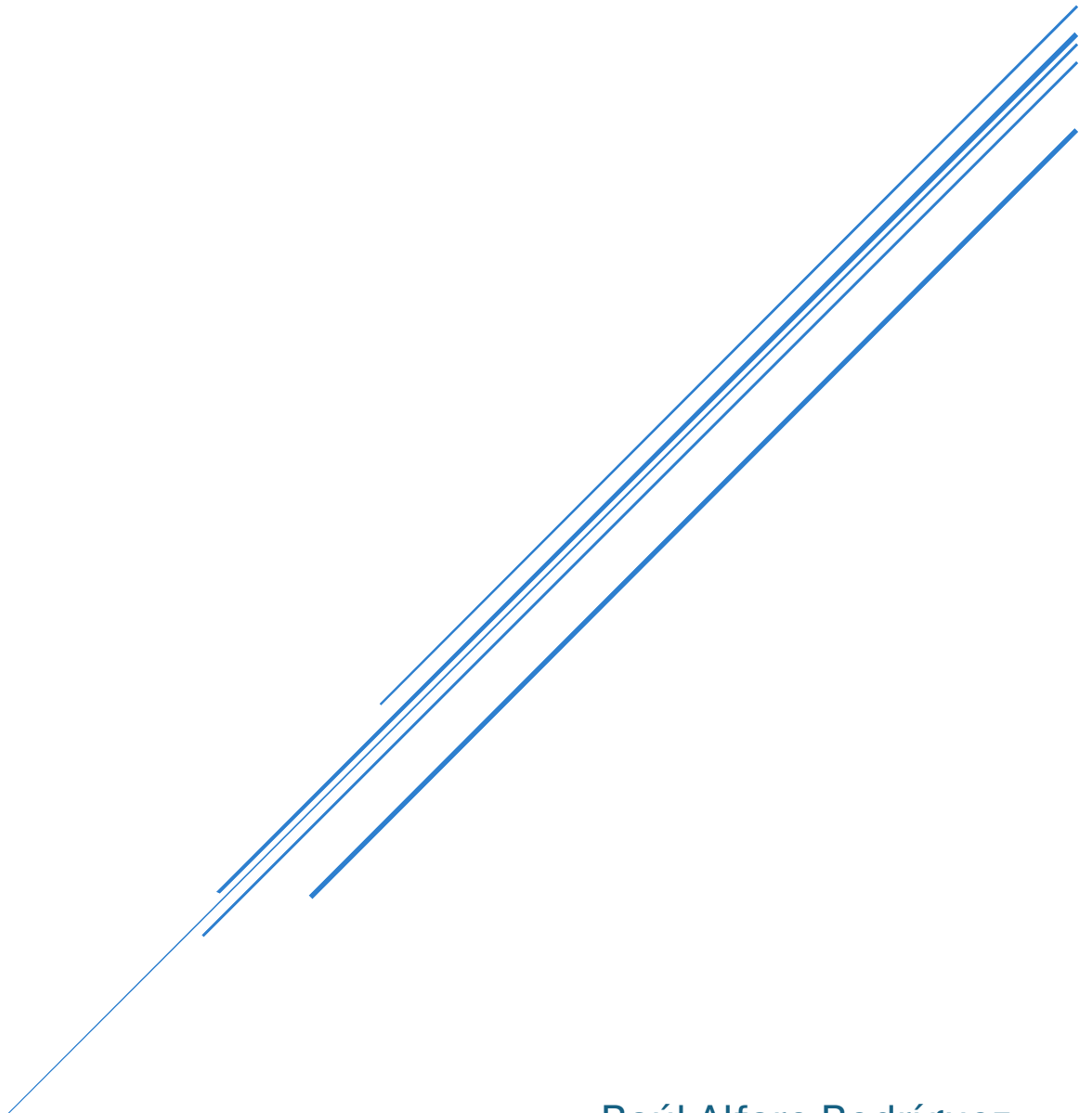


DOCUMENTO DE REQUERIMIENTOS

luiKi Kart



Raúl Alfaro Rodríguez
Eyden Su Díaz

DESCRIPCIÓN GENERAL

Objetivo

Se requiere desarrollar un juego de carreras multijugador en tiempo real donde:

1. Los jugadores se autentican con un *nickname* y pueden:
 - Crear partidas (eligiendo pista, vueltas y modo de juego).
 - Unirse a partidas existentes.
 - Competir en una pista predefinida (25x25) moviéndose con teclas.
2. **El servidor** sincroniza las posiciones de los jugadores en tiempo real, valida movimientos y determina al ganador (quien complete las vueltas primero).
3. **El sistema** debe mostrar un ranking de partidas anteriores y gestionar partidas temporales (se cierran si no inician en 3 minutos).

Desafío principal: Garantizar sincronización en tiempo real entre múltiples clientes y evitar inconsistencias.

Alcance

- **Frontend:** Interfaz web con menú (crear/unirse a partidas, ranking) y renderizado de pista (25x25).
- **Backend:** Servidor con API REST + WebSockets para gestión de partidas y sincronización en tiempo real.
- **Base de Datos:** Almacenamiento de rankings y configuración de pistas.
- **Funcionalidades:**
 - Movimiento con teclas (validado en servidor).
 - Temporizador de partidas (3 minutos de espera).
 - Detección de ganador (vueltas completadas).

LISTA DE RERQUERIMIENTOS

Requerimientos de Usuario

Funcionales

ID: RU-F001

Descripción: Los usuarios deben poder autenticarse con un nickname para acceder al sistema.

Importancia: Alta

Estimación Inicial: 1 día

Estrategia de Implementación: Campo de texto simple sin validación de usuarios persistente

ID: RU-F002

Descripción: Los usuarios deben poder crear una partida, seleccionando tipo de juego, pista, vueltas y cantidad de jugadores.

Importancia: Alta

Estimación Inicial: 3 días

Estrategia de Implementación: Formulario interactivo con opciones predefinidas y generación de ID único.

ID: RU-F003

Descripción: Los usuarios deben poder unirse a partidas existentes, visualizando lista actualizada en tiempo real.

Importancia: Alta

Estimación Inicial: 3 días

Estrategia de Implementación: Lista dinámica consumiendo API del servidor con WebSockets o polling.

ID: RU-F004

Descripción: Los usuarios deben poder seleccionar un vehículo antes de iniciar la partida.

Importancia: Media

Estimación Inicial: 2 días

Estrategia de Implementación: Menú desplegable con opciones de vehículos predefinidos.

ID: RU-F005

Descripción: Los usuarios deben poder controlar su vehículo con las teclas de dirección (flechas).

Importancia: Alta

Estimación Inicial: 2 días

Estrategia de Implementación: Event listeners en el frontend y validación de movimientos con el servidor.

ID: RU-F006

Descripción: Los usuarios deben poder ver el ranking de partidas anteriores.

Importancia: Media

Estimación Inicial: 2 días

Estrategia de Implementación: Tabla ordenada por tiempo/puestos, consumiendo datos del servidor.

No Funcionales

ID: RU-NF001

Descripción: La interfaz debe ser intuitiva y responsiva en dispositivos de al menos 1024x768 px.

Importancia: Media

Estimación Inicial: 2 días

Estrategia de Implementación: Diseño adaptable con CSS Grid/Flexbox.

ID: RU-NF002

Descripción: El sistema debe responder a acciones del usuario en eficientemente.

Importancia: Alta

Estimación Inicial: Depende de backend

Estrategia de Implementación: Optimización de llamadas API.

Requerimientos de Sistema

Funcionales

ID: RS-F001

Descripción: El sistema debe gestionar múltiples partidas simultáneas con sincronización en tiempo real.

Importancia: Alta

Estimación Inicial: 5 días

Estrategia de Implementación: Servidor con WebSockets para actualizaciones en vivo.

ID: RS-F002

Descripción: El sistema debe validar movimientos y posiciones de los vehículos en la pista.

Importancia: Alta

Estimación Inicial: 3 días

Estrategia de Implementación: Lógica de validación en servidor y respuesta a clientes.

ID: RS-F003

Descripción: El sistema debe finalizar partidas cuando se completen las vueltas o el tiempo límite.

Importancia: Alta

Estimación Inicial: 2 días

Estrategia de Implementación: Contador de vueltas y temporizador en el servidor.

ID: RS-F004

Descripción: El sistema debe generar y asignar un ID único a cada partida creada.

Importancia: Media

Estimación Inicial: 1 día

Estrategia de Implementación: UUID o hash único generado en el servidor.

ID: RS-F005

Descripción: El sistema debe eliminar partidas no iniciadas después de 3 minutos.

Importancia: Media

Estimación Inicial: 1 día

Estrategia de Implementación: Temporizador en el servidor con limpieza automática.

No Funcionales

ID: RS-NF001

Descripción: El sistema debe soportar al menos 50 usuarios concurrentes.

Importancia: Alta

Estimación Inicial: Depende de infraestructura

Estrategia de Implementación: Balance de carga y optimización de conexiones.

ID: RS-NF002

Descripción: Las actualizaciones de posición deben reflejarse en los clientes en menos de 500 ms.

Importancia: Alta

Estimación Inicial: 3 días

Estrategia de Implementación: WebSockets con mensajes optimizados.

Requerimientos de Software

Funcionales

ID: RSW-F001

Descripción: El software debe incluir al menos una pista predefinida de 25x25 con temática visual.

Importancia: Alta

Estimación Inicial: 2 días

Estrategia de Implementación: Matriz de celdas con propiedades definidas (inicio, fin, carriles).

ID: RSW-F002

Descripción: El software debe persistir datos de rankings y partidas finalizadas.

Importancia: Media

Estimación Inicial: 3 días

Estrategia de Implementación: Base de datos SQL/NoSQL o archivos JSON.

ID: RSW-F003

Descripción: El software debe permitir configuración de parámetros (tiempo de espera, límite de movimientos).

Importancia: Baja

Estimación Inicial: 1 día

Estrategia de Implementación: Archivo de configuración en el servidor.

No Funcionales

ID: RSW-NF001

Descripción: El código debe estar versionado en GitHub con acceso público o para el profesor.

Importancia: Alta

Estimación Inicial: 0.5 días

Estrategia de Implementación: Repositorio estructurado con README.md.

ID: RSW-NF002

Descripción: El software debe ser desarrollado en tecnologías web estándar (HTML5, CSS3, JavaScript).

Importancia: Media

Estimación Inicial: -

Estrategia de Implementación: Uso de frameworks modernos (React, Vue, etc.).

ID: RSW-NF003

Descripción: El backend debe ser escalable para futuras funcionalidades.

Importancia: Media

Estimación Inicial: -

Estrategia de Implementación: Arquitectura modular con APIs bien definidas.