44
7.1 Requisitos de la interfaz .....	44
7.2 Diagrama de flujo de navegación .....	45
7.3 Diseño esquemático de la UI.....	47
8 Arte.....	53
8.1 Estética del juego .....	53
8.1.1 Referencias artísticas .....	53
8.1.2 Lenguaje visual del juego .....	54
8.2 Concept .....	58
8.2.1 Concepts del coche .....	58
8.2.2 Concepts de props.....	59
8.3 Diseño visual de la UI .....	64
8.3 Assets 3D .....	65
9 Sonido.....	66
9.1 Efectos de sonido .....	66
9.2 Música .....	67
10 Producción.....	68
10.1 Modelo de negocio del juego y plan de financiación.....	68
10.1.1 Información sobre el usuario .....	68
10.1.2 Mapa de Empatía .....	69
10.1.3 Caja de Herramientas .....	70
10.1.4 Modelo de Lienzo .....	71
10.2 Marketing .....	72
10.3 Versiones preliminares del producto .....	73
10.4 Producto final .....	74
11 Conclusiones.....	75
Referencias y bibliografía .....	76

Anexo .....	77
-------------	----

## HISTORIAL DE VERSIONES

### VERSIÓN 0.1 (26/09/23)

Borrador del GDD, versión de presentación de la idea inicial de acuerdo con el tema:

- Escritura de [1 Introducción](#).
- Escritura de [2 Referencias y concepto de juego](#). Se incluyen diagramas de flujo para el *gameplay loop*.
- Escritura de [3 Objetivo del proyecto: enseñanza del Pensamiento Computacional a niños pequeños](#). Se incluye una tabla sobre las destrezas.

### VERSIÓN 0.2 (28/09/23)

Inclusión de la plantilla con todos los capítulos y apartados que se consideran necesarios en este momento.

### VERSIÓN 0.3 (05/10/23)

Modificación del concepto de juego y explicación de mecánicas inicial:

- Cambio de [2.2 Concepto de juego y bases de la jugabilidad](#) para adaptarse mejor a la temática y simplificar el gameplay. Se incluyen nuevas ilustraciones y diagramas.
- Escritura de los dos primeros apartados del capítulo 5: [5.1 Cámara y perspectiva](#) y [5.2 Gameplay loop en detalle](#). Se incluyen ilustraciones y diagramas.

### VERSIÓN 0.4 (08/10/23)

Ampliación del capítulo de mecánicas, ahora describiendo obstáculos y gestión de recursos:

- Escritura del tercer apartado del capítulo 5: [5.3 Mecánicas de juego en detalle](#). Se incluyen ilustraciones y diagramas explicativos.

Explicación de la progresión en el juego:

- Escritura de los casi todos los apartados del capítulo 6: [6.1 Objetivos del jugador](#) y [6.2 Estructura del juego](#). Se incluyen ilustraciones y diagramas explicativos.

## VERSIÓN 0.6 (12/10/23)

Cambios en la elección de vocabulario para los diferentes elementos de juego, de tal manera que este sea consistente en todo el documento.

Modificación del *gameplay loop* general y adición de secciones para la explicación detallada de los puntos de control en las fases de juego:

- Retoque en los diagramas de flujo de [2.2.3 Gameplay loop](#) para hacerlos consistentes en vocabulario con el resto del documento y reflejar cambios en el diseño del juego.
- Escritura de los apartados [Colocación de notas en la pista](#) y [Cantado de notas durante la carrera](#).

## VERSIÓN 0.7 (17/10/23)

Reorganización de los apartados de los capítulos 7 y 8, referentes a UI/UX y arte, respectivamente:

- Ahora el diseño de UI va antes del capítulo de arte y se cambia el último apartado a [7.3 Diseño esquemático de la UI](#), para incluir en una versión futura los bocetos o esquemas de las pantallas para todas las plataformas.
- El apartado de arte ahora se desglosa en más apartados, aún vacíos. Se incluye uno para el diseño visual de la UI y otro para *concept art*.

Integración de parte del contenido del apartado 7 que ya está listo:

- Integración de los apartados [7.1 Requisitos de la interfaz](#) y [7.2 Diagrama de flujo de navegación](#). Se incluyen diagramas.

## VERSIÓN 0.8 (20/10/23)

Ecritura del [4 Trasfondo](#) del juego, con una ambientación con imágenes de referencia y una narrativa donde se cuenta la historia del copiloto al que el jugador encarna en el juego.

Integración del modelo de negocio, revisado, en el GDD, concretamente en el apartado [10.1 Modelo de negocio del juego y plan de financiación](#).

Paleta de colores del documento de diseño visual del juego, logo del juego, del estudio y Splash Art añadidos.

## VERSIÓN 1.0 (22/10/23)

Versión del GDD para la entrega del prototipo del videojuego. En esta iteración se incorpora contenido a:

- Apartado [7.3 Diseño esquemático de la UI](#), con mockups de las pantallas diseñados hasta la fecha (integración del documento de diseño de interfaces revisado).
- Apartados [8.1 Estética del juego](#) (integración del documento de diseño visual del juego revisado) y [8.2 Concept](#), con imágenes y bocetos.

## ÍNDICE DE FIGURAS

### ÍNDICE DE ILUSTRACIONES

Ilustración 1. Colin McRae Rally 3 (PS2) .....	12
Ilustración 2. DiRT Rally (PS4).....	12
Ilustración 3. WRC 10 (PC) .....	13
Ilustración 4. Virtual Racing (Arcade) .....	13
Ilustración 5. Ridge Racer (Arcade) .....	14
Ilustración 6. Daytona USA (Arcade) .....	14
Ilustración 7. Art of Rally (PC).....	15
Ilustración 8. You Suck at Parking (Xbox One).....	15
Ilustración 9. Gameplay loop de un tramo.....	17
Ilustración 10. Gameplay loop de una etapa completa .....	17
Ilustración 11. Leyenda para diagramas de flujo sobre el gameplay .....	18
Ilustración 12. WRC: Audi Quattro de Walter Röhrl en 1985.....	21
Ilustración 13. WRC: Toyota Celica GT-Four de Carlos Sainz en 1990 .....	22
Ilustración 14. Perspectiva y movimiento de la cámara .....	24
Ilustración 15. Gameplay loop de la fase de reconocimiento .....	26
Ilustración 16. Gameplay loop de la fase de ejecución .....	28
Ilustración 17. Leyenda para el comportamiento del vehículo.....	29
Ilustración 18. Iconos de las instrucciones de tipo de trazado .....	29
Ilustración 19. Comportamiento del coche en recta.....	30
Ilustración 20. Comportamiento del coche en curva poco cerrada.....	31
Ilustración 21. Comportamiento del coche en curva cerrada.....	31
Ilustración 22. Comportamiento del vehículo en curva muy cerrada.....	32
Ilustración 23. Comportamiento del coche en horquilla .....	32
Ilustración 24. Sketch de una rampa .....	33
Ilustración 25. Sketch de una meseta.....	34
Ilustración 26. Sketch de un valle.....	34
Ilustración 27. Sketch de un hundimiento .....	35
Ilustración 28. Sketch de un charco.....	35
Ilustración 29. Sketch de un bache seguido de un charco en una curva poco cerrada .	36

Ilustración 30. Sketch de las barras de combustible y daño .....	37
Ilustración 31. Sketch de los recursos: sin combustible.....	38
Ilustración 32. Sketch de los recursos: coche destrozado.....	38
Ilustración 33. Estructura de un tramo cronometrado (días 1 a 3 -fases 1, 2 y 3-) .....	41
Ilustración 34. Estructura de una etapa completa cronometrada (último día -fase 4-) .....	41
Ilustración 35. Estructura del juego.....	42
Ilustración 36. Estructura de un evento .....	42
Ilustración 37. Diagrama de navegación de inicio del juego .....	45
Ilustración 38. Diagrama de navegación: menús y juego.....	46
Ilustración 39. Diseño de la Splash Screen con el logo de la empresa.....	47
Ilustración 40. Diseño de la selección de idioma (primera vez que se inicia el juego) ..	47
Ilustración 41. Diseño de la pantalla de inicio.....	48
Ilustración 42. Diseño del pop-up de inicio de sesión, si no se ha iniciado previamente en el dispositivo .....	48
Ilustración 43. Diseño del pop-up de registro .....	49
Ilustración 44. Diseño del pop-up de inicio de sesión .....	49
Ilustración 45. Diseño del menú principal .....	50
Ilustración 46. Diseños de los pop-ups de tienda, garaje y ajustes, respectivamente ..	50
Ilustración 47. Diseño de la pantalla de selección de etapa .....	51
Ilustración 48. Diseño de la pantalla de selección de tramo en la etapa.....	51
Ilustración 49. Representación del flujo de navegación en Figma .....	52
Ilustración 50. Ridge Racer (Namco, 1993) .....	53
Ilustración 51. Art of Rally (Funselektor Labs, 2020).....	53
Ilustración 52. Muestra de mubenk regular .....	56
Ilustración 53. Mapa de caracteres de checkbox regular .....	56
Ilustración 54. Logo provisional del juego .....	57
Ilustración 55. Diseño de posible fondo .....	57
Ilustración 56. Concept de coche en perspectiva.....	58
Ilustración 57. Concept de coche de perfil .....	59
Ilustración 58. Concept de un punto de control .....	59
Ilustración 59. Concept de la salida .....	60
Ilustración 60. Concept de diferentes variantes de árbol .....	60

Ilustración 61. Concept de diferentes rocas.....	61
Ilustración 62. Concept de un barril .....	61
Ilustración 63. Concept de una pila de ruedas .....	62
Ilustración 64. Concept de una valla de obra .....	62
Ilustración 65. Concept de un cono de tráfico .....	63
Ilustración 66. Mapa de Empatía del usuario final.....	69
Ilustración 67. Caja de Herramientas .....	70
Ilustración 68. Canvas del modelo de negocio.....	71

## ÍNDICE DE TABLAS

Tabla 1. Destrezas del Pensamiento Computacional en el juego .....	20
--	----

## 1 INTRODUCCIÓN

Este documento cubre todos los aspectos referentes al diseño del juego **Rally Team Tactics (RTT)**, en desarrollo por el equipo de *Barracuda Interactive* para PC, navegadores web y dispositivos Android y cuyo lanzamiento está previsto para finales de 2023.

### 1.1 INTRODUCCIÓN AL JUEGO Y SU ENFOQUE AL APRENDIZAJE PENSAMIENTO COMPUTACIONAL

Como breve introducción al juego, **Rally Team Tactics (RTT)** es un *arcade racer* donde el jugador asume el rol del copiloto de un equipo de rally. Como tal, este debe analizar el trazado del circuito para dar las instrucciones adecuadas al piloto antes de que comience la prueba.

El juego pone gran énfasis en el ensayo y error gracias a la posibilidad de volver al inicio. El jugador tiene así la posibilidad de resolver el problema basándose en lo acontecido anteriormente. La identificación de patrones (respuesta de las físicas del vehículo al relieve y el itinerario) le permiten mejorar la elección de instrucciones y hacer así una gestión de los recursos (gasolina y daño recibido) más eficiente en su siguiente intento.

### 1.2 CONTENIDO DEL DOCUMENTO

A lo largo de los siguientes capítulos, se cubre en este documento:

- Principales referencias y jugabilidad
- Propósito del proyecto: enseñanza del Pensamiento Computacional a niños pequeños
- Herramientas y programas utilizados
- Proceso de diseño del juego en todos sus ámbitos de manera extensiva
  - Trasfondo: ambientación, narrativa y entornos
  - Mecánicas y elementos de juego
  - Progresión
  - Interfaces y experiencia de usuario (UI y UX)
  - Arte
  - Música y sonido
- Producción y modelo de negocio

## 2 REFERENCIAS Y CONCEPTO DE JUEGO

### 2.1 PRINCIPALES REFERENCIAS

**Rally Team Tactics (RTT)** es un *arcade racer* inspirado por otros juegos de simulación de carreras o de conducción *arcade*, con ejemplos provenientes desde los 90 hasta la actualidad. Entre estas referencias, encontramos:

- Franquicias de rally y carreras *off-road* realistas:
  - **Colin McRae Rally**, conocida actualmente como **DiRT** y **DiRT Rally** (*Codemasters*, EA, 1998-2020). Ver Ilustración 1 e Ilustración 2.
  - **WRC** (múltiples desarrolladores, EA, 2001-2023). Ver Ilustración 3.

De estas se toman su énfasis en la simulación de las físicas de los vehículos sobre diferentes terrenos.



Ilustración 1. Colin McRae Rally 3 (PS2)



Ilustración 2. DiRT Rally (PS4)



Ilustración 3. WRC 10 (PC)

- Juegos de carreras para recreativas de los años 90:
  - **Virtua Racing** (Sega AM2, 1992). Ver Ilustración 4.
  - **Ridge Racer** (Namco, 1993). Ver Ilustración 5.
  - **Daytona USA** (Sega AM2, 1993). Ver Ilustración 6.

Estos juegos sirven como inspiración tanto para el estilo artístico colorido y retro de RTT como para mecánicas arcade como las pruebas contrarreloj, el uso del turbo o *leaderboards*.



Ilustración 4. Virtual Racing (Arcade)



Ilustración 5. Ridge Racer (Arcade)



Ilustración 6. Daytona USA (Arcade)

- Juegos *indie* de conducción *arcade* contemporáneos:
  - ***Art of Rally*** (*Funselektor Labs*, 2020): las principales influencias provenientes de este título son su *setting* en la época dorada del rally y su arte 3D *low-poly* de gran calidad y su iluminación. Ver Ilustración 7.
  - Otros aspectos por destacar:
    - Su perspectiva de cámara aérea.
    - Sus físicas que no llegan al realismo de los títulos del primer punto pero que encuentran un buen balance entre el control *arcade* y la simulación centrada en las características del terreno (relieve, superficie) y las condiciones climatológicas.

- **You Suck at Parking** (*Happy Volcano*, 2022): otra gran inspiración, especialmente por su naturaleza ensayo-error (reinicio rápido). El juego introduce además elementos de puzzles que ponen a prueba la memoria y el reconocimiento de patrones del jugador. También cuenta con una cámara aérea, con la posibilidad de visualizar el nivel libremente antes de empezar un intento. Este último elemento es una fuerte influencia para RTT. Ver Ilustración 8.



Ilustración 7. Art of Rally (PC)



Ilustración 8. You Suck at Parking (Xbox One)

## 2.2 CONCEPTO DE JUEGO Y BASES DE LA JUGABILIDAD

**Rally Team Tactics (RTT)** se concibe como un *arcade racer* en 3D ambientado en la época dorada del rally.

### 2.2.1 ROL DEL JUGADOR

El jugador toma el control de un **copiloto de rally** en lugar del piloto. Este giro le da una nueva perspectiva al género, pues en lugar de tener agencia directa sobre el vehículo, **se deben “cantar” las notas que contienen código nemotécnico y que describen las características de la carrera**. Además, como a los copilotos en la realidad, al jugador se le otorga la **responsabilidad de supervisar aspectos del coche** como el nivel de combustible restante o los daños sufridos.

Como no controlamos el coche directamente, qué notas “cantemos” determinarán el comportamiento de este. Para situar las anotaciones del copiloto en el trayecto, se permite en todo momento tener una “vista de pájaro” con una **cámara libre para navegar por toda la etapa**, como si tuviésemos acceso a las cámaras del helicóptero de televisión.

### 2.2.2 OBJETIVO DEL JUGADOR, JUGABILIDAD Y GESTIÓN DE RECURSOS

El objetivo es lograr el **mejor tiempo en cada etapa de rally**, que se divide en **tramos cronometrados**. Antes del día de la etapa completa, **el jugador se debe enfrentar a cada tramo cronometrado por separado**, refinando su estrategia de cara al día de la etapa.

Solo podemos **situar las instrucciones mientras el vehículo se encuentre parado**, por lo que para cambiarlas se debe reiniciar el tramo o etapa mediante **un reinicio que conserva las instrucciones seleccionadas**.

La experimentación puede llevar al jugador de chocar en la primera curva a conseguir su mejor tiempo en el tramo o etapa a base de ensayo y error. **Si el coche choca o sufre demasiados daños, es obligatorio regresar al inicio** para idear una nueva estrategia.

Siempre que se finaliza un tramo o etapa **se muestra el tiempo conseguido**, y si se ha completado el tramo o etapa en el pasado también si se ha superado al histórico y por cuánto.

La gestión de recursos juega un papel importante. Tanto el daño sufrido por el vehículo como el combustible restante deben tenerse en cuenta. Al inicio del tramo o etapa, el coche tiene 0% de daños y 100% de combustible.

- Si el coche sufre una caída o impacta con algún objeto su integridad se ve comprometida, y el **medidor de daño (damage)** incrementa en diferentes cantidades.

- El tanque de combustible siempre se va consumiendo, pero utilizar el turbo o acelerar tras una frenada importante producen un mayor consumo del **medidor de gasolina (fuel)**.

### 2.2.3 GAMEPLAY LOOP

El *gameplay loop* básico se refleja en el siguiente diagrama para un tramo:

#### Gameplay Loop de un tramo

En un tramo los jugadores solo pueden elegir instrucciones para diez secciones de pista, que determinan el tramo. Pueden buscar mejorar su tiempo en este tramo para prepararse para la etapa final

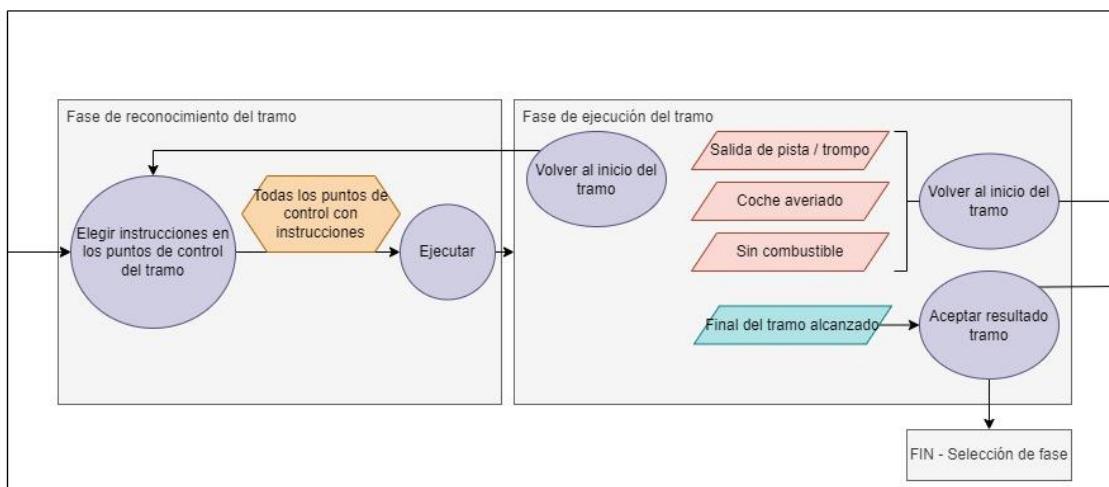


Ilustración 9. Gameplay loop de un tramo

Y el siguiente para una etapa en su totalidad:

#### Gameplay Loop de una etapa completa

En la etapa completa los jugadores comienzan con las instrucciones que eligieron en los tramos en los días anteriores, ahora pueden variarlas para enlazar dichos tramos de manera óptima

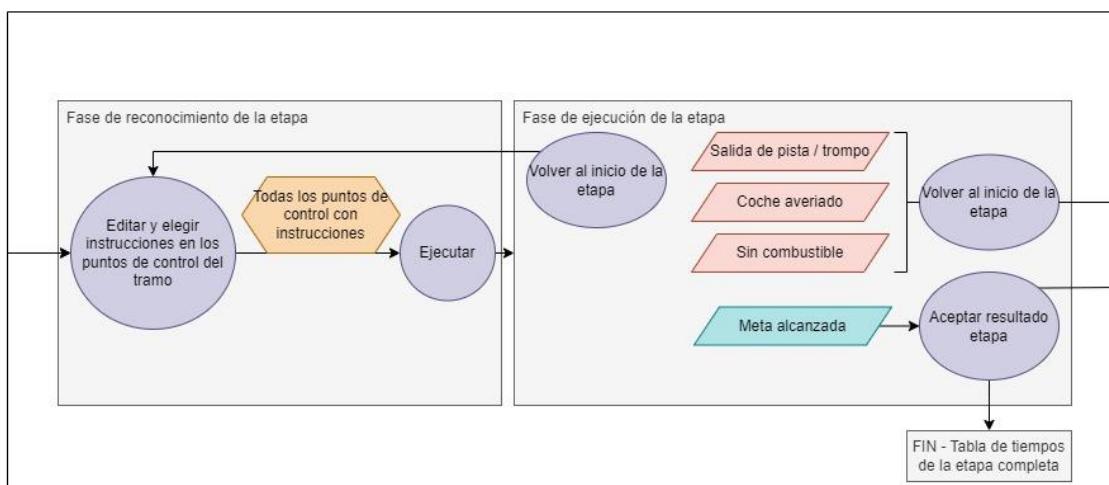


Ilustración 10. Gameplay loop de una etapa completa

En la leyenda que se muestra seguidamente se explica el significado de las formas en los diagramas de flujo referentes a *gameplay*:



Ilustración 11. Leyenda para diagramas de flujo sobre el gameplay

## FASE DE RECONOCIMIENTO

Como se plasma en el diagrama, el jugador puede elegir anotaciones (instrucciones) durante la **fase de reconocimiento**. Una vez se hayan seleccionado las instrucciones deseadas para todos los puntos de control, se puede proceder a ejecutar.

## FASE DE EJECUCIÓN

El paso a la **fase de ejecución** arranca el vehículo desde el punto de salida. Ahora podemos ver el vehículo realizar el trayecto de acuerdo a las órdenes situadas en reconocimiento. Durante la ejecución seguimos teniendo el control de la cámara y podemos reiniciar en cualquier momento, volviendo a la fase de reconocimiento.

## POSIBLES EVENTOS Y SUS CONSECUENCIAS

Se pueden producir una serie de eventos negativos, que nos forzarán a reiniciar, como una salida de pista completa o el consumo del tanque de gasolina.

Por otro lado, los eventos positivos posibles son alcanzar el final del tramo o la meta. La meta es el final solo en el último tramo y en la etapa completa. Si el jugador acepta el resultado (no reinicia) se daría por finalizado el tramo o etapa. Además. De tratarse de una etapa completa, se mostraría la **pantalla de clasificación (leaderboards)** correspondiente; de lo contrario, se pasaría directamente a la selección de fase.

### 3 OBJETIVO DEL PROYECTO: ENSEÑANZA DEL PENSAMIENTO COMPUTACIONAL A NIÑOS PEQUEÑOS

#### 3.1 LOS VIDEOJUEGOS EN EL APRENDIZAJE DEL PENSAMIENTO COMPUTACIONAL

Llamamos Pensamiento Computacional (PC) a la habilidad cognitiva relacionada con el uso de estrategias computacionales para la resolución de problemas. Su aprendizaje desde una edad temprana puede proporcionar amplios beneficios en la vida cotidiana. Por este motivo, la enseñanza del PC es parte del currículo escolar en España y otros países desde la etapa de educación infantil.

Los videojuegos orientados a potenciar el PC pueden tener una mayor efectividad que otras actividades como la programación o los juegos de mesa. Los niños y niñas en la actualidad tienen una relación estrecha con el mundo de los videojuegos, y las posibilidades que estos ofrecen en la enseñanza apenas se han explotado.

Por los motivos citados anteriormente, el juego **en desarrollo** pretende explorar ese nicho, creando una experiencia lúdica que impulse el uso del PC dentro de un entorno de entretenimiento.

#### 3.2 PÚBLICO OBJETIVO

El *target* del videojuego son niños y niñas desde 8 hasta 12 años. Esta demográfica juega con frecuencia a títulos de cierta complejidad como *Fortnite* o *Roblox*, por lo que se tiene en cuenta este factor a la hora de diseñar el juego: pese a tratarse de un producto educativo, no subestima las capacidades de los niños y evita las características habituales de los *serious games*. Se busca reforzar el PC sin hacerlo explícito, comprometer la diversión ni la posibilidad de que el juego pueda ser disfrutado por audiencias de mayor edad.

#### 3.3 DESTREZAS DEL PENSAMIENTO COMPUTACIONAL EN EL JUEGO

Desde la concepción del juego se busca potenciar múltiples facultades propias del PC. Las destrezas particulares que se pretenden ejercitan jugando RTT son las siguientes:

Destreza del PC	Cómo se pone en práctica en el juego
Abstracción	Esta habilidad se instruye gracias a la decisión de diseño que pone al jugador en los pies del copiloto. El jugador debe abstraerse de la reacción inmediata de la conducción, situándose en una <b>capa superior de planificación y toma de decisiones</b> . La “vista de pájaro” pone énfasis en esta capacidad de <b>ver el problema en su totalidad y reconocer así sus elementos clave</b> .

<b>Pensamiento algorítmico</b>	En RTT se pone a disposición del jugador un conjunto de notas que el copiloto “canta” en determinados momentos, según su colocación y orden. Se debe <b>crear un algoritmo</b> , al fin y al cabo, una <b>secuencia de acciones</b> concretas que da lugar al mejor resultado posible.
<b>Descomposición</b>	En base a la respuesta del vehículo en un determinado tramo con una determinada serie de instrucciones, el jugador puede adquirir la capacidad de <b>deducir</b> cómo se va a comportar el coche en el mismo escenario u otro similar en función de sus decisiones en la fase de reconocimiento.
<b>Evaluación</b>	El RTT se presta especial atención a la <b>gestión de recursos</b> , es este caso, el combustible y el daño sufrido. Se deben considerar a corto y medio plazo la consumición de combustible y el desgaste del coche. Esto implica no solo tener en mente el tramo inmediatamente posterior, sino la etapa en su totalidad. Reservar combustible para un momento concreto o arriesgar la integridad del vehículo en el punto adecuado de la pista son ejemplos de decisiones que pueden dar lugar a mejores tiempos.  Además, mediante el ensayo y error, el jugador asimila la <b>detección de fallos</b> , y toma <b>decisiones ajustadas al objetivo</b> . Si el coche responde de una determinada forma a una secuencia de instrucciones que producen un consumo de recursos correspondiente, puede usar ese conocimiento para siguientes intentos.
<b>Generalización</b>	RTT pone el foco en la repetición para encontrar soluciones mejores. <b>Resolver el problema en base a soluciones anteriores</b> es parte fundamental del diseño. Cuando el jugador <b>encuentra patrones y similitudes</b> en la reacción del vehículo ante distintos obstáculos o trazados, aprende a utilizar las estrategias que llevan a un resultado más satisfactorio.  Aunque el jugador no conoce al detalle las físicas del coche (ni se pretende que lo haga), a medida que se va familiarizando con su comportamiento, toma decisiones que le resultan más favorables.

Tabla 1. Destrezas del Pensamiento Computacional en el juego

### 3.4 PLATAFORMAS

Como los niños suelen disponer de tabletas en los centros educativos o en su hogar, es fundamental que el juego se desarrolle no solo para **escritorio y navegadores Web**, sino también para dispositivos táctiles Android (en este caso **tablets Android**).

## 4 TRASFONDO

### 4.1 AMBIENTACIÓN

*RTT* está ambientado en la actualidad, y el jugador viajará alrededor del mundo compitiendo cara a cara con los mejores pilotos.

El mundo de *Rally Team Tactics* es una realidad alternativa donde la época dorada de los *rallies* nunca cesó y esta competición automovilística mantiene una tremenda popularidad.

Se suele considerar que la época dorada del rally fue la década de los 80, cuando se creó la categoría de Grupo B, que permitía a los fabricantes desarrollar coches extremadamente potentes y veloces, y las restricciones de velocidad y técnicas eran mínimas. Las carreras eran espectaculares y peligrosas: los pilotos se ponían en riesgo su vida en cada etapa y las masas de espectadores se agolpaban al borde de las carreteras para ver a sus ídolos de cerca. Algunos de los modelos más emblemáticos de esta época fueron el Audi Quattro, el Lancia Delta S4, el Peugeot 205 T16 o el Ford RS200.



Ilustración 12. WRC: Audi Quattro de Walter Röhrl en 1985



Ilustración 13. WRC: Toyota Celica GT-Four de Carlos Sainz en 1990

## 4.2 NARRATIVA

El varias veces campeón del mundo de rally **Erik Ström** estaba en el pico de su carrera. Su destreza y agresividad en la carretera lo hacían casi imparable.

Pese a haber crecido en una familia humilde, el talento de Ström lo había llevado de la pobreza a ser una de las figuras más admiradas y reconocibles del mundo del automovilismo.

No obstante, su escalada al zenit no fue sin consecuencias para su salud mental. Tras una serie de derrotas en etapas donde era el favorito, Ström no era capaz de concentrarse en la pista. Jóvenes pilotos estaban despuntando, y Ström no tenía un coche competitivo y estaba harto de seguir las instrucciones de los ingenieros, a los que consideraba ineptos. Las averías eran demasiado frecuentes, y la actuación cautelosa del equipo estaba acabando con su paciencia. Desesperado, firmó con un constructor dispuesto a dejarle desobedecer las órdenes del equipo, incluido su copiloto, si lo veía necesario. Además, sabía que su nuevo coche no cumplía con la normativa, y que las piezas prohibidas habían sido camufladas y para pasar irregularmente las comprobaciones de seguridad.

La temporada de rally de 2013 comenzó bien para Ström, posicionándose entre los tres primeros justo antes de una de las etapas más temidas del circuito anual: la *Acropolis Rally of Greece*. Terrenos montañosos, acantilados rocosos y calor sofocante sobre un asfalto antiguo y unas secciones campo a través desafiantes. No era la etapa para la audacia ni el riesgo, pero Ström no dudó en jugarse el campeonato en tierras griegas.

El día de la carrera, el piloto sueco estaba codo a codo con los mejores tiempos marcados por sus rivales más cercanos en la clasificación, y decidió acelerar en uno de los tramos más infames de la etapa: la curva del abismo. Fue aquí donde Ström, cegado por la ambición, no frenó a tiempo y cayó por el barranco. Cuando los equipos médicos le encontraron, Erik, inconsciente, no respondía. Pero la suerte le había sonreído, pues no había sido un accidente fatal.

Sin embargo, esto le supondría su ingreso hospitalario, donde estaría años en rehabilitación, sometiéndose a numerosas operaciones, hasta que pudo recuperar parte de la movilidad. Su tren inferior no era funcional, nunca podría volver a competir. Su tiempo de recuperación le llevó a replantearse su vida y su carrera profesional: sabía que no podría volver a competir, y no encontraba su propósito para seguir adelante. Fueron años oscuros, pero Ström superó la depresión y, tras finalizar su rehabilitación, regresó al mundo de los *rallies*. Su discapacidad le impedía conducir, pero sabía que tenía mucho conocimiento que aportar a las nuevas generaciones.

En 2019, fundó una escuela en su ciudad natal y hizo de mentor para decenas de aspirantes a futuros pilotos. Se aseguraba también de que los jóvenes pusiesen su salud mental y física por encima de cualquier objetivo en la pista.

Ström era una figura respetada en el automovilismo y los rumores de su vuelta a los *rallies* se multiplicaban, pero nadie sabía qué rol tomaría de ser así. A sus 56 años, en 2022, se asoció con algunos de sus amigos y antiguos conocidos de la industria para montar un equipo y firmar un contrato con una reputada constructora. Sus coches llevaban los colores de su país natal en su carrocería.

Erik tenía claro cuál era su sitio ahora: el de copiloto. Quería redimirse por todos sus errores y llevar el talento de su escuela a lo más alto a base de aprendizaje y constancia.

Dado el pistoletazo de salida a la temporada 2023, Ström acaparaba todas las portadas: ¿será este el verdadero regreso del rey de los *rallies*?

## 5 MECÁNICAS Y ELEMENTOS DE JUEGO

### 5.1 CÁMARA Y PERSPECTIVA

La perspectiva de la cámara es **aérea**, emula la vista desde un helicóptero de televisión. El entorno tridimensional se puede navegar tanto en fase de reconocimiento como en fase de ejecución.

Las limitaciones de movimiento de la cámara son:

- Imposibilidad de hacer *zoom*
- Imposibilidad de regular la altitud
- La inclinación es fija, de tal manera que se observen con suficiente claridad las características del terreno y el trayecto del circuito

El jugador tiene control, por tanto, sobre:

- La posición en el plano horizontal definido a la altitud predeterminada
- El giro de la cámara, 360 grados de rotación

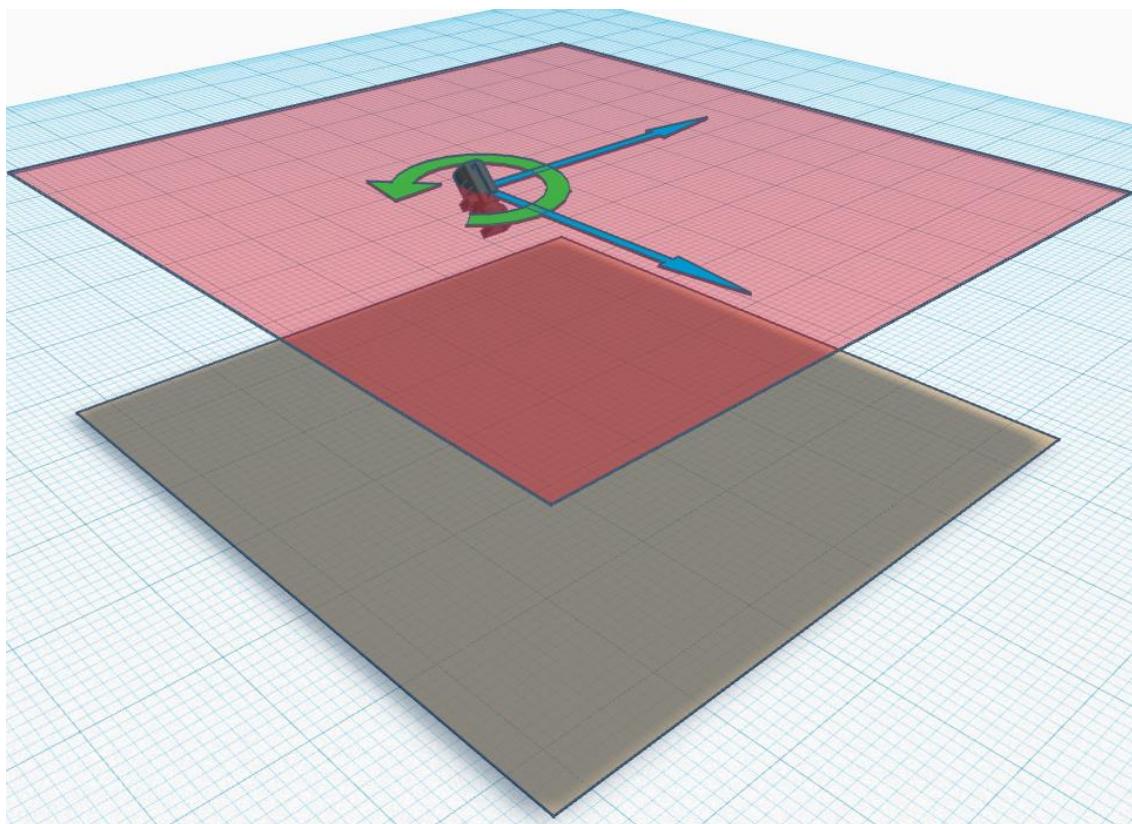


Ilustración 14. Perspectiva y movimiento de la cámara

Las flechas azules y verdes representan el movimiento horizontal y giro de la cámara, respectivamente, mientras que el plano rojo indica la altitud fija, y como se puede observar, la inclinación de la cámara también está bloqueada (ver Ilustración 14).

## 5.2 GAMEPLAY LOOP EN DETALLE

En este apartado se van a describir con detenimiento las fases que componen el *gameplay loop* del juego.

Como se indicó en el apartado [2.2.3 Gameplay loop](#), el *gameplay loop* de RTT está separado en dos fases o turnos: reconocimiento y ejecución. Esto es igual en los días de preparación (los distintos tramos) y en el día final (la etapa completa).

### 5.2.1 FASE DE RECONOCIMIENTO EN DETALLE

Durante el reconocimiento, el jugador puede navegar con libertad tal como se ha descrito en [5.1 Cámara y perspectiva](#), y, además:

Seleccionar un punto de control

Limpiar todo el circuito de instrucciones

Comenzar prueba

Acceder al menú de ajustes

Los puntos de control son puntos del circuito donde se debe especificar la anotación que se va a “cantar” al piloto cuando llegue a cada uno de ellos. Al **seleccionar un punto de control**, el jugador tiene las siguientes opciones:

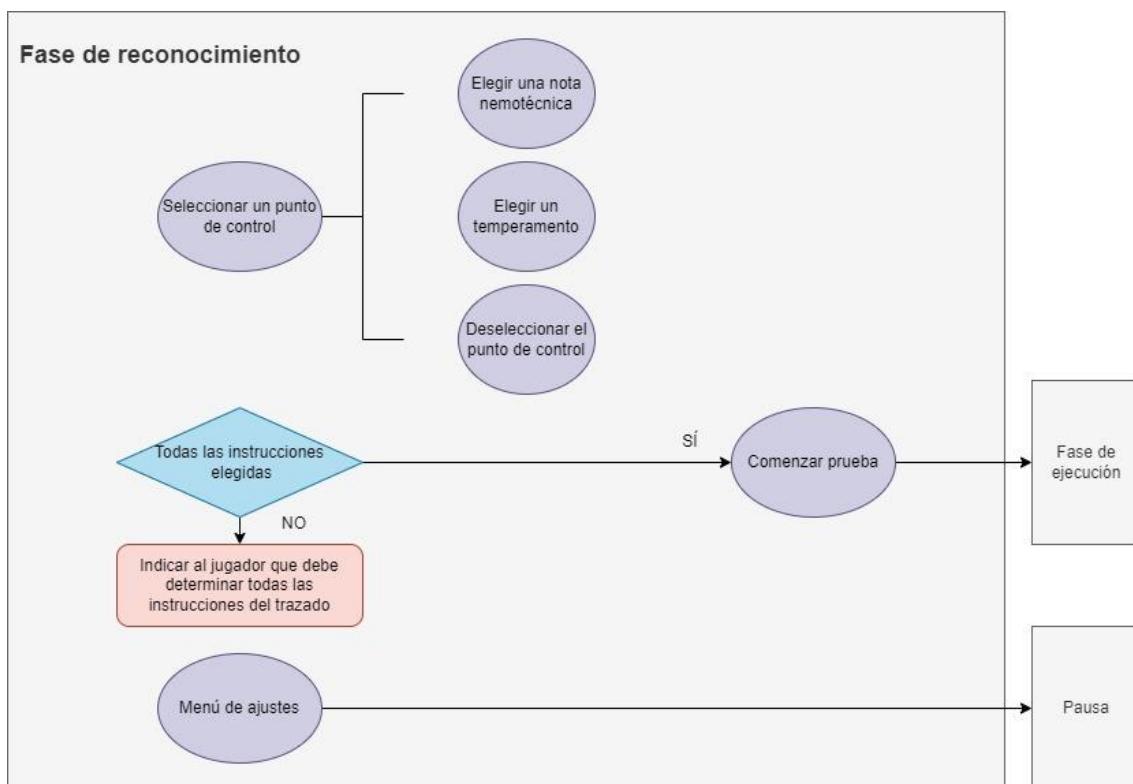
- A. **Seleccionar la nota nemotécnica** que ve más conveniente en función de el trazado inmediatamente posterior al punto de control. Entre estas notas encontramos, por ejemplo, “curva muy cerrada” o “recta”.
- B. **Elegir el temperamento** con el que realizar la acción. Esto es posible gracias a un *slider continuo* que permite determinar si se quiere un acercamiento **más cauteloso o más agresivo**.
  - Por defecto, cuando se coloca una nota por primera vez en un punto, el *slider* se encuentra en una posición intermedia, moderada.
  - Para ilustrar esta idea, supongamos que tenemos la nota “curva muy cerrada”: cuanto más agresivos seamos, el frenado se producirá más tarde y con mayor fuerza, mientras que, si optamos por un acercamiento más cauteloso, el frenado será más largo y comenzará mucho antes.  
En el caso de una nota de tipo “recta”, los dos extremos se corresponderían con reducir la velocidad y acelerar, siendo el punto medio del *slider* mantener la velocidad.  
Todos los casos se explican al detalle más adelante ([5.3 Mecánicas de juego](#)).
- C. **Deseleccionar un punto de control**, que es tan sencillo como pulsar o clicar en otro lugar del circuito.

Por otro lado, en todo momento se puede **limpiar el circuito**, que sería equivalente a comenzar de cero el nivel: todas las instrucciones situadas se eliminarían.

La acción que daría paso a la fase de ejecución es **comenzar la prueba**, eso sí, esto solo se puede hacer si se han colocado notas en todos los puntos de control.

Además de todo lo anterior, siempre es posible **entrar en el menú de ajustes**, pausando el juego. Aquí, entre otras cosas (configuración), se permite salir de la prueba.

El siguiente diagrama muestra el *gameplay loop* correspondiente a la fase de reconocimiento en mayor detalle:



**Ilustración 15. Gameplay loop de la fase de reconocimiento**

Ver la leyenda de la Ilustración 11 para el significado de las diferentes formas.

### COLOCACIÓN DE NOTAS EN LA PISTA

En este apartado se hace énfasis en el sistema de selección de puntos de control. Los puntos de control deben quedar claramente señalizados en la pista, y deben proveerse opciones de configuración a los jugadores para que los puedan distinguir con facilidad. Se explican a continuación cómo se plantea esta cuestión en función de la fase de juego en la que se encuentre el jugador:

- El uso de banderas a los laterales de la pista es un marcador diegético de la posición de estos puntos de control, que recordemos se encuentran entre secciones de pista cuya separación el jugador no puede discernir. Esta

**señalización se mantiene entre fases de reconocimiento y ejecución**, ya que es parte del mundo de juego.

- De forma extradiegética, se puede mostrar un muro luminoso perpendicular al trazado en esos puntos para hacerlos más notables. Además, el color puede variar si este ha sido elegido ya o si está vacío aún. **Este muro solo es visible en fase de reconocimiento.**
- También de forma extradiegética, aquellos puntos de control donde se ha seleccionado instrucción nemotécnica (nota) y temperamento pueden mostrar la elección de manera flotante o sobre la pista. El temperamento recordemos está controlado por un *slider*, pero el ícono puede variar únicamente al superar un determinado valor en dicho *slider*. Por ejemplo, la barra se dividiría en tres partes, y el ícono solo cambiaría cuando se pasase a un tercio distinto. Como el muro luminoso, **estos iconos son únicamente visibles en fase de reconocimiento.**

---

### 5.2.2 FASE DE EJECUCIÓN EN DETALLE

Durante la fase de ejecución, se simula la prueba de *rally* de acuerdo con las notas nemotécnicas y temperamento elegidos a lo largo del trazado. El vehículo recorre el trayecto automáticamente, reaccionando según sus físicas con el relieve y las curvas de la carretera. Además, el copiloto “canta” las notas cuando el coche pasa por cada una de ellas (*voiceover*) para dotar de mayor realismo a la carrera.

Además de mover libremente la cámara, hay dos acciones que se pueden realizar siempre en esta fase:

- A. **Acceder al menú de ajustes** supone lo mismo que en la fase de reconocimiento
- B. **Reiniciar** implica que el coche volverá a la salida del tramo/etapa, conservándose todas las notas (instrucciones) elegidas previamente

Si el coche choca quedando inmóvil, se queda sin combustible o sufre daños irreparables, estas son las únicas opciones disponibles.

No obstante, si se completa el tramo o etapa de principio a fin, también se nos da la posibilidad de **aceptar el resultado**, que se muestra en el temporizador junto con la diferencia de tiempo respecto al histórico. En caso de que el jugador vea el resultado satisfactorio, entonces se le reconducirá a una pantalla donde podrá ver una clasificación con otros jugadores.

Veamos un diagrama al detalle de esta fase:

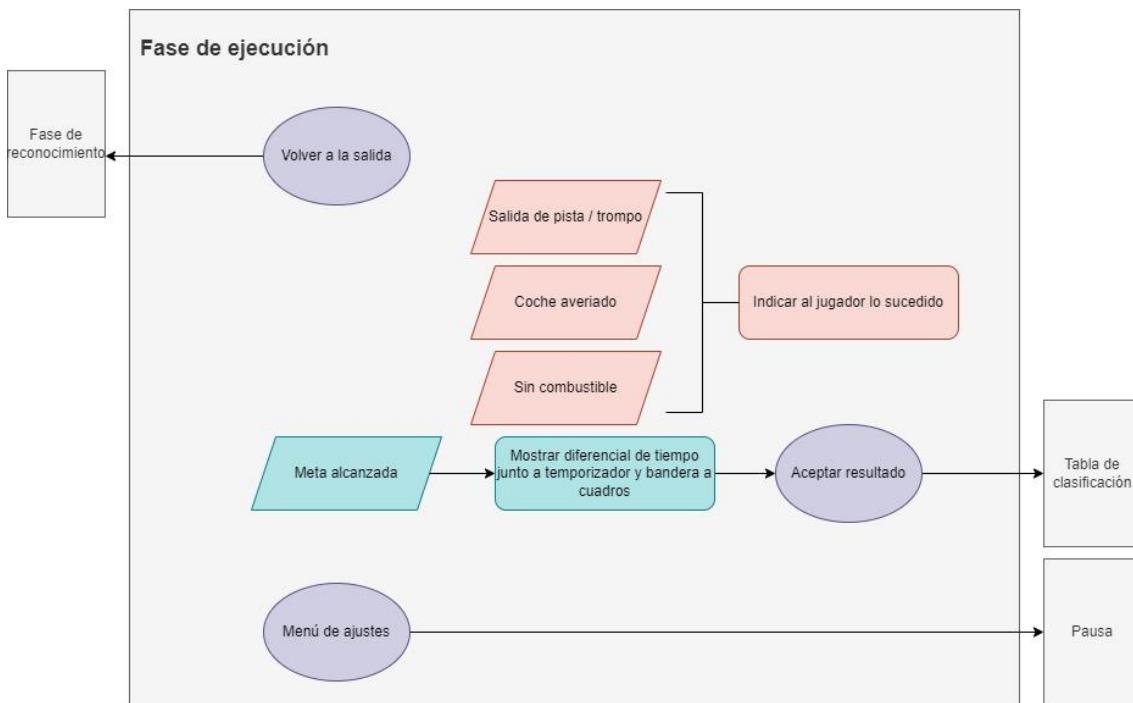


Ilustración 16. Gameplay loop de la fase de ejecución

Ver la leyenda de la Ilustración 11 para el significado de las diferentes formas.

#### CANTADO DE NOTAS DURANTE LA CARRERA

Al igual que para el reconocimiento, este apartado trata cómo se indican los puntos de control al jugador, pero esta vez en la fase de ejecución.

- Las banderas físicas que marcan los puntos de control siguen presentes en esta etapa, son elementos diegéticos.
- Los muros luminosos ya no están presentes, ni tampoco los iconos de instrucción y temperamento, es decir, ningún elemento extradiegético flotante se ve.
- Para que el jugador tenga un refuerzo sobre las decisiones tomadas en la fase de reconocimiento, cada vez que el vehículo pase por un punto de control, **el copiloto (voiceover) “canta” las notas**, no solo diciendo el nombre de esta, sino también el grado de agresividad determinado por el temperamento. Al ser el temperamento continuo y no discreto, como en la fase de reconocimiento, se tiene que discretizar en varios estados. Estos estados corresponden con los tercios del *slider*, siendo estos equivalentes a acompañar la nota nemotécnica en el “cantado” de expresiones: “curva cerrada, tomar con cuidado”, “recta, recorrer con agresividad” o “curva muy cerrada, acometer con moderación”.

Otra forma de recordar al jugador durante la fase de ejecución las instrucciones elegidas sería **mostrarlas además en el HUD como iconos** al tiempo que son “cantadas” (esto podría ser configurable).

## 5.3 MECÁNICAS DE JUEGO EN DETALLE

En este apartado se describen las mecánicas de *Rally Team Tactics* con detenimiento.

En primer lugar, se va a explicar el comportamiento del vehículo ante las diferentes curvas y obstáculos si se elige su nota nemotécnica correcta, así como el efecto del temperamento en cada caso. En caso de no elegirse la nota correspondiente, el coche seguirá las directrices de la seleccionada, dando lugar a resultados que normalmente serán negativos, saliendo de la pista o chocando, pero que en raras ocasiones pueden optimizar el recorrido. La clave al fin y al cabo es la experimentación.

*RTT* presenta una serie de secciones de carretera predefinidas por su curvatura, todas ellas de la misma longitud. Es justo entre estas secciones en el trazado donde se eligen las notas y el temperamento.

### 5.3.1 COMPORTAMIENTO DEL VEHÍCULO EN CADA TIPO DE SECCIÓN

A continuación, se mostrarán bocetos de todos los tipos de sección y se explica cómo se comportaría el vehículo de ser seleccionada la nota correcta, en tres casos concretos en el *slider* de temperamento: cauteloso, moderado y agresivo. La interpolación entre estos casos daría lugar a los comportamientos intermedios.

La siguiente leyenda indica el significado del código de colores de las flechas situadas en el trazado:

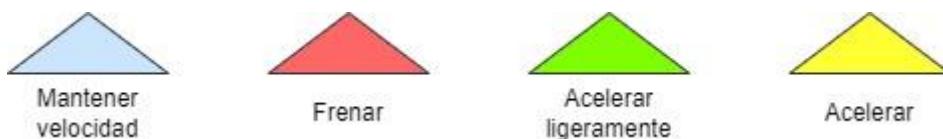


Ilustración 17. Leyenda para el comportamiento del vehículo

La imagen inferior muestra los iconos de las notas del código nemotécnico creados para los bocetos:

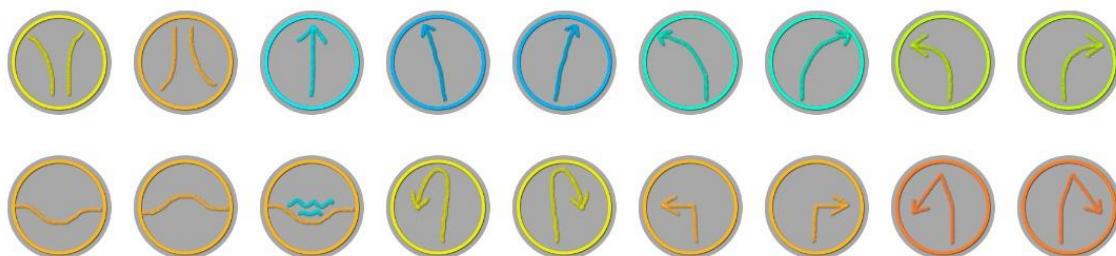


Ilustración 18. Iconos de las instrucciones de tipo de trazado

## OBSERVACIONES IMPORTANTES

- Las líneas guía no se muestran al jugador en ningún momento, son para *debug* y diseño.
- No hay solo siete o tres posiciones en el *slider*, no es discreto: el resto de posibles puntos son intercalaciones entre los extremos y el centro.
- Otro detalle de importancia es que en todos los casos presentados en cada sección se da por hecho que el coche empieza en el centro, lo cual no será así en la mayor parte de los casos. Siempre se va a tender progresivamente a adoptar esa posición intermedia. Por ejemplo, si se acaba de salir de una curva pegado a la izquierda y a continuación hay una recta, el vehículo buscará el centro. En resumen, el coche va a buscar siempre acercarse lo más posible al trayecto ideal.
- No se especifican valores de los parámetros como la velocidad en el diseño, se deben probar diferentes números durante la programación del juego hasta que se encuentren los valores de velocidad, frenado (reducción de la velocidad) y aceleración (aumento de la velocidad) que mejor plasman el desempeño de un coche de *rally in-game*.

## RECTA

En este primer caso se van a mostrar ejemplos más graduales, para que se vea cómo se interpola entre un extremo del *slider* de temperamento y el opuesto.



Ilustración 19. Comportamiento del coche en recta

Como se observa, a medida que se toma un perfil más agresivo, la aceleración es más fuerte y tiene una mayor duración.

---

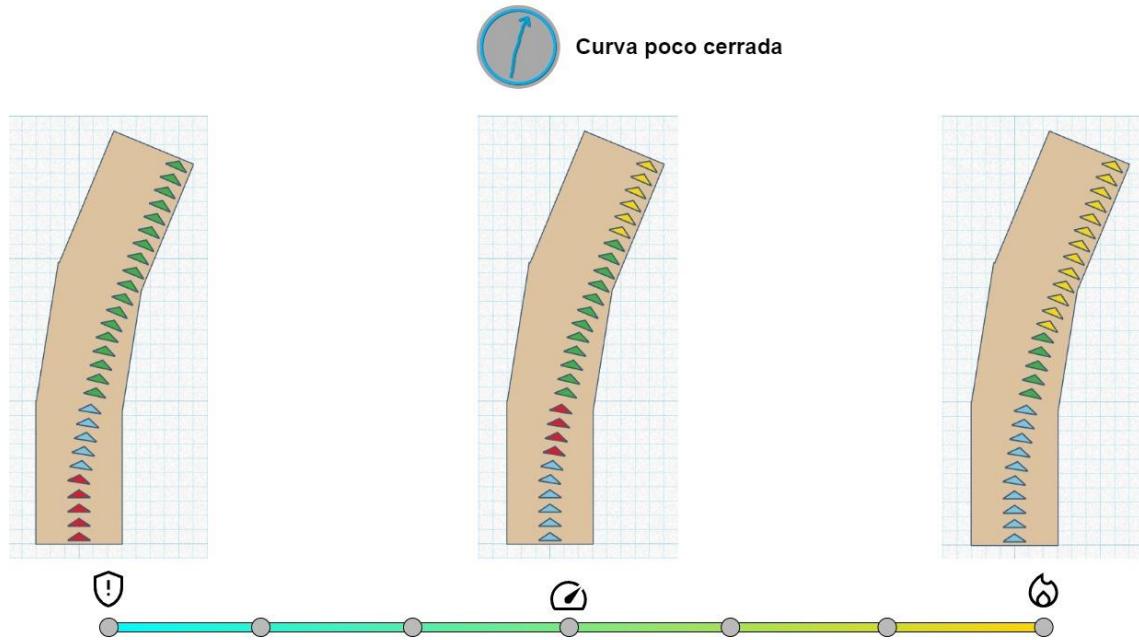
**CURVA POCO CERRADA**


Ilustración 20. Comportamiento del coche en curva poco cerrada

Aquí se observa por primera vez la presencia de tramos de frenado. **No hay diferentes variantes de frenado como con la aceleración, solo un tipo que supone una reducción gradual de la velocidad.**

---

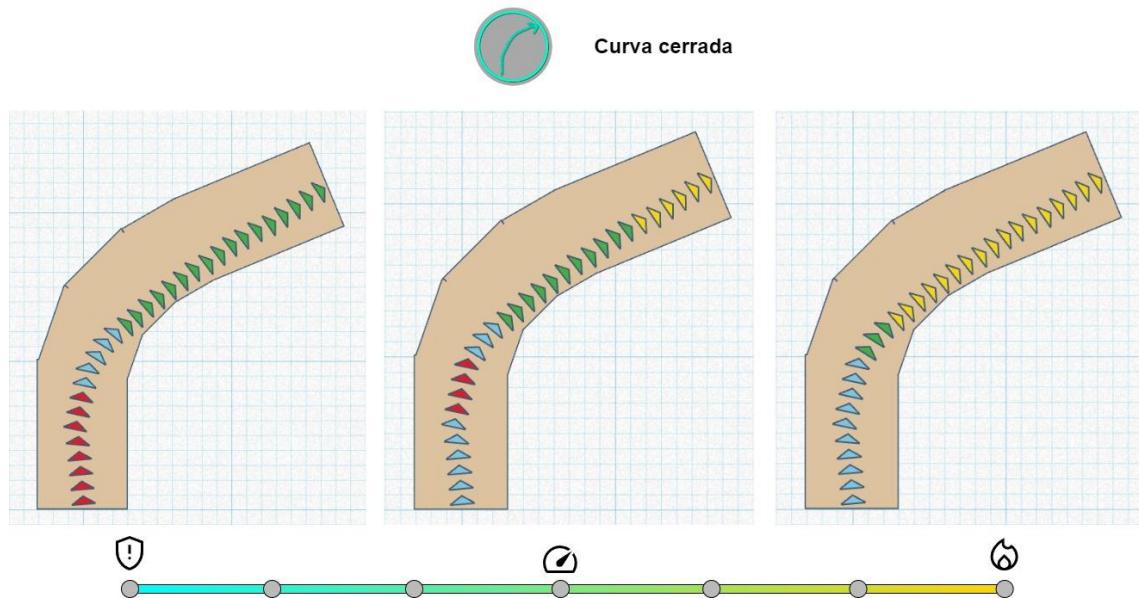
**CURVA CERRADA**


Ilustración 21. Comportamiento del coche en curva cerrada

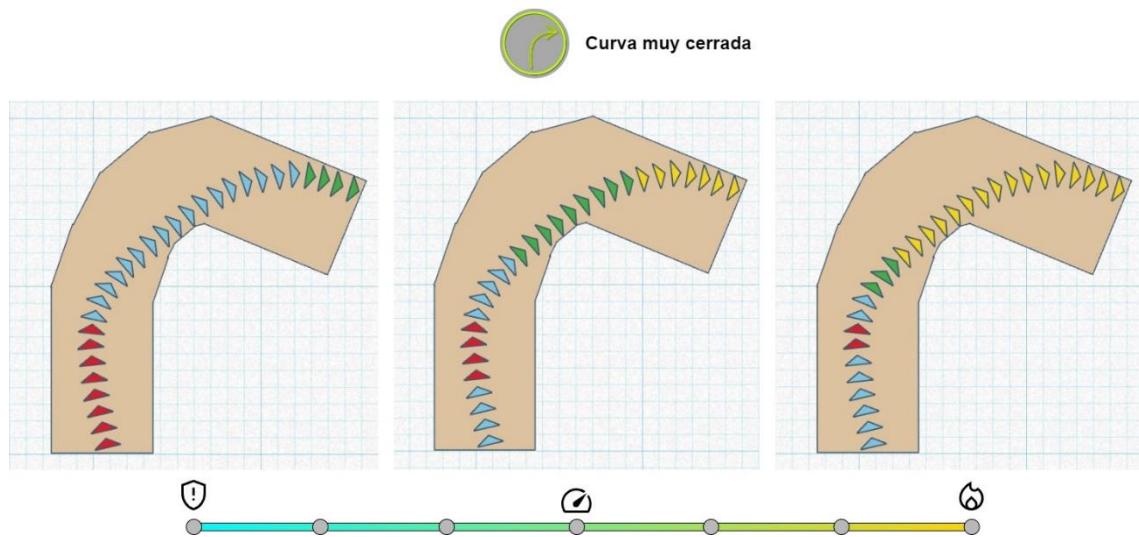
**CURVA MUY CERRADA**

Ilustración 22. Comportamiento del vehículo en curva muy cerrada

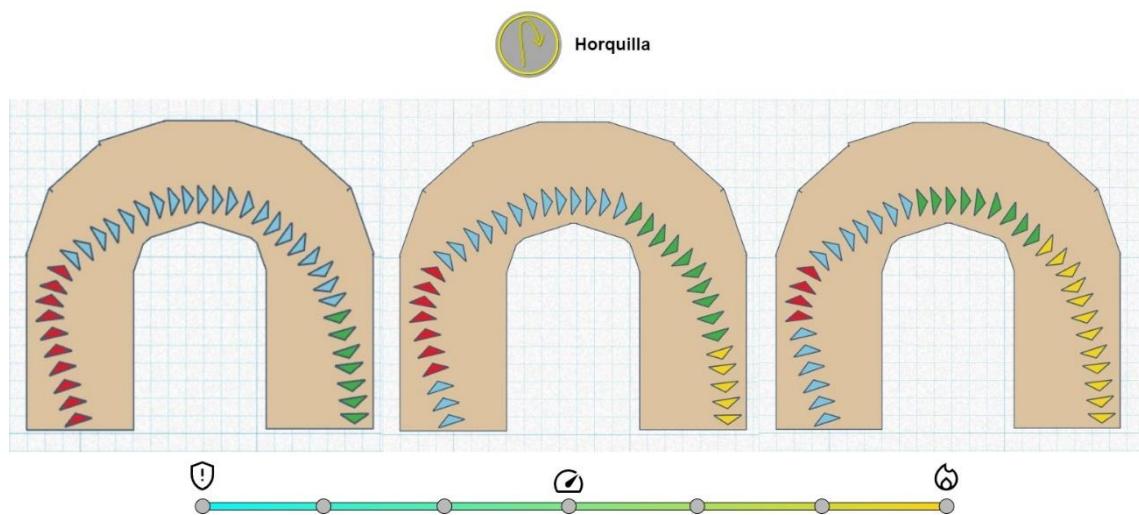
**HORQUILLA**

Ilustración 23. Comportamiento del coche en horquilla

### 5.3.2 RELIEVE EN EL TRAZADO

Para dar mayor diversidad a las etapas y plantear diferentes escenarios aún con secciones repetidas, se distribuyen a lo largo del itinerario obstáculos en forma de variaciones del relieve. Estas variaciones no afectan a cómo afronta el coche cada sección, pues eso está completamente determinado por la nota elegida y el temperamento seleccionado. Lo que hacen estos obstáculos es generar reacciones en las físicas del vehículo que le dificultan seguir el trayecto objetivo o provocan daños.

Es aquí donde entra en juego un buen manejo del temperamento: considerar el relieve de las secciones siguientes para determinar con qué nivel de cuidado o agresividad queremos entrar en cada una. Por ejemplo, si tenemos una recta con un bache a la mitad quizás sea más inteligente entrar con cautela, mientras que si se presenta una pendiente al inicio quizás sea mejor salir de la curva anterior con mayor agresividad para no perder mucha velocidad. Además, las colisiones del vehículo y caídas al dar saltos suponen daños, otro factor a considerar a la hora de enfrentarse a obstáculos en el circuito.

Los obstáculos se distribuyen en tres categorías: baches, depresiones y charcos. En principio estos ocupan todo el ancho de la pista, variando solo en altitud y longitud:

- Los **baches** serán porciones de una sección con una elevación seguida de un descenso. Pueden ser más o menos agudos en función del diseño de nivel elegido. Además, existen dos variantes: **rampas** y **mesetas**.

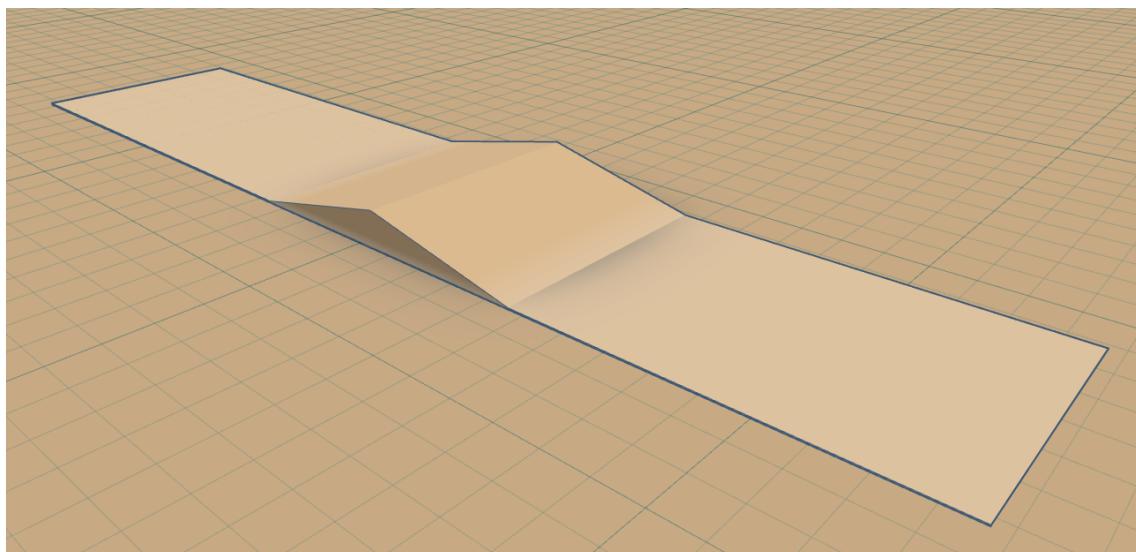


Ilustración 24. Sketch de una rampa

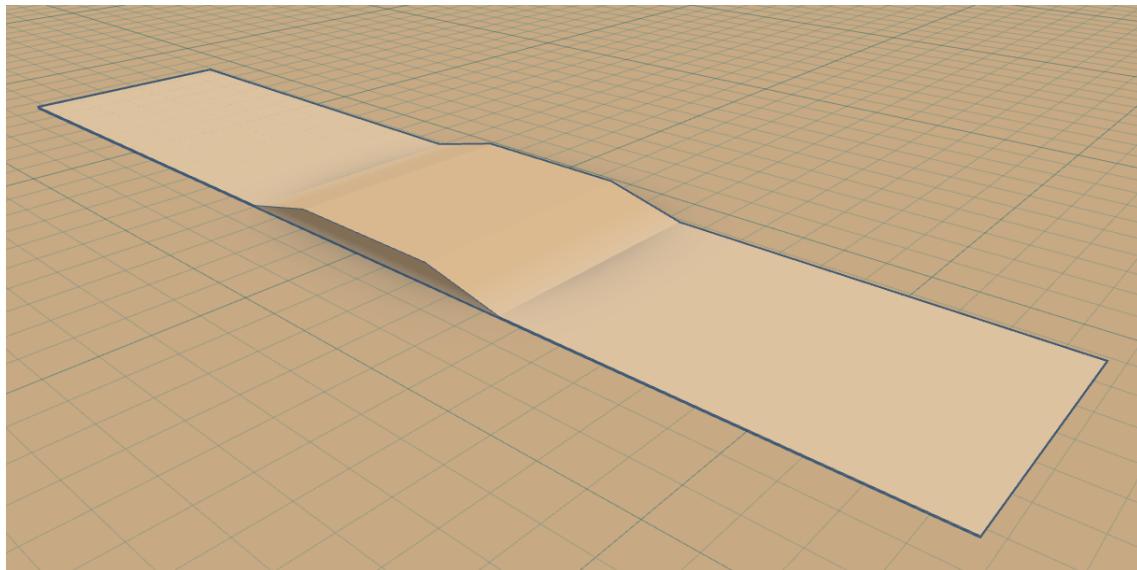


Ilustración 25. Sketch de una meseta

- Las **depresiones** son justo lo opuesto a los baches, es decir, hundimientos en el terreno de mayor o menor tamaño y longitud, a elección del diseñador de niveles. Las depresiones tienen dos variantes: **valles** y **hundimientos**.

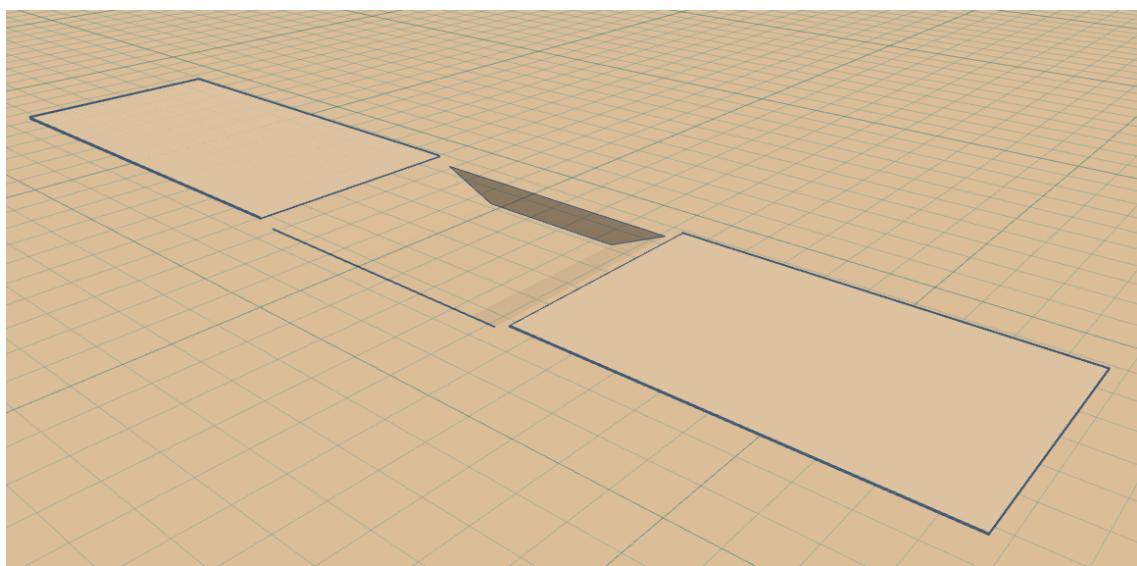


Ilustración 26. Sketch de un valle

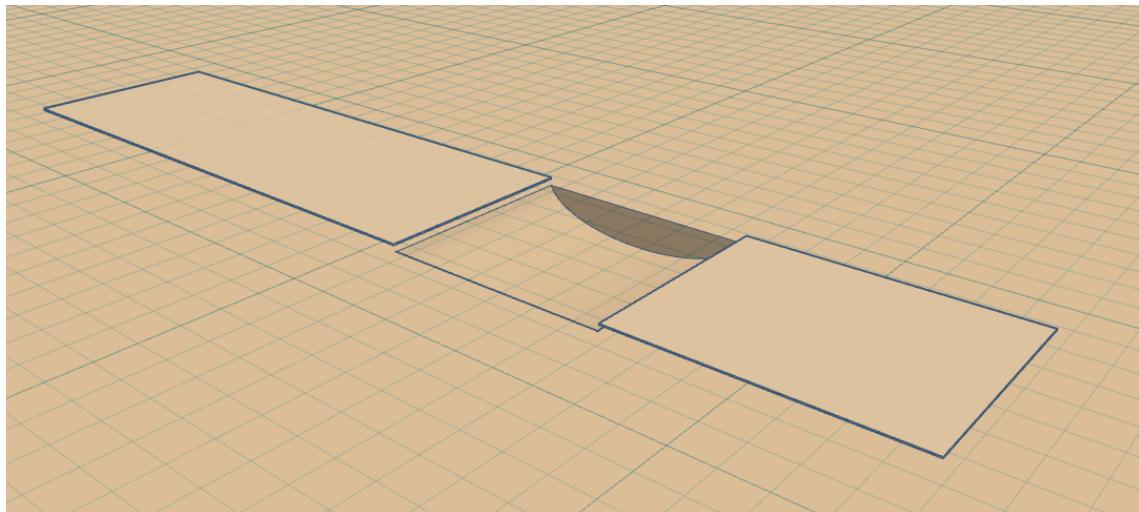


Ilustración 27. Sketch de un hundimiento

- Los **charcos** son zonas de una sección cubiertas por una fina capa de agua que ralentizan el avance del coche o incluso hacerlo resbalar y perder la dirección en curvas. Realmente se trata de valles de muy poca profundidad que están llenos de agua.

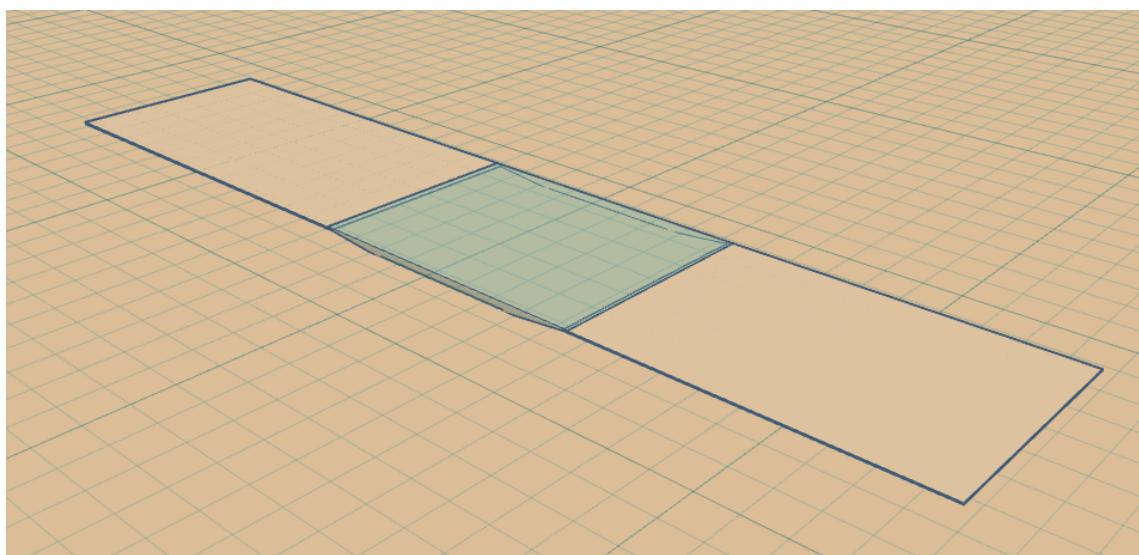


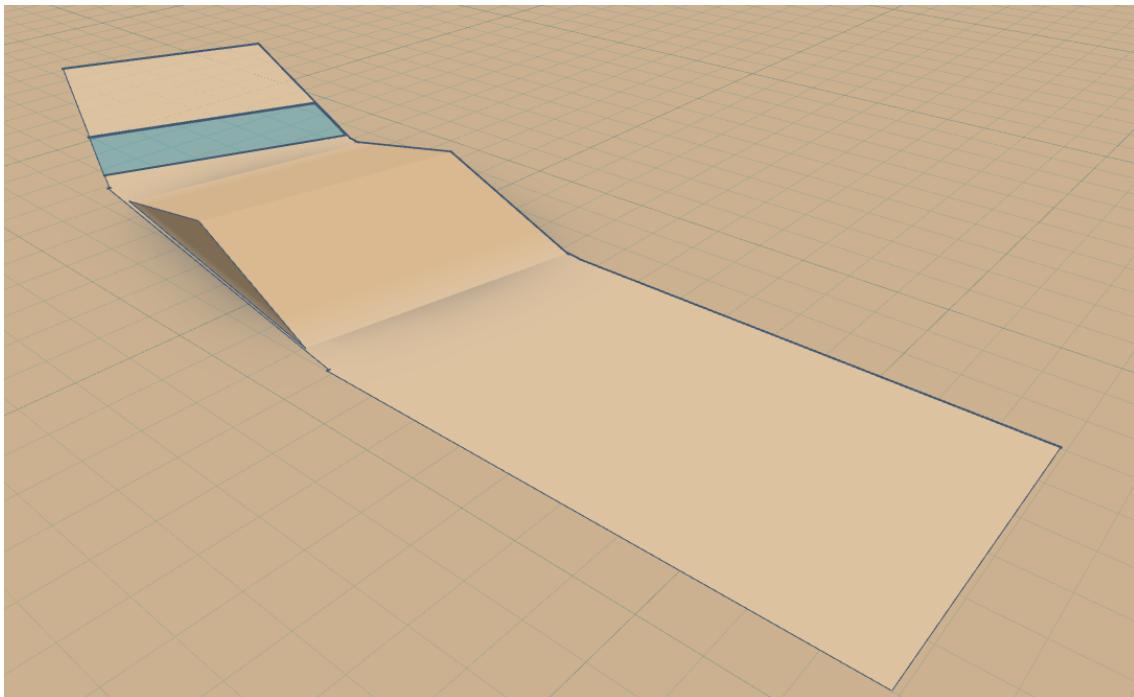
Ilustración 28. Sketch de un charco

---

## CONCATENANDO OBSTÁCULOS

Los obstáculos se pueden concatenar.

Pongamos un ejemplo y cómo este supondría una toma de decisión para el jugador: en el *sketch* mostrado a continuación se observa un bache seguido de un charco en una curva poco cerrada:



**Ilustración 29. Sketch de un bache seguido de un charco en una curva poco cerrada**

En este caso, salvo que el coche esté altamente dañado, quizás lo más interesante sería ejecutar la curva de manera agresiva, aprovechando el salto tras la rampa inicial del bache para evitar el charco y así no perder velocidad. Ahora bien, esto puede aplicar solo en caso de que realmente estemos dispuestos a intercambiar daño por un mejor tiempo en esta sección: quizás más adelante hay otro obstáculo que va a deteriorar el coche significativamente y se debe reconsiderar por tanto la estrategia aquí.

### 5.3.3 GESTIÓN DE RECURSOS: COMBUSTIBLE Y DAÑO SUFRIDO

Uno de los pilares fundamentales del *gameplay* de *Rally Team Tactics* es el uso adecuado de los recursos provistos al inicio de cada prueba al jugador. Estos son el **combustible en el tanque** y los **daños sufridos**.

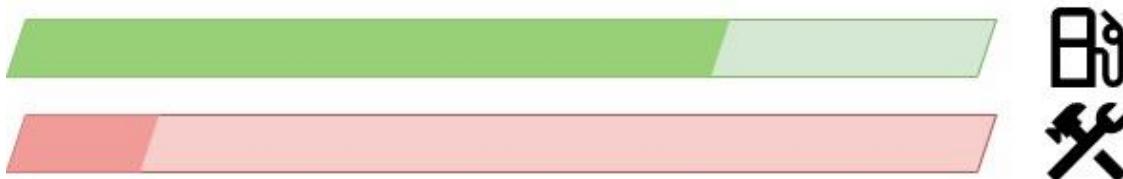


Ilustración 30. Sketch de las barras de combustible y daño

Son **representados en el HUD como barras horizontales o verticales**, según la preferencia del diseñador de UI y UX.

Al comenzar o reiniciar una prueba diaria (es decir, una fase -un tramo concreto o la etapa final-), el combustible del vehículo está siempre al 100%, mientras que el daño comienza al 0%.

#### CONSUMO DE RECURSOS

Se va a realizar seguidamente un análisis del diseño de la consumición de los recursos antes descritos: combustible y daño.

#### COMBUSTIBLE

Recordemos que comenzamos con 100 unidades de combustible (100%). El combustible se gasta de manera constante si se mantiene la velocidad, a un ritmo determinado por los programadores que llamaremos  $c$  y que se mide en unidades por segundo (uds/s).

- Durante el frenado, se consume  $1,5c$ .
- Durante la aceleración ligera, se consume  $1,5c$ .
- Durante la aceleración, se consume  $2c$ .
- Si el coche supera una determinada velocidad, se consume  $1,5c$ .

Estos ritmos están sujetos a cambios cuando se lleven a la implementación en el motor de juegos, deben estar balanceados y variar de manera consistente.

#### DAÑOS

El daño inicia en 0 unidades de daño (0%). El daño se produce siempre que el coche colisione con el terreno, ya sea horizontal o verticalmente. Por ejemplo, tras caer por un salto, según la altura desde la que se caiga, se sufriría un determinado número de unidades de daño. No se definen ahora valores exactos, pues se requerirá de pruebas en el motor de juegos para decidir qué valores son más apropiados.

## QUEDARSE SIN RECURSOS

El consumo de todo el combustible o sufrir un daño irreparable supondrían (alcanzar el 0% o el 100%, respectivamente) la detención del coche y forzarían a reiniciar la prueba o salir de esta.

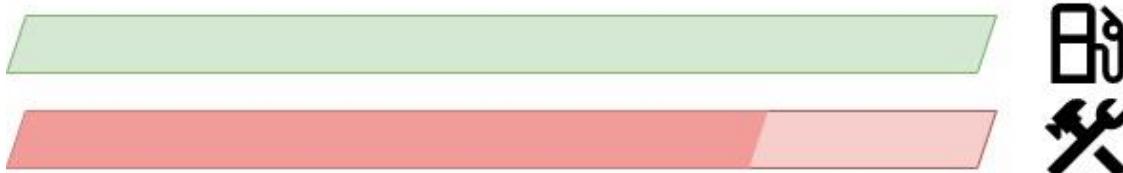


Ilustración 31. Sketch de los recursos: sin combustible

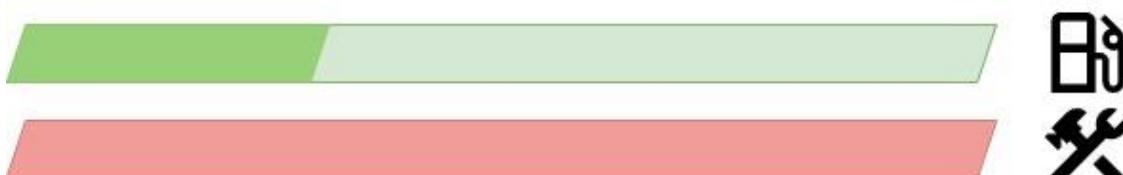


Ilustración 32. Sketch de los recursos: coche destrozado

## RECURSOS ENTRE FASES

Como el coche se repara y recarga gasolina siempre que terminamos una prueba, se debe mostrar al jugador encima de cada día (o fase), cuánto daño sufrió y cuánto combustible consumió en su último intento guardado.

Esto permite al jugador considerar si su desempeño en dicho tramo es apropiado para enfrentarse a la etapa completa. Recordemos que la etapa completa incluye todos los tramos de los días anteriores, y como se va a disponer de la misma cantidad de combustible y la misma resistencia que en los tramos separados, el jugador debe aprender no solo obtener buenos tiempos en los tramos, sino a hacerlo con un consumo de combustible y daños que sea proporcional a su dificultad y que lleve a una gestión óptima de recursos en la fase final.

## 5.4 CONTROLES Y PERIFÉRICOS

## 6 PROGRESIÓN

### 6.1 OBJETIVOS DEL JUGADOR

El objetivo del jugador es conseguir el mejor tiempo en cada etapa de *rally*, que presenta un trazado distinto y una ambientación única. Como se ha explicado con detenimiento en apartados anteriores, para lograr dicho tiempo, el jugador deberá ser capaz de seleccionar las instrucciones y el temperamento en cada sección que produzcan los mejores resultados.

Gestionar los recursos adecuadamente es crucial para llegar a la meta en primer lugar, por tanto, un manejo inteligente de estos puede dar mejores resultados.

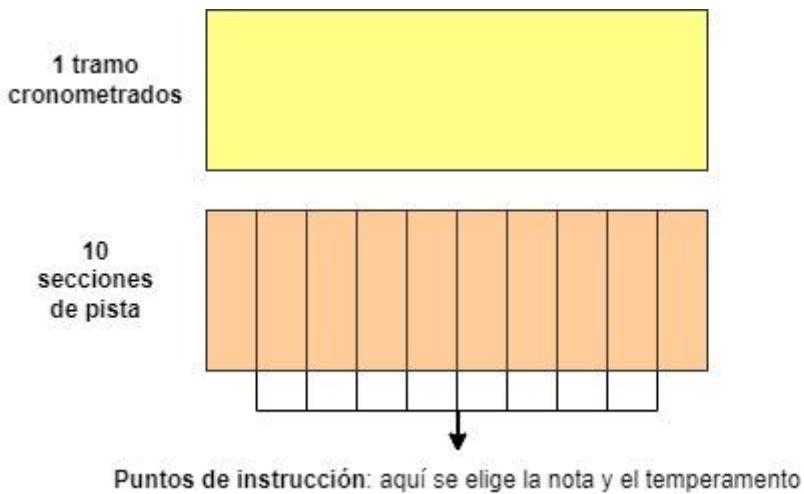
### 6.2 ESTRUCTURA DEL JUEGO

El juego se divide en varios **eventos**, pero el progreso no es lineal, es decir, se pueden completar en el orden que se desee. La etapa de cada evento presenta su propio itinerario y características, no necesariamente suponiendo un incremento o decremento de dificultad. Son simplemente distintas pruebas que se pueden jugar en cualquier orden en función de la preferencia del jugador.

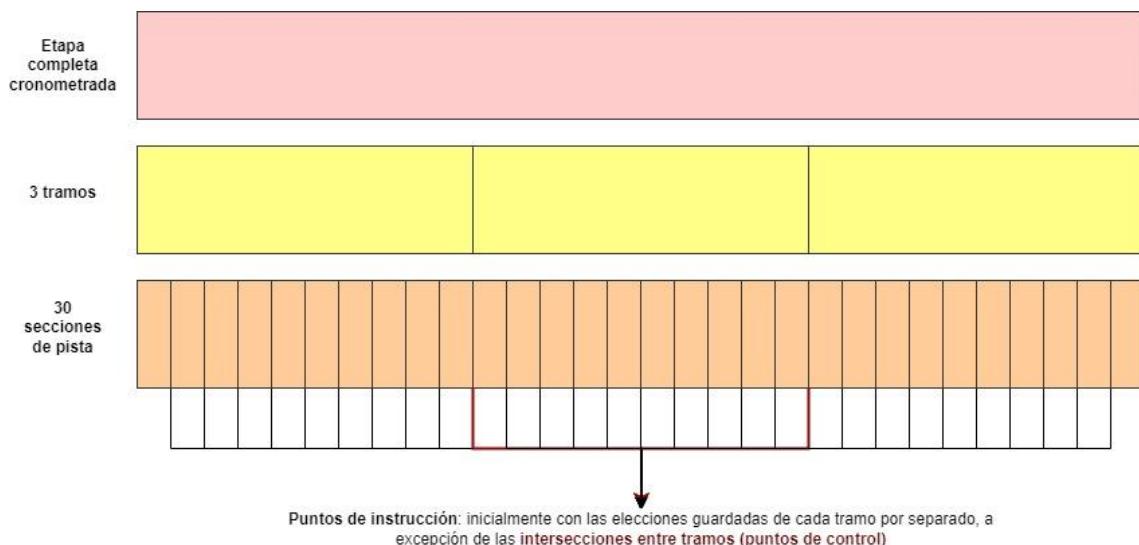
Para preparar al jugador de cara a una **etapa completa**, que se compone de **tres tramos cronometrados**, cada evento se divide **cuatro días ficticios o fases**:

- Los tres primeros días ponen al jugador en cada uno de los tramos en orden de aparición en la etapa. Una vez que se completa un día correspondiente a un tramo, se guarda el resultado en tiempo y recursos y se desbloquea el siguiente día si no ha sido desbloqueado ya. Se pueden intentar estos tramos repetidas veces antes o después de la etapa final, son pruebas preparatorias. Los tramos se componen de **diez secciones de pista**.
- Cuando se comienza la etapa completa el último día, los puntos de control estarán ya por defecto elegidos tal y como quedaron en los tramos por separado (cuando se intentaron por última vez), a excepción de las intersecciones entre tramos, que deben ser seleccionadas.

Los siguientes esquemas muestran la estructura de cada día o fase de tramo cronometrado y de un día o fase de etapa completa cronometrada:



**Ilustración 33. Estructura de un tramo cronometrado (días 1 a 3 -fases 1, 2 y 3-)**



**Ilustración 34. Estructura de una etapa completa cronometrada (último día -fase 4-)**

A continuación, se presentan varios esquemas que ilustran la estructura del juego y de cada evento:

### Estructura del juego

Supongamos que se ha completado C, se debe hacer visible al jugador. En todos los casos, se muestra el porcentaje de completitud, se podría reforzar el número con el borde del selector.



Ilustración 35. Estructura del juego

### Estructura de un evento

Solo se puede acceder al día siguiente si se ha completado el anterior. Una vez completado un día, se guarda el tiempo, combustible gastado y daño sufrido en este. Se mostrará entonces, y cambiará cada vez que se vuelve a intentar, con el último resultado. En el caso de la etapa completa, se mostrará siempre el mejor resultado al último

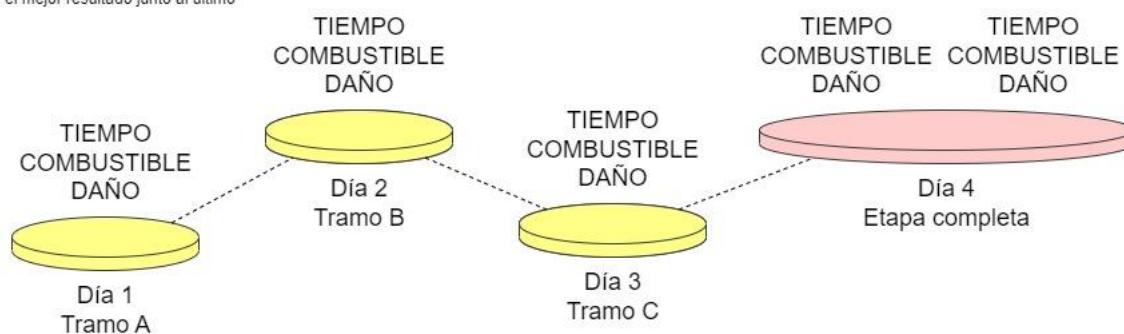


Ilustración 36. Estructura de un evento

## 6.3 CLASIFICACIÓN EN LÍNEA

## 7 INTERFAZ DE USUARIO

### 7.1 REQUISITOS DE LA INTERFAZ

El siguiente esquema muestra los requisitos que se han considerado para el diseño y flujo de la interfaz de usuario:

#### Interfaz Gráfica Amigable

- La interfaz debe ser colorida, amigable y atractiva para niños, con personajes y elementos visuales que les resulten llamativos

#### Tutorial Interactivo

- Incluir un tutorial intuitivo que enseñe los conceptos básicos

#### Feedback Inmediato

- Proporcionar retroalimentación instantánea cuando los niños tomen decisiones o completen tareas, reforzando positivamente el aprendizaje

#### Indicadores de Progreso

- Mostrar un medidor de progreso o puntuación para que los niños puedan ver su avance en el juego

#### Incentivos y Recompensas:

- Ofrecer recompensas, medallas o logros por completar tareas o niveles, fomentando la motivación y el compromiso

#### Sistema de Pistas

- Proporcionar pistas o sugerencias cuando los niños se enfrenten a desafíos difíciles, sin revelar la solución completa

#### Interfaz Táctil o Controlador Simple

- El juego debe ser fácil de controlar con un toque en pantalla o con controles simples que los niños puedan entender fácilmente

#### Instrucciones Visuales y de Voz

- Si es posible, incluir instrucciones visuales y de voz para ayudar a los niños en la comprensión de las tareas y conceptos

#### Registro de Actividad

- Mantener un registro de las acciones y decisiones de los niños para que los padres o maestros puedan revisar el progreso y el aprendizaje

#### Ajustes de Dificultad

- Permitir a los padres o tutores ajustar la dificultad del juego según la edad y habilidades del niño

#### Compatibilidad Multiplataforma

- Hacer que el juego esté disponible en varias plataformas (Web, PC, tabletas) para una mayor accesibilidad. Hay que tener en cuenta el aspect ratio y las características de cada una

#### Música y Sonidos Atractivos

- Incluir música y efectos de sonido que sean agradables y reconocibles, que permitan reconocer patrones

## 7.2 DIAGRAMA DE FLUJO DE NAVEGACIÓN

Seguidamente se presentan diagramas de flujo referentes a la navegación por los menús y pantallas:

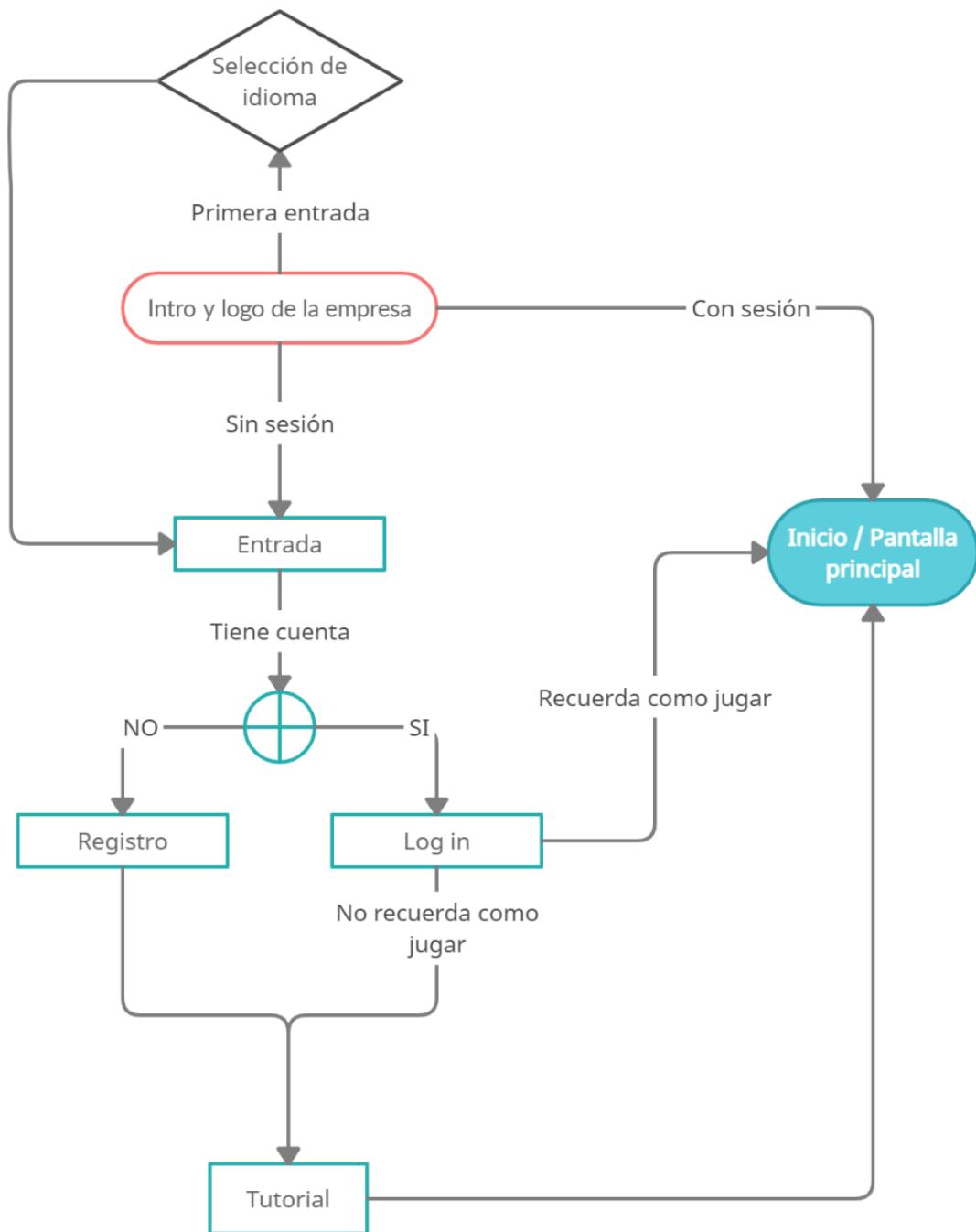


Ilustración 37. Diagrama de navegación de inicio del juego

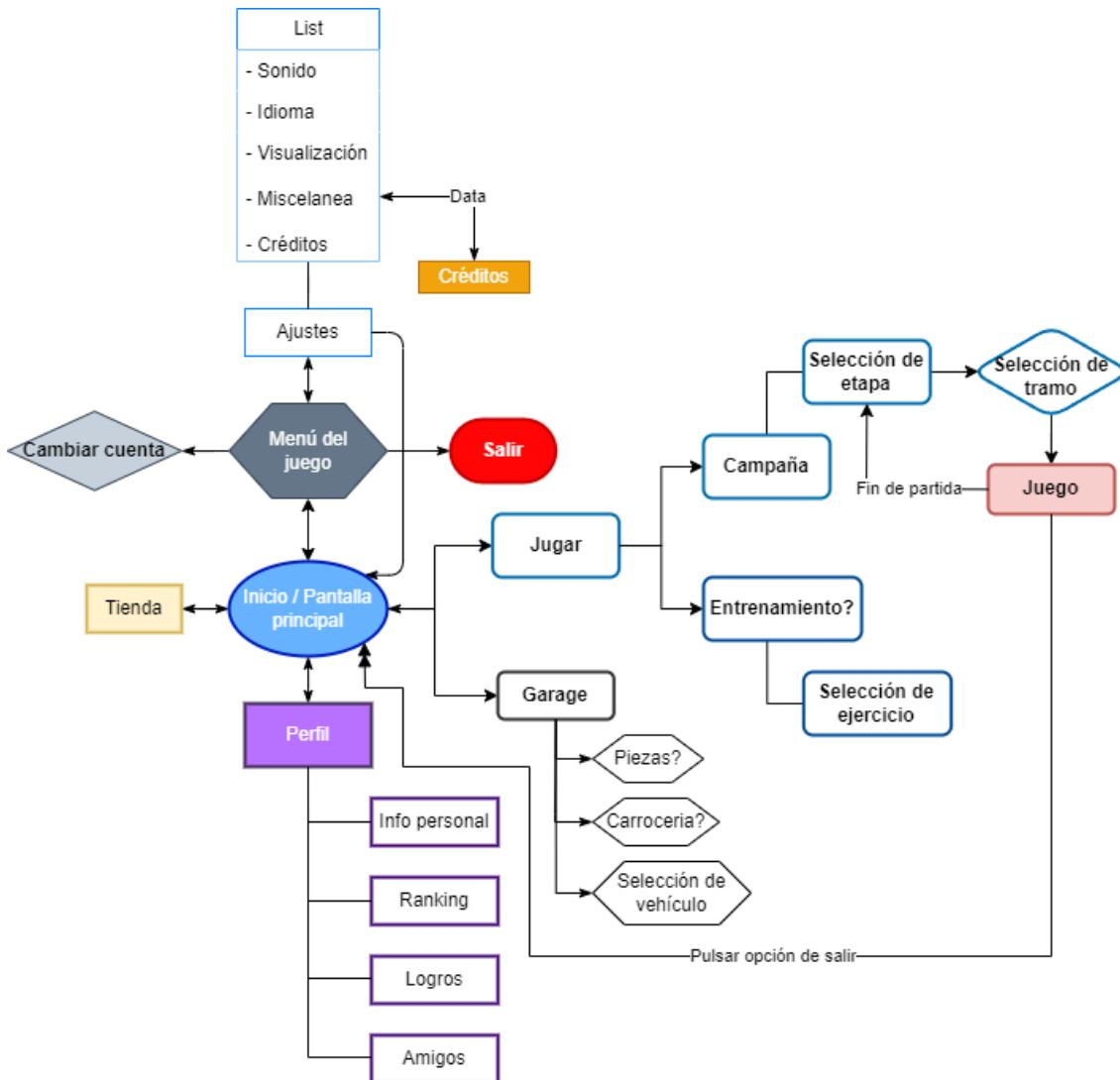


Ilustración 38. Diagrama de navegación: menús y juego

En el diagrama superior se presentan algunas ramas marcadas con interrogación, estas son aquellas que se han planteado como futuras adiciones al juego que en principio no estarán en las versiones preliminares.

### 7.3 DISEÑO ESQUEMÁTICO DE LA UI

En este apartado se muestran *mockups* de las pantallas, menús y pop-ups que se han diseñado para el prototipo (no incluyen la parte in-game). El formato de estas es para juego en navegador:



Ilustración 39. Diseño de la Splash Screen con el logo de la empresa



Ilustración 40. Diseño de la selección de idioma (primera vez que se inicia el juego)



Ilustración 41. Diseño de la pantalla de inicio

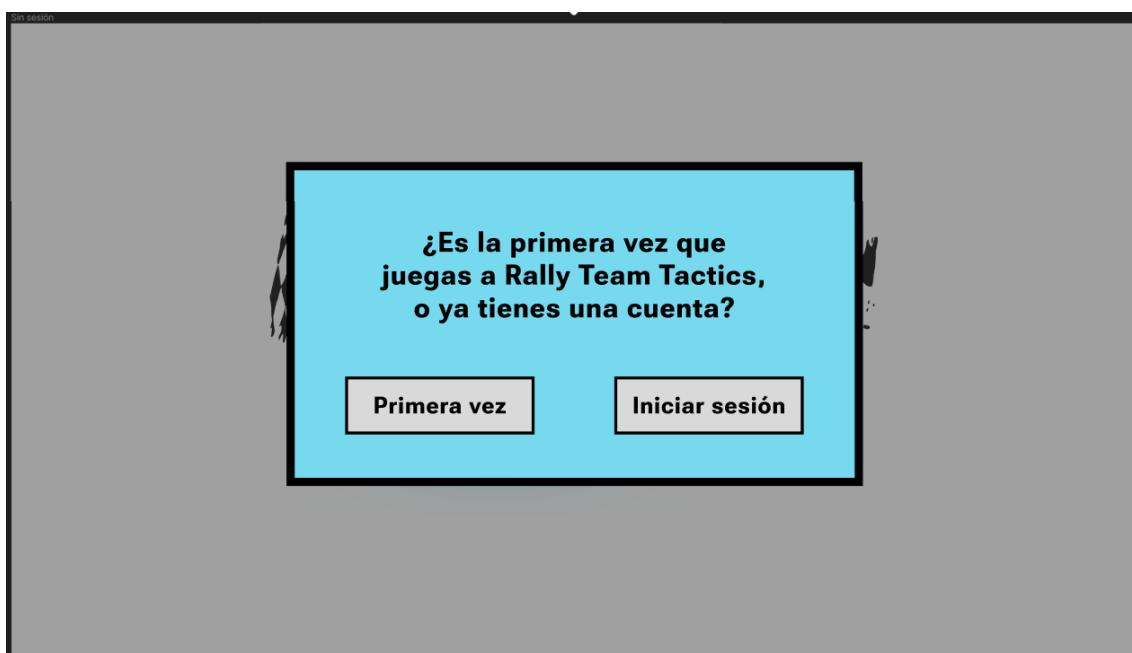


Ilustración 42. Diseño del pop-up de inicio de sesión, si no se ha iniciado previamente en el dispositivo

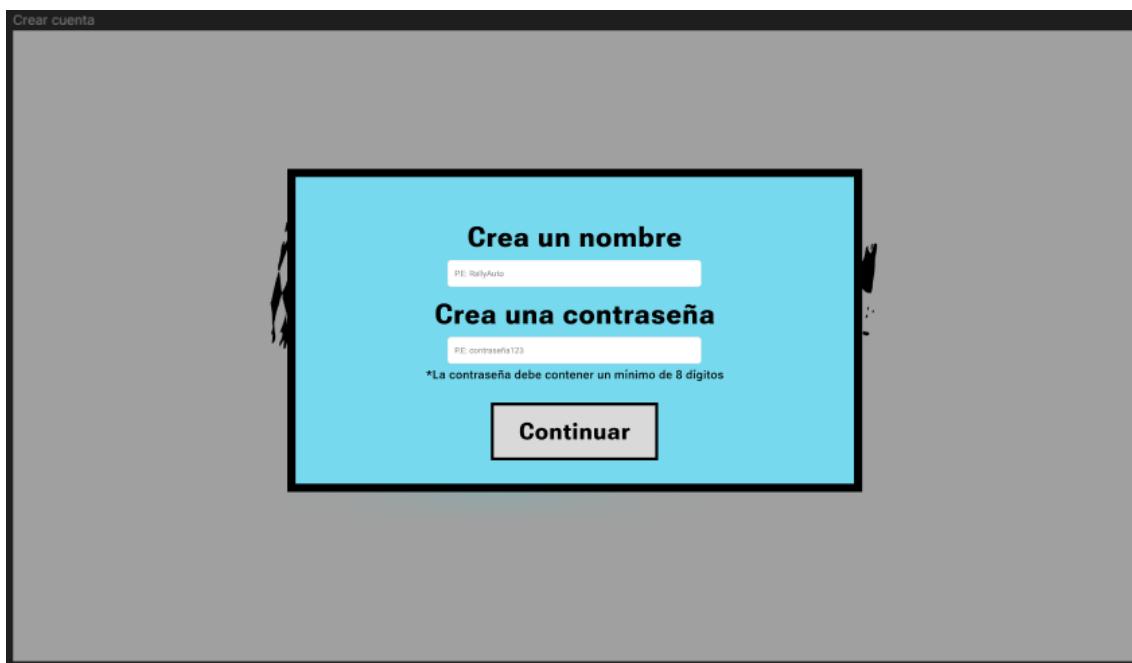


Ilustración 43. Diseño del pop-up de registro

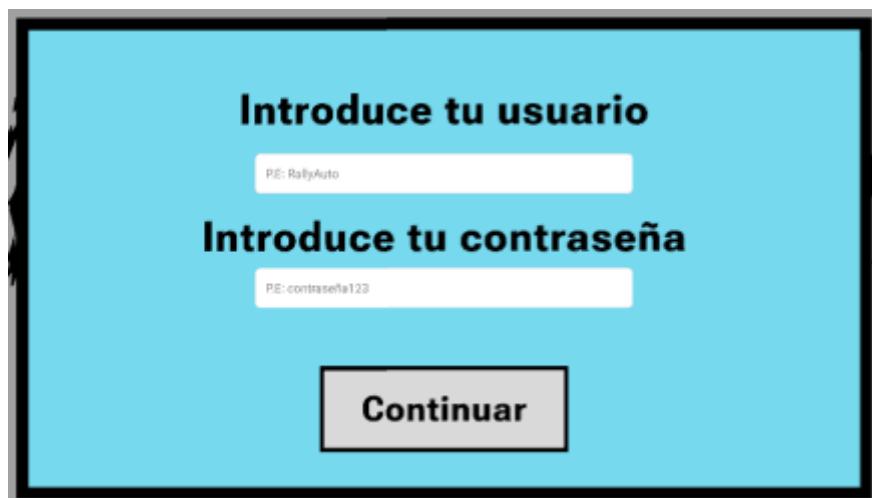


Ilustración 44. Diseño del pop-up de inicio de sesión

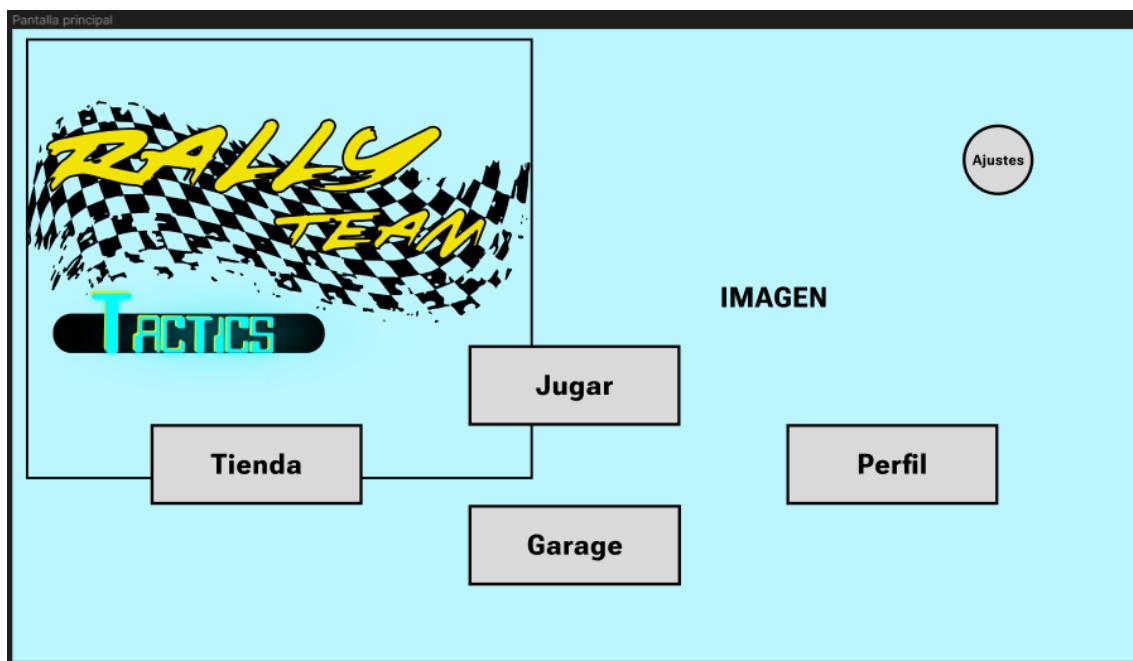


Ilustración 45. Diseño del menú principal

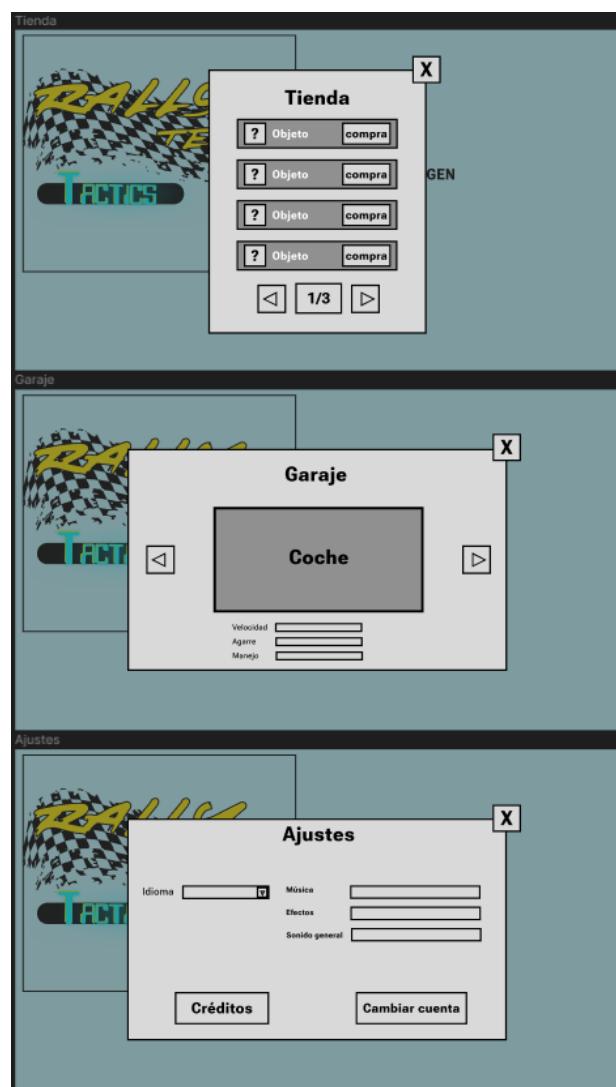


Ilustración 46. Diseños de los pop-ups de tienda, garaje y ajustes, respectivamente

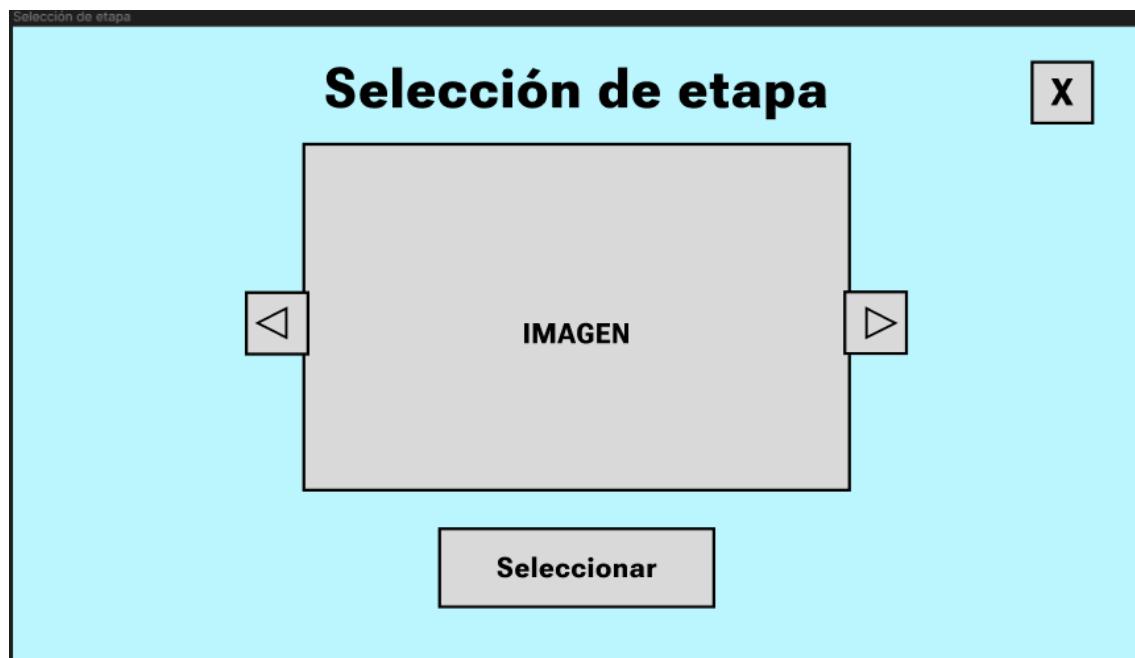


Ilustración 47. Diseño de la pantalla de selección de etapa

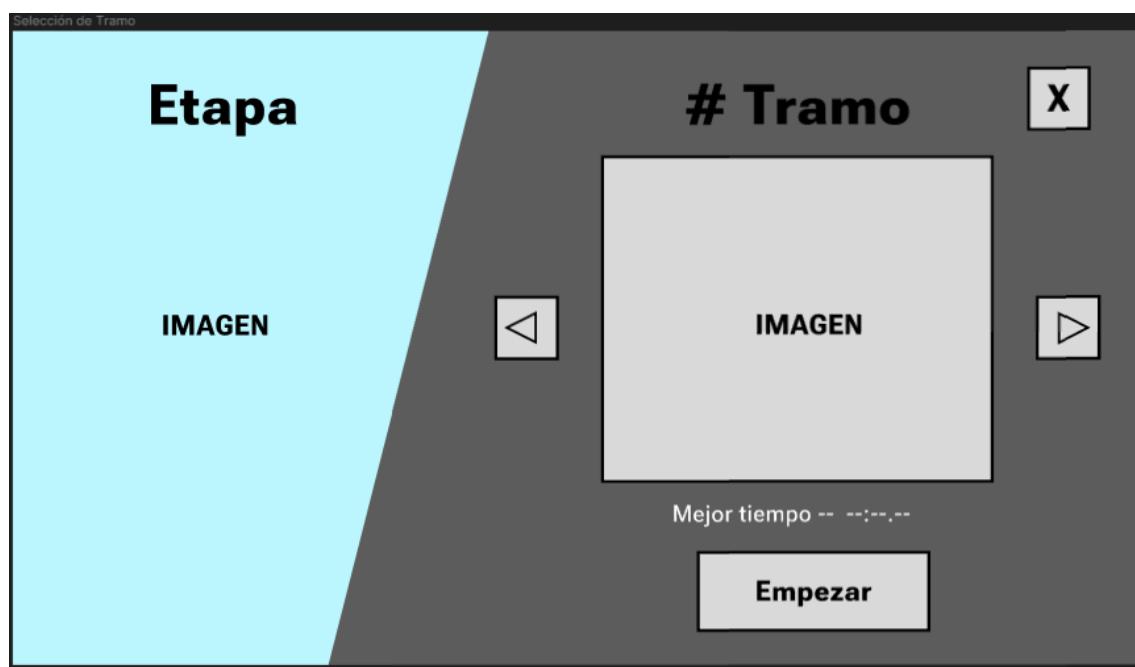


Ilustración 48. Diseño de la pantalla de selección de tramo en la etapa

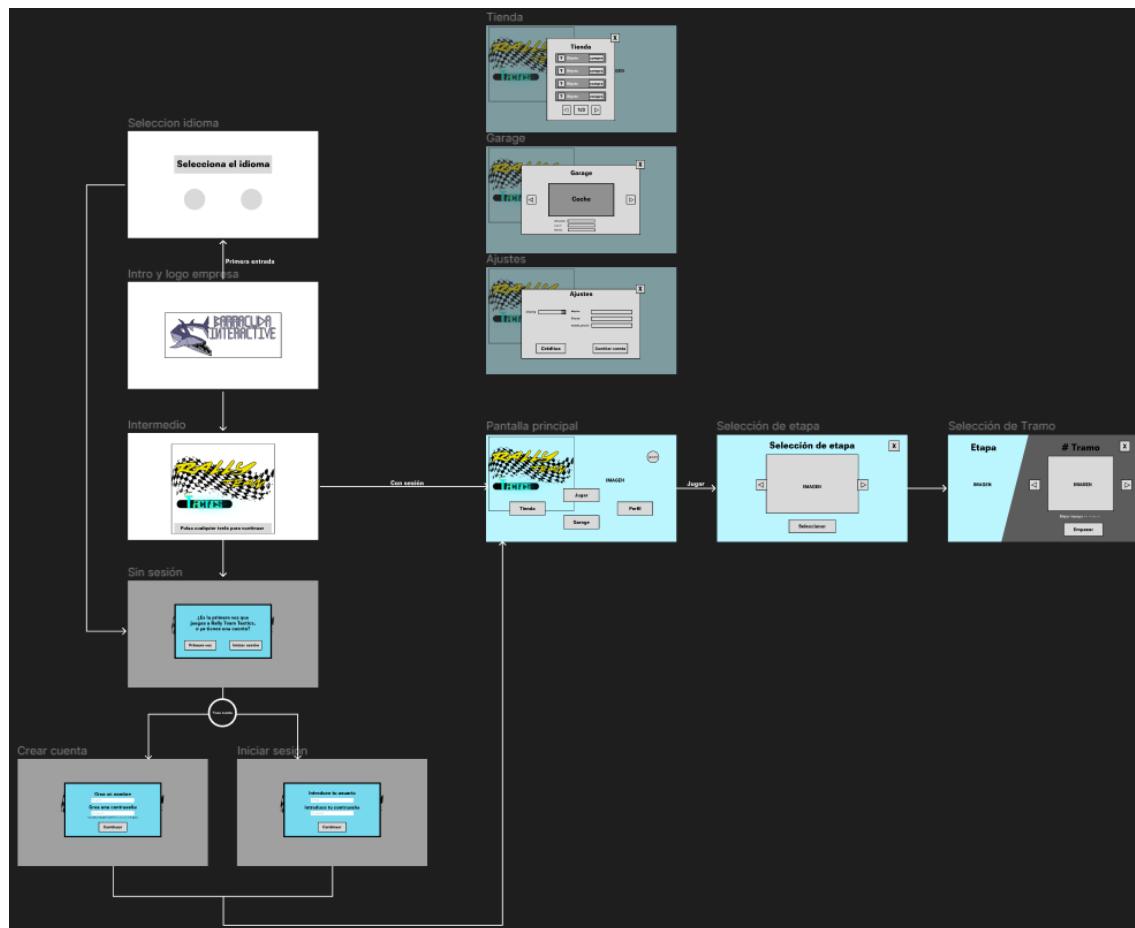


Ilustración 49. Representación del flujo de navegación en Figma

En la imagen superior, se proporciona una representación gráfica del diagrama de flujo de las pantallas con el fin de facilitar la comprensión de la navegación por los menús. Más abajo se incluye un enlace al proyecto en *Figma*, donde se ha recreado el flujo de navegación para previsualizar su funcionamiento.

[https://www.figma.com/file/LILMOV1jsncNQxOc0rWFNt/RTT?type=design&node\\_id=0%3A1&mode=design&t=zzw8euf4DUNr2HJW-1](https://www.figma.com/file/LILMOV1jsncNQxOc0rWFNt/RTT?type=design&node_id=0%3A1&mode=design&t=zzw8euf4DUNr2HJW-1)

## 8 ARTE

### 8.1 ESTÉTICA DEL JUEGO

#### 8.1.1 REFERENCIAS ARTÍSTICAS

Como se explicó en el apartado [2.1 Principales referencias](#), el juego presentará una estética *low-poly* colorida, inspirada en especial por el *arcade racer Ridge Racer* (Namco, 1993) y el *indie* de *rallies* contemporáneo *Art of Rally* (Funselektor Labs, 2020).



Ilustración 50. Ridge Racer (Namco, 1993)



Ilustración 51. Art of Rally (Funselektor Labs, 2020)

### 8.1.2 LENGUAJE VISUAL DEL JUEGO

Para que todo el contenido relacionado con *RTT*, tanto jugable como promocional (y su documentación), siguiesen el mismo estilo visual, se elaboró un documento de estilo de juego. En este se recogen el logo provisional, así como la paleta de colores y la fuente a utilizar.

Este apartado del GDD recoge los aspectos más importantes de del documento.

#### PALETAS DE COLOR

Los colores base del juego son el amarillo #ede318, azul #42f4fb y negro #000000.

A continuación, se muestran dichos colores y seguidamente, unos colores adicionales de acento:

#### COLOR PRINCIPAL



#### SOMBRA DEL COLOR PRINCIPAL



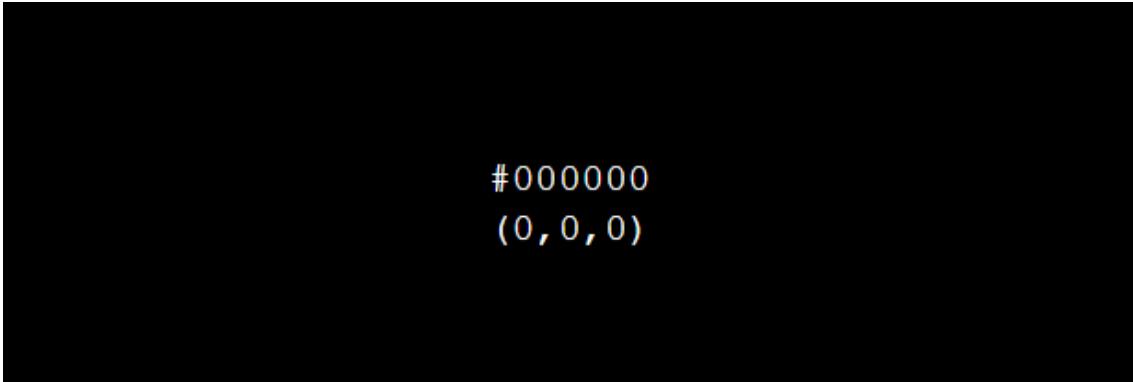
#ede318 #d5cc15 #bdb513 #a59e10 #8e880e #76710c

#### COLOR COMPLEMENTARIO AL COLOR PRINCIPAL



#ede318

#1822ed

**NEGRO**

#000000  
(0, 0, 0)

**COLOR SECUNDARIO**

#42F4FB  
(66, 244, 251)

**SOMBRA DEL COLOR SECUNDARIO**

#42f4fb #3bdbe1 #34c3c8 #2eaaaaf #279296 #217a7d

**COLOR COMPLEMENTARIO AL SECUNDARIO**

#42f4fb

#fb4942

## TIPOGRAFÍAS

Los tamaños se adecuarán según necesidades.

### MUBENK REGULAR

<https://www.behance.net/gallery/100102693/Mubenk-Brush-Font-Free-Download>



Ilustración 52. Muestra de mubenk regular

### CHECKBOOK REGULAR

<https://www.1001freefonts.com/es/checkbook.font>

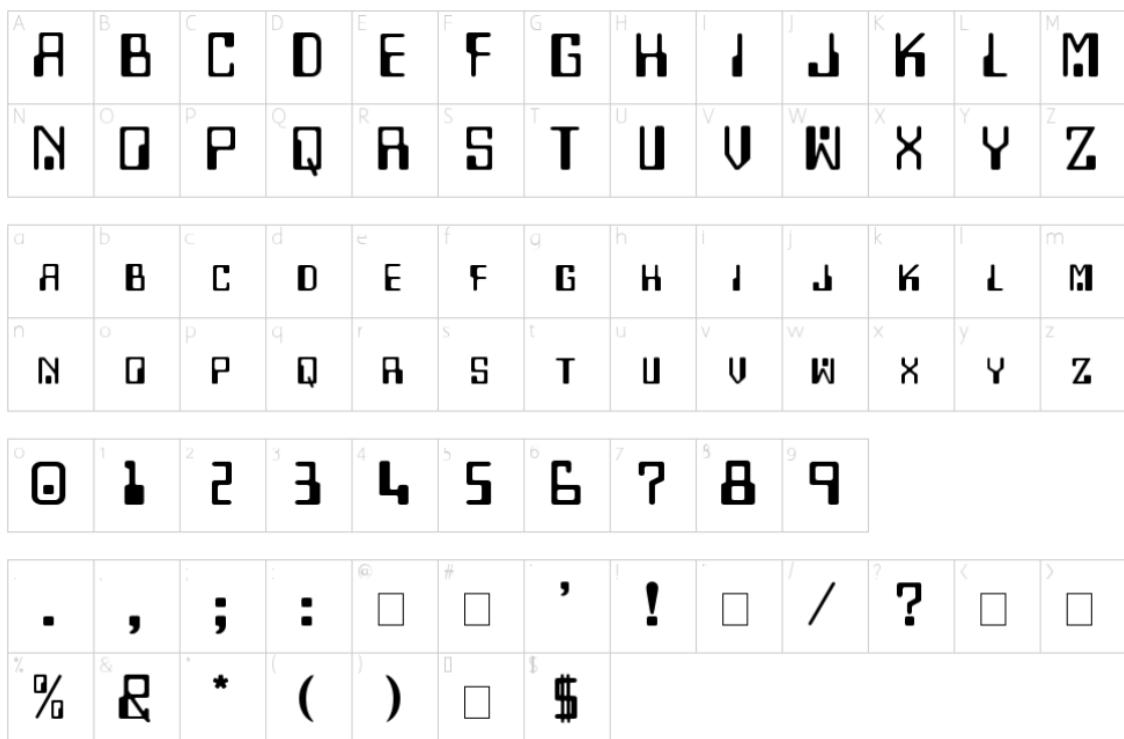


Ilustración 53. Mapa de caracteres de checkbook regular

---

### LOGO PROVISIONAL DEL JUEGO

El logo provisional de *RTT* utiliza las tipografías *mubenk* y *checkbook* y presenta principalmente dos colores, amarillo y cian, cada uno en una palabra del título. El negro se usa para detalles como los bordes.



Ilustración 54. Logo provisional del juego

---

### FONDOS

Se utilizará un fondo amarillo más oscuro al que se le pueden añadir detalles en negro como degradados o el patrón de bandera de carreras.



Ilustración 55. Diseño de posible fondo

## 8.2 CONCEPT

Se presentan en este apartado una serie de *concepts* de elementos del juego, como el coche o los *props* de los escenarios.

### 8.2.1 CONCEPTS DEL COCHE

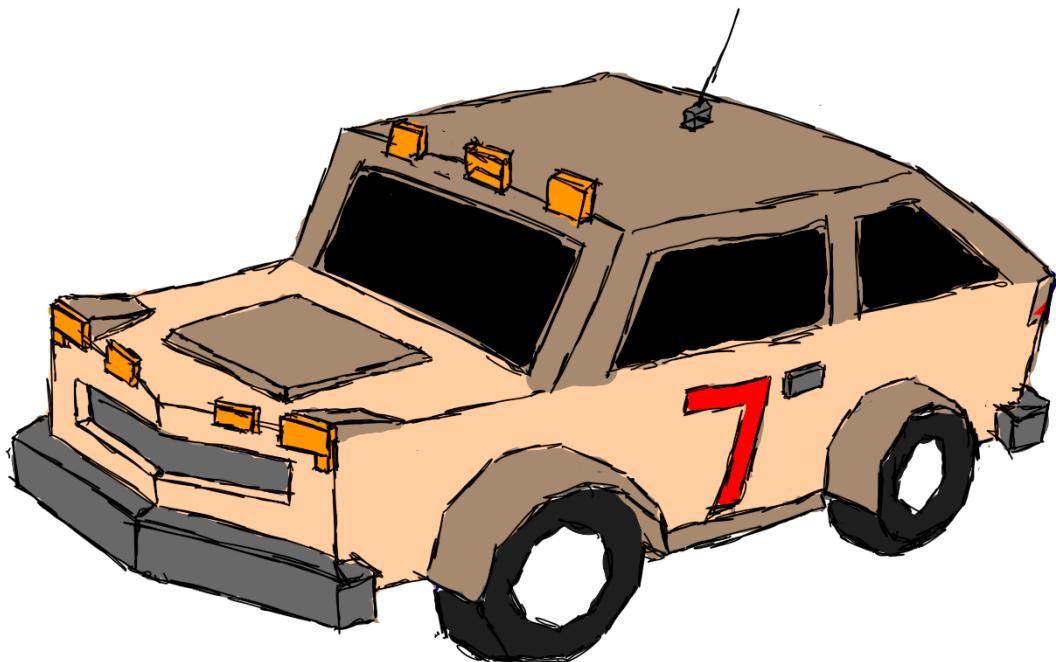


Ilustración 56. Concept de coche en perspectiva



Ilustración 57. Concept de coche de perfil

---

#### 8.2.2 CONCEPTS DE PROPS

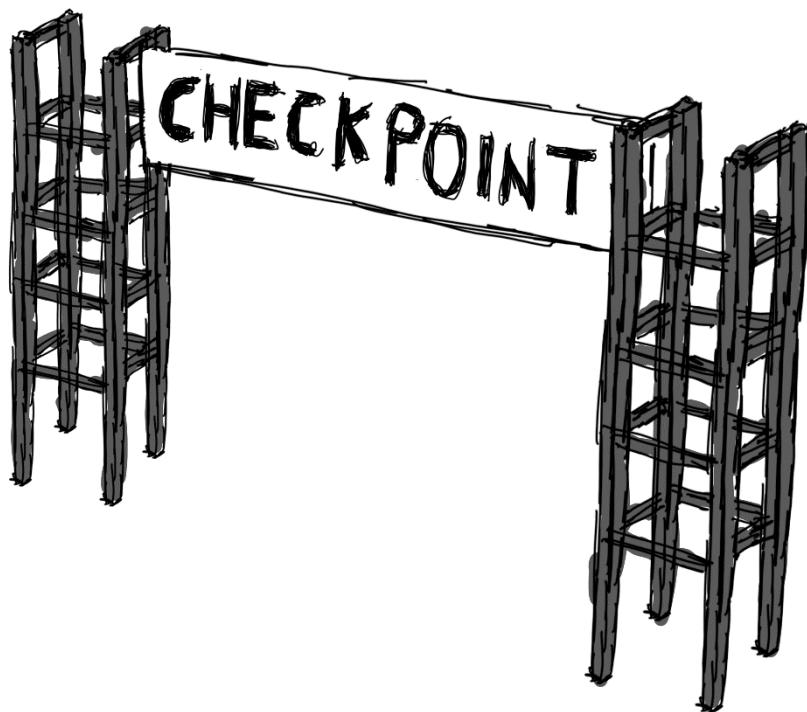


Ilustración 58. Concept de un punto de control

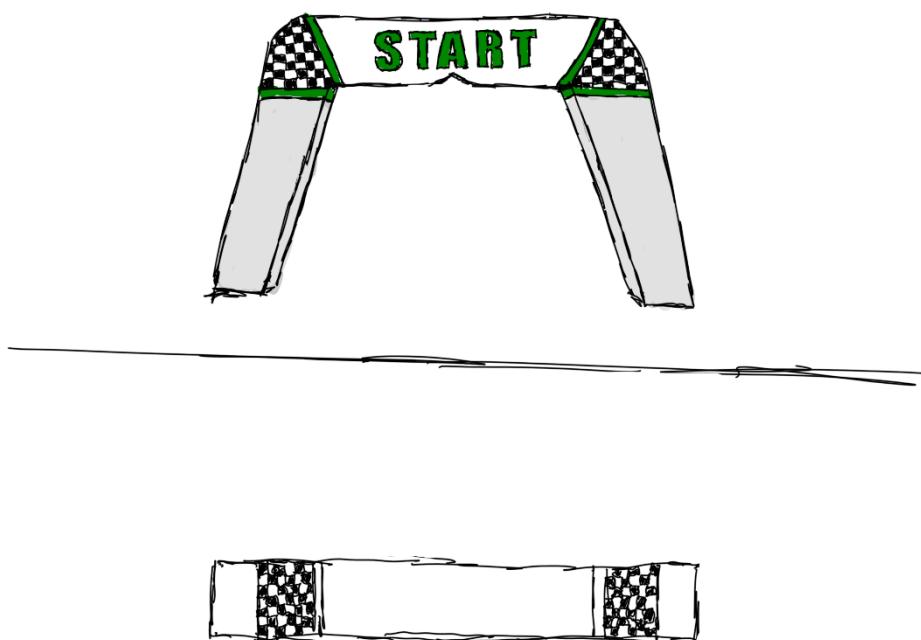


Ilustración 59. Concept de la salida

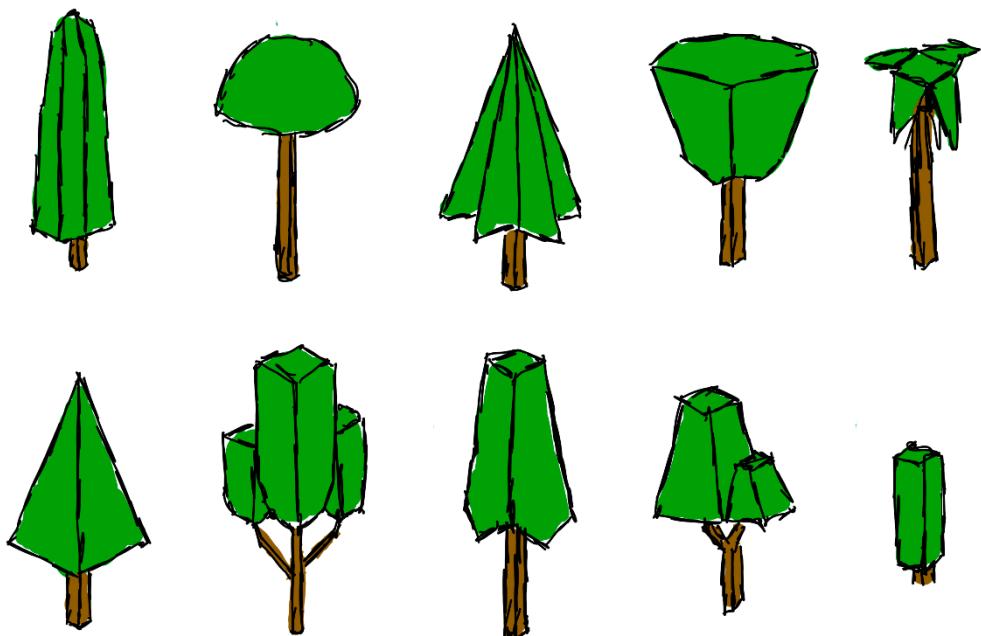


Ilustración 60. Concept de diferentes variantes de árbol

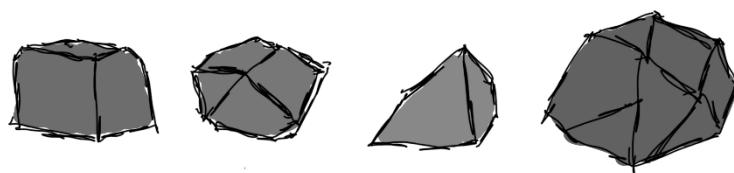
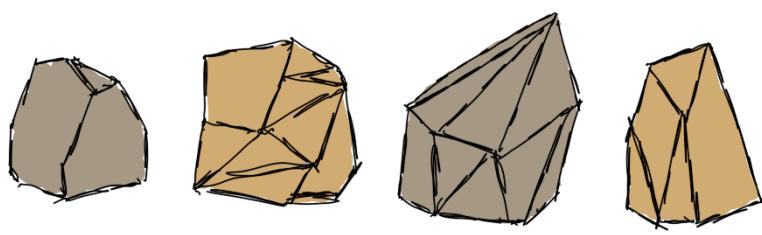


Ilustración 61. Concept de diferentes rocas

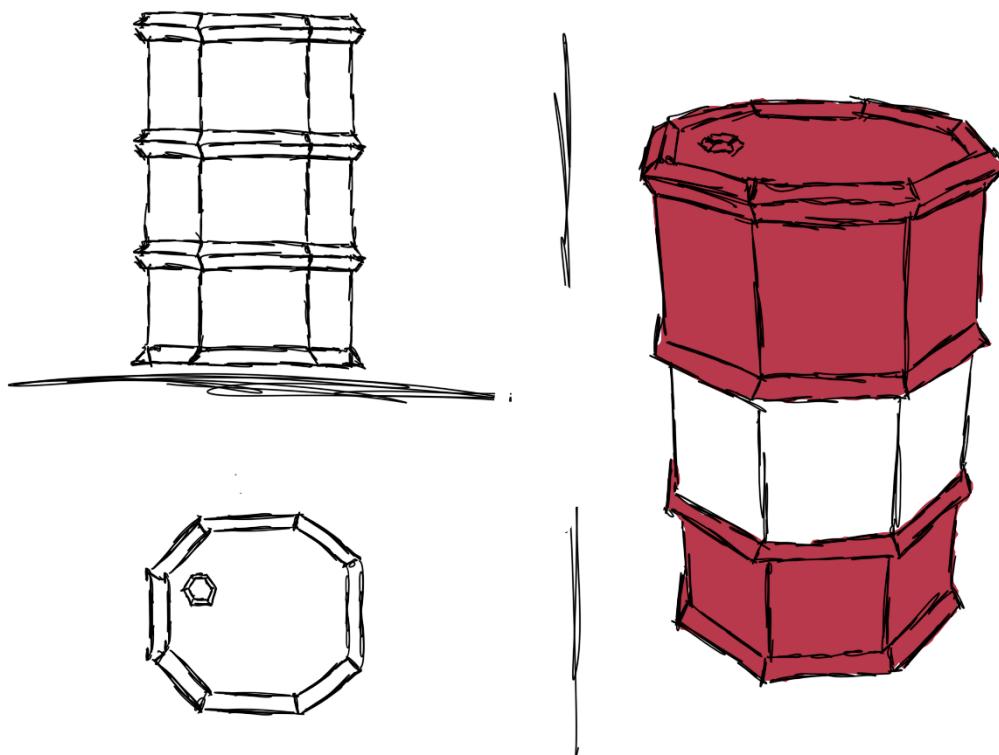


Ilustración 62. Concept de un barril

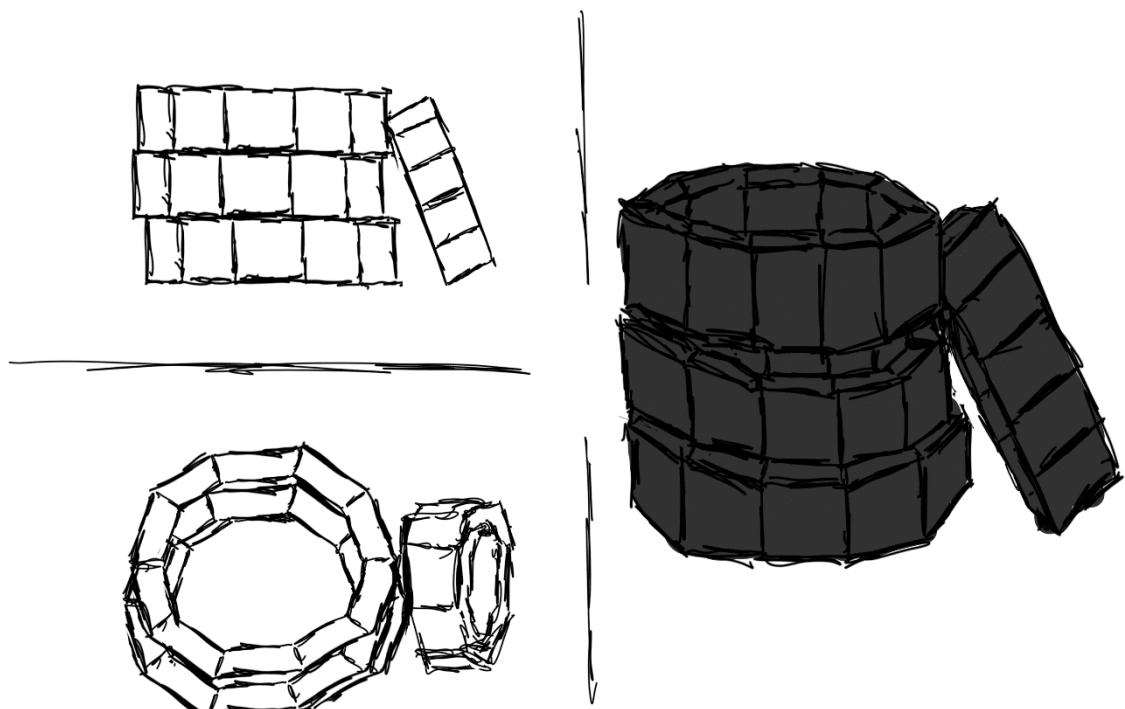


Ilustración 63. Concept de una pila de ruedas

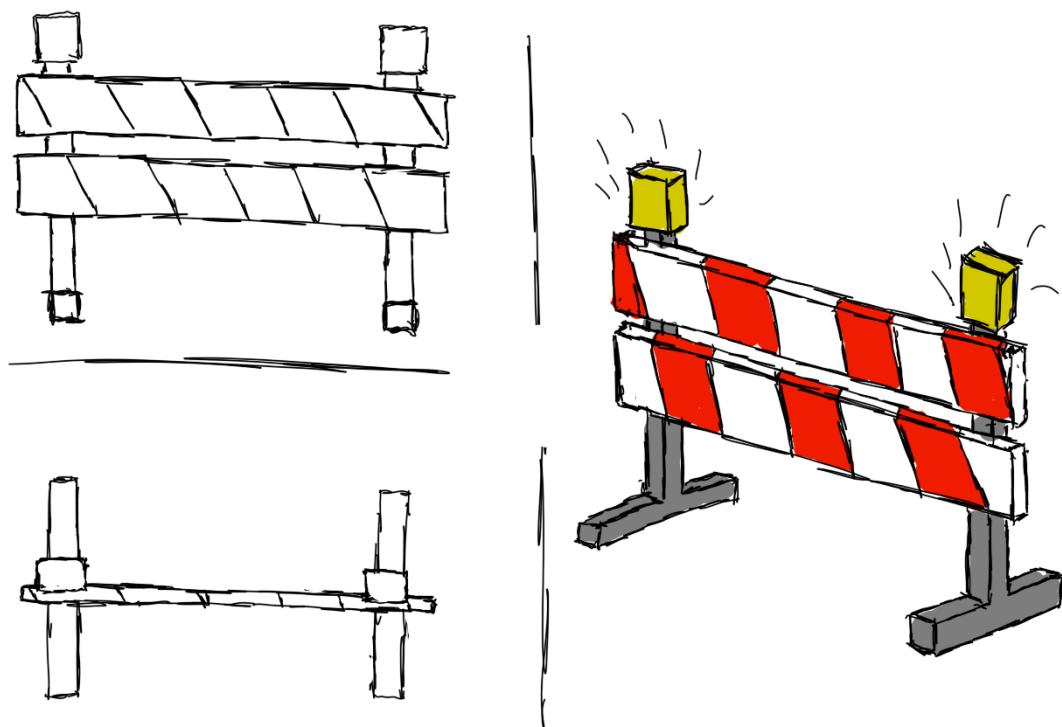


Ilustración 64. Concept de una valla de obra

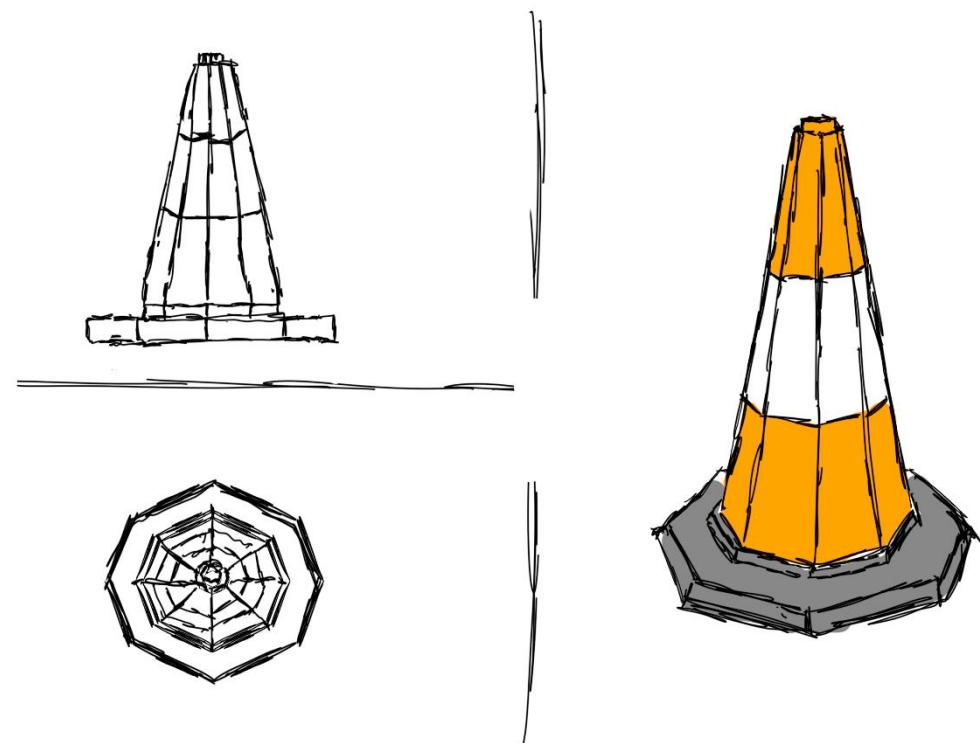


Ilustración 65. Concepto de un cono de tráfico

## 8.3 DISEÑO VISUAL DE LA UI

## 8.3 ASSETS 3D

## 9 SONIDO

### 9.1 EFECTOS DE SONIDO

## 9.2 MÚSICA

## 10 PRODUCCIÓN

### 10.1 MODELO DE NEGOCIO DEL JUEGO Y PLAN DE FINANCIACIÓN

#### 10.1.1 INFORMACIÓN SOBRE EL USUARIO

¿Quién es?	<ul style="list-style-type: none"> <li>• Existen dos tipos de usuarios, un primer usuario-cliente que sería una entidad financiera/educativa, que obtiene datos del usuario final mediante analíticas, y un usuario final, que es el niño que juega al videojuego y genera las métricas.</li> </ul>
¿Qué quiere?	<ul style="list-style-type: none"> <li>• Por parte del cliente-usuario, quiere obtener las métricas y potenciar el pensamiento computacional de los niños.</li> <li>• Por parte del usuario final, quiere diversión y entretenimiento sin tener la sensación de estar haciendo una actividad académica.</li> </ul>
¿Aficiones?	<ul style="list-style-type: none"> <li>• El usuario final es un niño de edad temprana, entre los 8 y 12 años, que frecuenta videojuegos de corte actual, de todos los géneros. Tiene nociones y se asume que comprende el funcionamiento-mecánicas del medio audiovisual.</li> </ul>
¿Situación?	<ul style="list-style-type: none"> <li>• Por parte del cliente-usuario, su situación es de dependencia con instituciones públicas/privadas para poder financiar proyectos innovadores como el estudio del Pensamiento Computacional en centros educativos.</li> <li>• Por parte del usuario final, su situación es la de adquisición habilidades formativas a través de innovaciones que sustituyen la educación tradicional. Esta gamificación camufla la docencia en sí misma.</li> </ul>
¿Actividad?	<ul style="list-style-type: none"> <li>• Por parte del cliente-usuario, su actividad son labores educativas, administrativas y corporativas.</li> <li>• Por parte del usuario final, su actividad es la asistencia y participación en las clases como alumnos del centro educativo.</li> </ul>
¿Cómo es?	<ul style="list-style-type: none"> <li>• Por parte del cliente-usuario, es una entidad educativa compuesta por expertos en docencia y que busca encontrar métodos más efectivos y vanguardistas para generar recursos y/o enfoques académicos.</li> <li>• Por parte del usuario final, es una persona con energía que necesita reposo entre clases, que aprende mejor a través de metodologías lúdicas que tradicionales.</li> </ul>
¿Qué necesita?	<ul style="list-style-type: none"> <li>• El cliente-usuario necesita conocer las capacidades de sus estudiantes de cara a mejorar su método de enseñanza. Al conocer sus aptitudes, la de los usuarios finales, podrá dirigir mejor sus esfuerzos a la hora de personalizar la educación, plan académico, etcétera. Asimismo, la entidad educativa mejorará su posición de prestigio al ofrecer mejores programas de enseñanza.</li> </ul>

### 10.1.2 MAPA DE EMPATÍA

Se muestra a continuación el Mapa de Empatía del usuario final, niños y niñas entre 8 y 12 años:

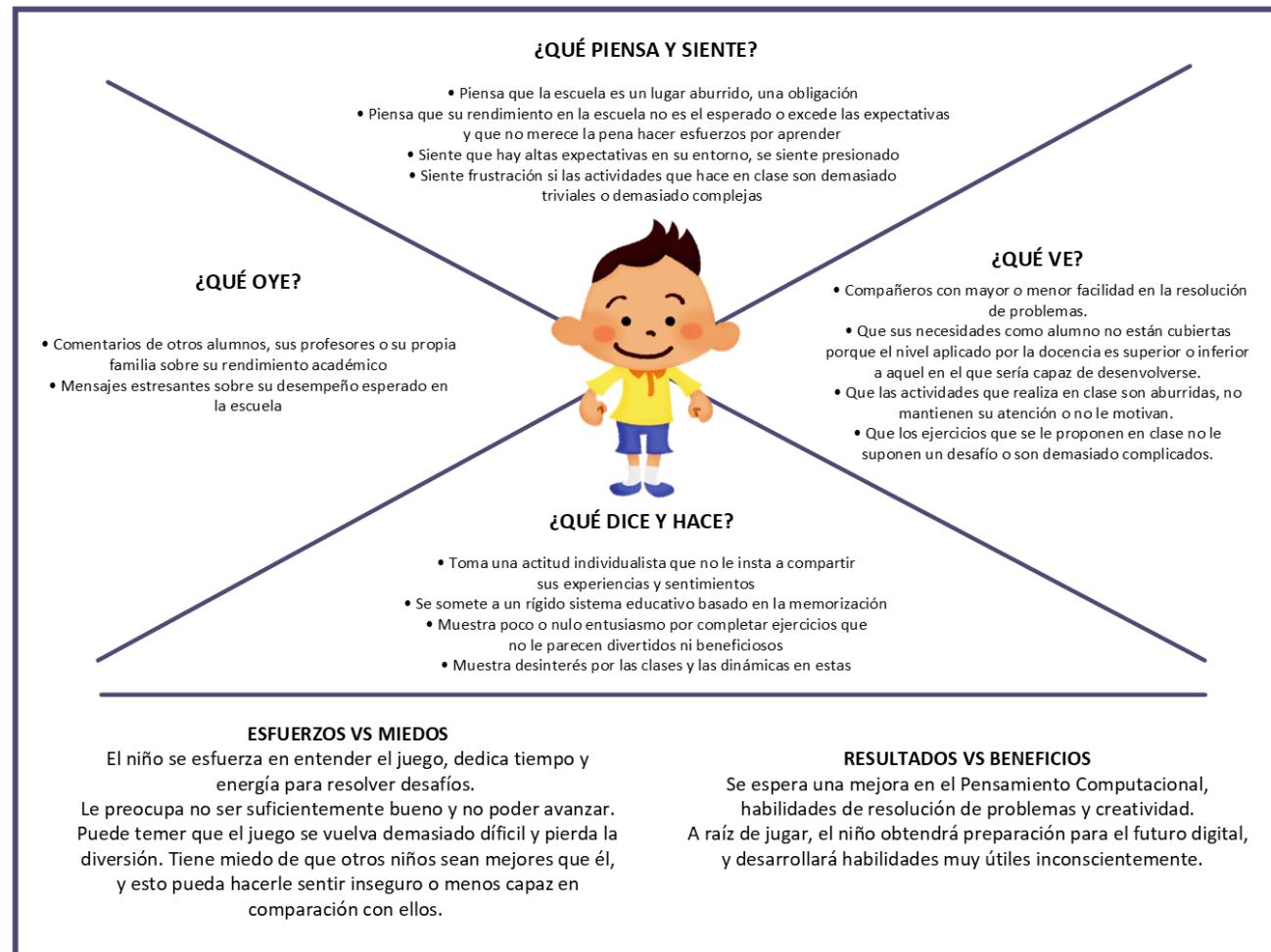


Ilustración 66. Mapa de Empatía del usuario final

### 10.1.3 CAJA DE HERRAMIENTAS

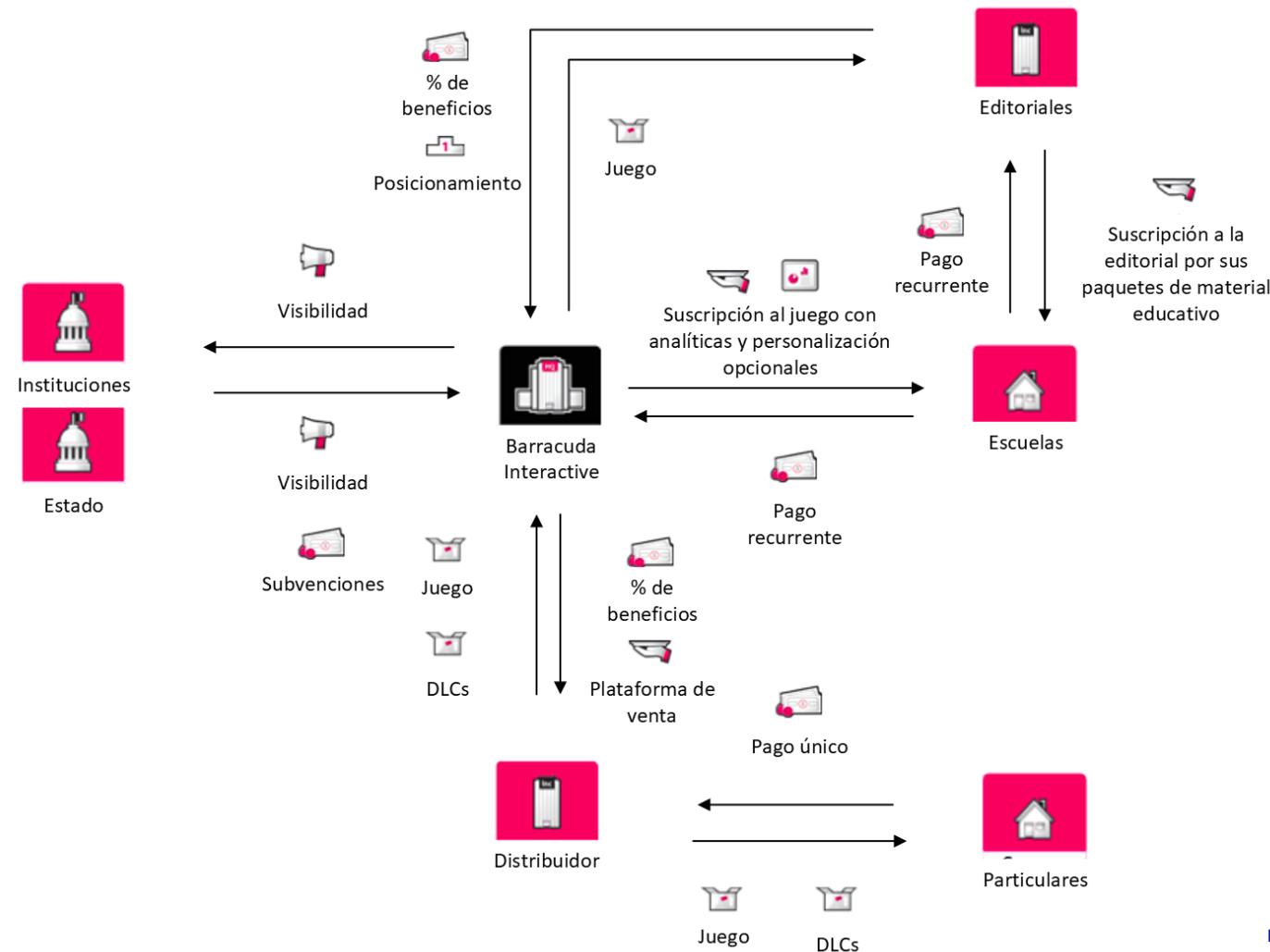


Ilustración 67. Caja de Herramientas

#### 10.1.4 MODELO DE LIENZO

Se presenta seguidamente el Canvas elaborado para Rally Team Tactics:



\*

**Acuerdos institucionales:** mantener conversaciones y llegar a acuerdos con instituciones nacionales e internacionales y el Estado para financiar nuestro videojuego y promover su uso en las aulas.

**Integración con el modelo educativo:** asegurarse de que el producto sea adecuado a los currículos escolares actuales.

**Promoción y marketing:** promoción del videojuego en ferias del sector educativo, en colaboración con las editoriales, en las plataformas de distribución y en redes sociales.

**Mantenimiento y desarrollo de nuevo contenido:** supervisión de la recepción al juego tras el lanzamiento para el despliegue de parches y el potencial desarrollo de contenido adicional.

Ilustración 68. Canvas del modelo de negocio

## 10.2 MARKETING

## 10.3 VERSIONES PRELIMINARES DEL PRODUCTO

## 10.4 PRODUCTO FINAL

## 11 CONCLUSIONES

## REFERENCIAS Y BIBLIOGRAFÍA

## ANEXO