

**I.E.S Fuengirola Nº1**

**DESARROLLO DE APLICACIONES WEB**

**PROYECTO INTERMODULAR**

# **WTCS Paddock**

**Departamento de Informática**

**Manual Técnico**

**Autor: D. Jorge Caro Almohalla**

## ÍNDICE DE CONTENIDOS

<b>1 - Sobre éste proyecto</b>	<b>3</b>
1.1 - Control de versiones	3
1.2 - Licencia de uso	3
<b>2 - Análisis del problema</b>	<b>3</b>
2.1 - Introducción al problema	3
2.2 - Antecedentes	3
2.3 - Objetivos	3
2.4 - Requisitos	3
2.4.1 - Funcionales	3
2.4.2 - No funcionales	3
2.5 - Recursos	3
2.5.1 - Software	3
2.5.2 - Hardware	3
<b>3 - Diseño de la solución software</b>	<b>4</b>
3.1 - Modelados	4
3.1.1 - Casos de uso	4
3.1.2 - Interacción	4
3.1.3 - Estado	4
3.1.4 - Actividad	4
3.2 - Prototipado gráfico	4
3.2.1 - Escritorio	4
3.2.2 - Tablets / Smartphones	4
3.3 - Base de datos	4
3.3.1 - Diseño Conceptual (ER)	4
3.3.2 - Diseño lógico (tablas normalizadas)	4
<b>4 - Implementación</b>	<b>4</b>
4.1 - Codificación	4
4.1.1 - Usabilidad	4
4.1.2 - Backend	4
4.1.3 - Frontend	4
4.2 - Pruebas	4
<b>5 - Documentación</b>	<b>5</b>
5.1 - Empaquetado / Distribución	5
5.2 - Instalación	5
5.3 - Manual de Usuario / Referencia	5
<b>6 - Conclusiones</b>	<b>6</b>
<b>7 - Bibliografía</b>	<b>6</b>

### 1 - Sobre éste proyecto

## **1.1 - Control de versiones**

## **1.2 - Licencia de uso**

# **2 - Análisis del problema**

## **2.1 - Introducción al problema**

## **2.2 - Antecedentes**

## **2.3 - Objetivos**

## **2.4 - Requisitos**

### **2.4.1 - Funcionales**

### **2.4.2 - No funcionales**

## **2.5 - Recursos**

### **2.5.1 - Software**

### **2.5.2 - Hardware**

## **3 - Diseño de la solución software**

### **3.1 - Modelados**

#### **3.1.1 - Casos de uso**

#### **3.1.2 - Interacción**

#### **3.1.3 - Estado**

#### **3.1.4 - Actividad**

### **3.2 - Prototipado gráfico**

#### **3.2.1 - Escritorio**

#### **3.2.2 - Tablets / Smartphones**

### **3.3 - Base de datos**

#### **3.3.1 - Diseño Conceptual (ER)**

#### **3.3.2 - Diseño lógico (tablas normalizadas)**

## **4 - Implementación**

### **4.1 - Codificación**

#### **4.1.1 - Usabilidad**

#### **4.1.2 - Backend**

#### **4.1.3 - Frontend**

### **4.2 - Pruebas**

## **5 - Documentación**

### **5.1 - Empaquetado / Distribución**

### **5.2 - Instalación**

### **5.3 - Manual de Usuario / Referencia**

## **6 - Conclusiones**

## **7 – Bibliografía**

## 1. Introducción

El mundo del *SimRacing* (automovilismo virtual) ha experimentado un crecimiento exponencial en la última década, pasando de ser un nicho de videojuegos a una disciplina competitiva seria. En este contexto, la gestión de ligas y campeonatos se ha vuelto una tarea compleja que requiere precisión y profesionalidad.

Este proyecto, titulado WTCS Paddock, nace de una necesidad real detectada en la liga *World Touring Car Series* (WTCS) en la que yo mismo participo. El objetivo es desarrollar una aplicación web multiplataforma que centralice la gestión deportiva, automatice el cálculo de resultados y ofrezca a los pilotos una interfaz moderna y accesible desde dispositivos móviles para seguir el campeonato.

### 1.1 Control de versiones

Para el desarrollo de este proyecto se utilizará Git como sistema de control de versiones y GitHub para el alojamiento del repositorio remoto. Se seguirá una estrategia de ramas (*branching*) simple, manteniendo una rama main para código estable y ramas secundarias para el desarrollo de nuevas características (feature/login, feature/standings).

### 1.2 Licencia de uso

Este proyecto se distribuye bajo la licencia Creative Commons Atribución-NoComercial 4.0 Internacional (CC BY-NC 4.0).

Esta elección refleja la filosofía de código abierto (*Open Source*) y el carácter académico del trabajo. Bajo esta licencia se establecen las siguientes condiciones:

- Compartir: Cualquier persona o entidad es libre de copiar y redistribuir el material en cualquier medio o formato.
- Adaptar: Se permite remezclar, transformar y construir a partir del material existente.
- Atribución: Se debe dar crédito al autor original (D. Jorge Caro Almohalla) de manera adecuada.
- No Comercial: No se permite hacer uso de este material ni de sus derivados con fines comerciales. Esto garantiza que la aplicación no podrá ser monetizada ni vendida por terceros, manteniéndose como una herramienta gratuita para la comunidad de *SimRacing* y la liga WTCS.

### 1.3 Metodología de desarrollo

Se seguirá una metodología ágil simplificada (Kanban) utilizando la herramienta Trello para la gestión de tareas. El desarrollo se dividirá en *sprints* centrados en funcionalidades clave: Base de Datos, Autenticación, Panel de Administración y Frontend Público.

---

## 2. Análisis del problema

### 2.1 Introducción al problema

La WTCS es una liga consolidada con una gran base de pilotos. Actualmente, toda la gestión de la competición se realiza de forma manual y fragmentada. La comunicación oficial ocurre en un servidor de Discord, mientras que los datos deportivos (clasificaciones, sanciones, inscripciones) se manejan mediante hojas de cálculo de Excel complejas y estáticas.

### 2.2 Antecedentes

El sistema actual presenta graves deficiencias de escalabilidad y usabilidad:

- Saturación de datos: Como indica el organizador, las hojas de cálculo han llegado a su límite físico ("*literally cannot fit every driver onto here anymore*"), lo que hace que no todos los pilotos se vean representados en las clasificaciones.
- Accesibilidad nula en móviles: Los pilotos deben hacer zoom en imágenes de tablas de Excel publicadas en Discord para ver sus puntos, lo cual ofrece una experiencia de usuario (UX) muy pobre en *smartphones*.
- Gestión manual propensa a errores: El cálculo de puntos, la aplicación de sanciones y la ordenación de la parrilla se hacen a mano, lo que consume horas de trabajo del organizador y aumenta el riesgo de error humano.

## 2.3 Objetivos

El objetivo principal es desarrollar una Aplicación Web Progresiva (PWA) que centralice la gestión deportiva y administrativa de la liga, sustituyendo exclusivamente el uso de hojas de cálculo (Excel).

La plataforma web actuará como un complemento estratégico al servidor de Discord actual. Mientras que la web se encargará de procesar datos, estadísticas y clasificaciones de forma dinámica, Discord se mantendrá como el núcleo social, canal de voz y chat instantáneo, liberándolo de la carga de actuar como repositorio de información estática.

Objetivos específicos:



1. Automatizar la clasificación: Eliminar el cálculo manual. Que el sistema procese automáticamente los puntos de Pilotos, Equipos y Fabricantes basándose en los resultados introducidos, reduciendo la carga de trabajo de la administración.
2. Centralización de Información: Migrar la documentación oficial (calendario, normativas, sanciones) de canales de texto dispersos en Discord a una web organizada y permanente.
3. Interfaz Mobile-First: Diseñar un *Dashboard* donde la información crítica (próxima carrera, top 3) sea visible de un vistazo en pantallas móviles, mejorando la experiencia de usuario respecto al zoom en imágenes estáticas.
4. Gestión de Roles: Diferenciar entre Administradores (organización) y Usuarios (pilotos), permitiendo una gestión segura de los datos.
5. Escalabilidad: Permitir un número ilimitado de pilotos y temporadas históricas, solucionando la limitación actual de espacio en las hojas de cálculo.

## 2.4 Requisitos

### 2.4.1 Requisitos Funcionales (Lo que hace la app):

- Sistema de Noticias: El administrador debe poder publicar anuncios (horarios, normativas) que aparezcan en la página principal.
- Gestión de Carreras: Visualización de un calendario con cuenta atrás para el próximo evento.
- Standings Interactivos: Tablas de clasificación filtrables por Pilotos, Equipos (Works/Privateer) y Marcas.

- Autenticación: Sistema de Login seguro para los pilotos.

#### 2.4.2 Requisitos No Funcionales (Cómo lo hace):

- Responsive Design: La web debe adaptarse fluidamente a resoluciones de escritorio, tablet y móvil.
- Rendimiento: Carga rápida de imágenes y datos.
- Usabilidad: Menús de navegación intuitivos y claros.

### 2.5 Recursos

#### 2.5.1 Software (Stack Tecnológico):

- Backend: PHP 8.2 con Framework Laravel.
- Base de Datos: MySQL.
- Panel de Administración: FilamentPHP (para gestión rápida de recursos).
- Frontend: Blade Templates + Tailwind CSS (para estilos) + JavaScript (para interactividad).
- Diseño: Figma (Wireframing).

#### 2.5.2 Hardware:

- Ordenador de desarrollo:  
SO: Windows 11  
CPU: 8 núcleos 16 hilos  
RAM: 32GB
- Dispositivo móvil Android para pruebas de entorno real.

### 3. Diseño de la solución software

#### 3.1 Modelados

(Aquí insertarás tus diagramas más adelante, deja el hueco)

- **3.1.1 Diagrama de Casos de Uso**
- **3.1.2 Diagrama de Flujo (Login o Inscripción)**

#### 3.2 Prototipado gráfico

Para la interfaz de usuario se ha optado por un diseño oscuro (*Dark Mode*) acorde a la estética del *Motorsport* y *SimRacing*.

Estructura Principal (Dashboard):

Se ha diseñado una página de aterrizaje (*Landing Page*) que prioriza la información inmediata:

1. Barra de Navegación (Navbar): Adaptable. En escritorio muestra enlaces directos (*Drivers, Teams, Standings*) y en móvil colapsa en un menú hamburguesa para ahorrar espacio.
2. Panel Superior (Hero Section): Dividido en tres bloques clave:
  - Standings: Top 3 de pilotos y equipos para referencia rápida.
  - Countdown: Cuenta atrás en tiempo real para la próxima carrera.
  - Track Info: Trazado del circuito de la próxima ronda.
3. Sección de Noticias: Un *grid* de tarjetas con las últimas novedades de la liga, sustituyendo los anuncios de texto plano de Discord.

**(Aquí pegarás la imagen de tu Wireframe de Figma que me enseñaste).**

### 3.3 Base de datos

El sistema requiere una base de datos relacional para gestionar la integridad de la competición.

#### 3.3.1 Diseño Conceptual (Entidades Clave):

- Users (Usuarios): Almacena los datos de los pilotos y administradores. Incluye roles y avatares.
- Teams (Equipos): Entidad fundamental para el campeonato de constructores. Se clasifican por tipo: *Works* (Oficiales) o *Privateer* (Privados).
- Seasons (Temporadas): Permite archivar datos de años anteriores y gestionar el campeonato actual.
- Races (Carreras): Eventos del calendario. Contienen fecha, circuito y estado (programada/finalizada).
- Tracks (Circuitos): Información estática sobre los trazados (nombre, país, imagen del mapa).
- Results (Resultados): La tabla pivote principal. Relaciona una Carrera con un Piloto, almacenando su posición final, vuelta rápida y penalizaciones. El sistema usará estos datos para calcular los puntos totales.