



Grado en Análisis de Negocios/Business Analytics

Trabajo Fin de Grado

**RACESCOPE: OPTIMIZACIÓN DE LA  
ESTRATEGIA DE CARRERA EN FÓRMULA 1  
MEDIANTE INTELIGENCIA ARTIFICIAL**

Autor

Álvaro González Tabernero

Dirigido por

Pablo Martín Muñoz

Madrid

Junio 2026



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

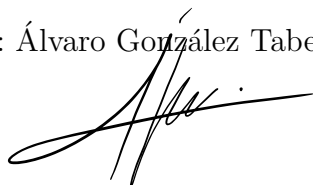
**RACESCOPE: OPTIMIZACIÓN DE LA ESTRATEGIA  
DE CARRERA EN FÓRMULA 1 MEDIANTE  
INTELIGENCIA ARTIFICIAL**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2025/26 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Álvaro González Tabernero

Fecha: 14 de junio de 2026



Autorizada la entrega del proyecto

**EL DIRECTOR DEL PROYECTO**

Fdo.: Pablo Martín Muñoz

Fecha: 14 de junio de 2026



## Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

**Advertencia:** Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, **NO** está permitido su uso en la elaboración del Trabajo Fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Álvaro González Tabernero, estudiante del Doble Grado en Ingeniería en Tecnologías de Telecomunicación y Análisis de Negocios/Business Analytics (GIT-BA) de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado **“RACESCOPE: OPTIMIZACIÓN DE LA ESTRATEGIA DE CARRERA EN FÓRMULA 1 MEDIANTE INTELIGENCIA ARTIFICIAL”**, declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Crítico:** Para encontrar contra-argumentos a una tesis específica que pretendo defender.
2. **Referencias:** Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.
3. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
4. **Interpretador de código:** Para realizar análisis de datos preliminares.
5. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.
6. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.
7. **Generador previo de diagramas de flujo y contenido:** Para esbozar diagramas iniciales.
8. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
9. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
10. **Traductor:** Para traducir textos de un lenguaje a otro.
11. **Gestión del control de versiones:** Para asistir en el uso de Git y GitHub (mensajes de commit, estructura del repositorio y operaciones rutinarias del control de versiones).

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para qué se ha usado ChatGPT u otras herramientas similares). Soy

consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

**Fecha:** 13 de junio de 2026

**Firma:**

A handwritten signature in black ink, consisting of several overlapping, fluid strokes that form a stylized, cursive-like mark.

# RACESCOPE: OPTIMIZACIÓN DE LA ESTRATEGIA DE CARRERA EN FÓRMULA 1 MEDIANTE INTELIGENCIA ARTIFICIAL

**Autor:** Álvaro González Tabernero

**Director:** Pablo Martín Muñoz

**Entidad colaboradora:** Universidad Pontificia Comillas, ICAI

## RESUMEN DEL PROYECTO

*RaceScope* es un sistema de exploración de estrategias pre-carrera para Fórmula 1 que combina un Transformer de predicción de ritmo con simulación Monte Carlo. El motor articula tres componentes: un perfil paramétrico lineal por piloto-circuito-compuesto que resuelve analíticamente la vuelta de pit; un Transformer v3 por piloto, con cabeza global de respaldo, que predice el ritmo de vuelta de forma no lineal; y un optimizador media-varianza que ordena las candidatas. Los datos provienen de OpenF1 (2023-2024 para entrenamiento, 2025 *held-out*). El sistema se entrega como una API FastAPI y una aplicación web React.

## Palabras clave:

inteligencia artificial, Transformers, simulación Monte Carlo, estrategia de carrera, Fórmula 1, predicción de tiempos de vuelta, degradación de neumáticos, aprendizaje profundo, optimización bajo incertidumbre, FastAPI.

## MOTIVACIÓN

La estrategia de paradas en boxes es uno de los problemas de decisión más estudiados de la Fórmula 1. Los equipos profesionales operan simuladores propietarios inaccesibles fuera del muro de boxes y la literatura académica produce simuladores reproducibles solo recreando el simulador entero. *RaceScope* cubre el hueco intermedio: un sistema *open-source* sobre datos públicos que ordena las estrategias en segundos sobre una interfaz web, dirigido al aficionado y al analista externo.

## OBJETIVOS

Diseñar, implementar y evaluar un sistema *end-to-end* de exploración de estrategias de paradas. Los cinco objetivos específicos son: (1) construir un pipeline sobre OpenF1; (2) entrenar Transformers por piloto con respaldo global; (3) ajustar un perfil paramétrico sensible a las condiciones térmicas; (4) integrar ambos modelos en un motor que clasifique bajo criterio media-varianza; y (5) entregar el sistema como una aplicación web sobre una API REST.

## METODOLOGÍA

La metodología se estructura en cuatro fases:

1. Ingestión y preprocesamiento de datos de telemetría F1 desde la API OpenF1.
2. Entrenamiento de modelos Transformer de predicción de tiempos de vuelta específicos por piloto, con *fallback* a un modelo global, y ajuste del perfil paramétrico de degradación por piloto-circuito-compuesto.
3. Diseño e implementación de un motor de estrategia organizado en tres componentes: resolución analítica de la vuelta óptima (Motor 1), refinamiento Monte Carlo del Top- $K$  mediante *rollout* autoregresivo del Transformer (Motor 2), y optimizador de cartera media-varianza (Motor 3).
4. Desarrollo de una API REST (FastAPI) y una aplicación web (React + Vite) para la exploración interactiva de estrategias pre-carrera.

## RESULTADOS

Sobre la temporada 2025 (*held-out*), el motor reporta un MAPE entre el 15,5 % y el 16,9 % en los cuatro pilotos evaluados (LEC, NOR, PIA, VER) y un sesgo positivo sistemático entre +636 s y +692 s, atribuible al margen frente a *safety car*. La simulación del campeonato Verstappen-Norris arroja una concordancia carrera a carrera del 50 % y declara a Verstappen vencedor por 70 puntos, frente al empate 11-11 real.

## CONCLUSIONES

El sistema cumple su objetivo general: ordena las estrategias pre-carrera en segundos sobre una interfaz reactiva, con una calibración del orden del 16 % en el peor caso. Las dos limitaciones estructurales son la deriva temporal entre las temporadas de entrenamiento y 2025, y la ausencia de información de carrera en tiempo real.

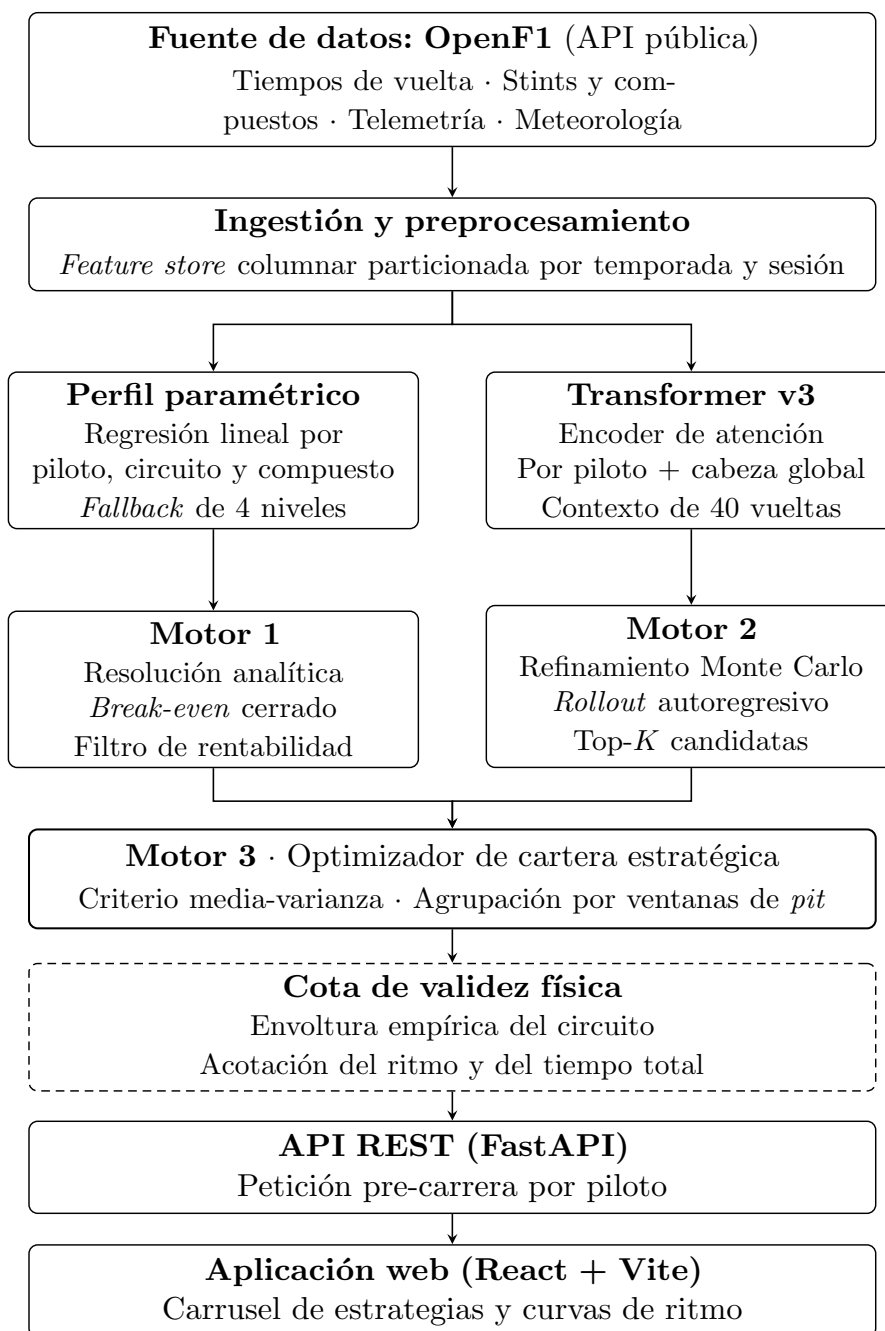


Figura 1: Arquitectura de *RaceScope*: flujo de datos desde OpenF1 hasta la API y la aplicación web. Elaboración propia.



# RACESCOPE: PRE-RACE STRATEGY OPTIMISATION IN FORMULA 1 USING ARTIFICIAL INTELLIGENCE

**Author:** Álvaro González Taberero

**Supervisor:** Pablo Martín Muñoz

**Collaborating Institution:** Universidad Pontificia Comillas, ICAI

## PROJECT ABSTRACT

*RaceScope* is a pre-race strategy exploration system for Formula 1 that combines a lap-pace Transformer with Monte Carlo simulation. The engine articulates three components: a linear parametric profile per driver-circuit-compound that solves the pit lap analytically; a per-driver Transformer v3 with a global fallback head that predicts lap pace non-linearly; and a mean-variance portfolio optimiser that ranks the candidates. Data is sourced from OpenF1 (2023-2024 for training, 2025 held-out). The system is delivered as a FastAPI service and a React web application.

## Keywords:

Artificial Intelligence, Transformers, Monte Carlo simulation, race strategy, Formula 1, lap time prediction, tyre degradation, deep learning, optimisation under uncertainty, FastAPI.

## MOTIVATION

Pit-stop strategy is one of the most studied decision problems in Formula 1. Professional teams operate proprietary simulators inaccessible outside the pit wall, and academic simulators are reproducible only by recreating the simulator itself. *RaceScope* covers the gap in between: an open-source system on public data that ranks strategies in seconds on a web interface, aimed at the fan and the external analyst.

## OBJECTIVES

Design, implement and evaluate an end-to-end pit-stop strategy exploration system. The five specific objectives are: (1) building an ingestion and preprocessing pipeline on top of OpenF1; (2) training per-driver Transformers with a global fallback; (3) fitting a parametric profile sensitive to thermal conditions; (4) integrating both models into an engine that ranks candidates under a mean-variance criterion; and (5) delivering the system as a web application on top of a REST API.

## METHODOLOGY

The methodology is structured into four phases:

1. Ingestion and preprocessing of F1 telemetry data from the OpenF1 API.
2. Training of per-driver Transformer models for lap time prediction with a fallback to a global model, and fitting of the parametric degradation profile per driver-circuit-compound.
3. Design and implementation of a strategy engine organised in three components: analytical resolution of the optimal pit lap (Motor 1), Monte Carlo refinement of the Top- $K$  via autoregressive Transformer *rollout* (Motor 2), and a mean-variance portfolio optimiser (Motor 3).
4. Development of a REST API (FastAPI) and a web application (React + Vite) for interactive pre-race strategy exploration.

## RESULTS

On the 2025 season held out entirely from training, the engine reports a MAPE between 15.5 % and 16.9 % across the four evaluated drivers (LEC, NOR, PIA, VER) and a systematic positive bias between +636s and +692s, attributable to the *safety-car* margin. The Verstappen-Norris championship simulation yields a 50 % race-by-race agreement and declares Verstappen the hypothetical champion by 70 points, against the real 11-11 tie.

## CONCLUSIONS

The system meets its general objective: it ranks pre-race strategies in seconds on a reactive interface, with a worst-case calibration of about 16 %. The two structural limitations are the temporal drift between the training seasons and 2025, and the absence of real-time race information.

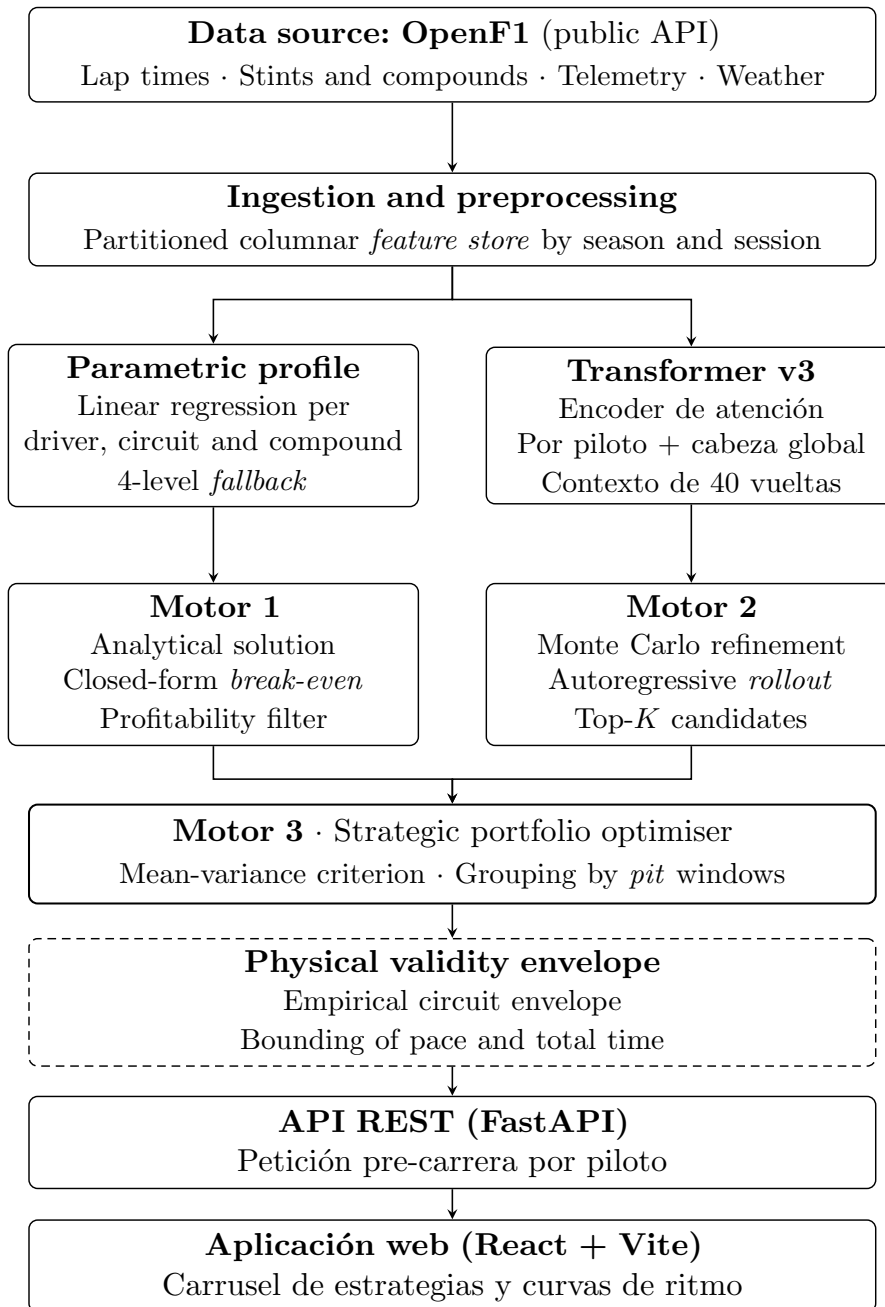


Figura 2: *RaceScope* system architecture: data flow from OpenF1 to the API and web application. Own elaboration.



## Agradecimientos

Quiero empezar agradeciendo a Pablo, mi tutor, por la confianza al aceptar dirigir un proyecto con tanta motivación personal detrás y por darme el espacio para llevarlo hasta el final esforzándome al máximo en cada paso.

A mis padres, a mi hermana Sofía y al resto de mi familia, por el apoyo constante durante los cinco años del doble grado y, en particular, durante los meses que ha durado este trabajo.

Mención especial para mi abuelo Pedro y mi tío Javier: por ser referencia, ejemplo, inspiración y compañía, tanto en lo personal como en lo profesional, a lo largo de todos estos años.

Y por último, una dedicatoria especial a mi primo y ahijado, Felipe. Ojalá dentro de unos años puedas leer este trabajo y engancharte a la competición del motor como ya lo está tu primo.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Objetivos . . . . .	2
1.3. Estructura del documento . . . . .	3
<b>2. Contexto y Estado del Arte</b>	<b>5</b>
2.1. La estrategia de carrera en Fórmula 1 . . . . .	5
2.1.1. El problema de la parada en boxes . . . . .	5
2.1.2. El ritmo de vuelta como variable clave . . . . .	6
2.2. Fuente de Datos: OpenF1 . . . . .	7
2.2.1. Dominios de datos relevantes para RaceScope . . . . .	7
2.3. Transformers para la predicción de tiempos de vuelta . . . . .	9
2.4. Modelos lineales jerárquicos para regímenes de datos limitados . . . . .	11
2.5. Simulación Monte Carlo y optimización bajo incertidumbre . . . . .	12
2.5.1. Fundamentos del método Monte Carlo . . . . .	12
2.5.2. Clasificación de estrategias bajo incertidumbre . . . . .	12
2.6. Trabajos relacionados . . . . .	13
2.6.1. Software y herramientas existentes . . . . .	13
<b>3. Metodología</b>	<b>15</b>
3.1. Visión general del sistema . . . . .	15
3.2. Pipelines offline y online . . . . .	16
<b>4. Caso de Estudio: RaceScope</b>	<b>19</b>
4.1. Pipeline de datos . . . . .	19
4.1.1. Ingestión de datos . . . . .	19
4.1.2. Preprocesamiento e ingeniería de características . . . . .	20
4.2. Modelo de predicción de tiempos de vuelta . . . . .	21
4.2.1. Arquitectura Transformer . . . . .	21
4.2.2. Estrategia de entrenamiento y modelos de fallback . . . . .	23

4.2.3.	Evolución del modelo: del LSTM al Transformer v3 . . . . .	24
4.3.	Modelo paramétrico de perfil de piloto . . . . .	26
4.3.1.	Formulación del modelo . . . . .	26
4.3.2.	Jerarquía de fallback de cuatro niveles . . . . .	27
4.4.	Motor de estrategia: tres motores que colaboran . . . . .	27
4.4.1.	Contexto de carrera . . . . .	28
4.4.2.	Motor 1: Resolución analítica de la vuelta óptima . . . . .	28
4.4.3.	Motor 2: Refinamiento Monte Carlo con el Transformer . . . . .	29
4.4.4.	Motor 3: Optimizador de cartera estratégica . . . . .	30
4.4.5.	Cota de validez física: envoltura del circuito y acotaciones de seguridad . . . . .	30
4.5.	Métricas de evaluación . . . . .	33
4.5.1.	Evaluación de los modelos de predicción de ritmo . . . . .	34
4.5.2.	Evaluación del motor de estrategia . . . . .	34
4.5.3.	Evaluación del sistema: latencia . . . . .	34
<b>5.</b>	<b>Resultados y Evaluación</b>	<b>37</b>
5.1.	Configuración experimental . . . . .	37
5.2.	Evaluación sobre la temporada 2025 . . . . .	38
5.2.1.	Métricas y procedimiento . . . . .	38
5.2.2.	Resultados agregados . . . . .	39
5.3.	Simulación del campeonato 2025: Verstappen vs Norris . . . . .	41
5.3.1.	Metodología . . . . .	41
5.3.2.	Resultados predichos y oficiales . . . . .	42
5.4.	Aplicación web . . . . .	43
5.4.1.	Estructura de navegación . . . . .	43
5.4.2.	Diseño visual . . . . .	45
5.4.3.	Recorrido del usuario . . . . .	45
<b>6.</b>	<b>Discusión, Conclusiones y Trabajo Futuro</b>	<b>51</b>
6.1.	Discusión de los resultados . . . . .	51
6.2.	Conclusiones . . . . .	53
6.3.	Limitaciones . . . . .	55
6.4.	Trabajo futuro . . . . .	56
6.4.1.	Mejoras a corto plazo . . . . .	56
6.4.2.	Extensiones del modelo . . . . .	57
6.4.3.	Evolución del sistema . . . . .	57
	<b>Bibliografía</b>	<b>59</b>
	<b>A. Alineación con los Objetivos de Desarrollo Sostenible</b>	<b>63</b>

<b>B. Evaluación de calibración: gráficas por piloto</b>	<b>65</b>
<b>C. Código fuente del proyecto</b>	<b>73</b>



# Índice de figuras

1.	Arquitectura de <i>RaceScope</i> : flujo de datos desde OpenF1 hasta la API y la aplicación web. Elaboración propia. . . . .	IX
2.	<i>RaceScope</i> system architecture: data flow from OpenF1 to the API and web application. Own elaboration. . . . .	XIII
3.1.	Arquitectura de <i>RaceScope</i> : flujo de datos desde OpenF1 hasta la API y la aplicación web. Incluye ingesta, dos modelos de ritmo (perfil paramétrico y Transformer), tres motores de estrategia y una cota de validez física. Elaboración propia. . . . .	17
5.1.	MAPE y sesgo medio del motor de predicción por piloto sobre las carreras evaluadas de 2025. Izquierda: precisión global. Derecha: sesgo calculado como media(predicho – real) en segundos. . . . .	40
5.2.	Error predicho – real carrera a carrera para Norris, Piastri, Verstappen y Leclerc. Comparten el eje vertical. La franja gris marca el margen $\pm 60$ s aplicado por el motor frente a un evento de <i>safety car</i> . . . . .	40
5.3.	Evolución acumulada de los puntos de Verstappen y Norris bajo el sistema de puntos 25/18 por carrera. Eje horizontal: índice de carrera en orden cronológico del calendario 2025. . . . .	44
5.4.	<i>Landing page</i> de <i>RaceScope</i> , con titular, llamada a la acción y vista previa animada de curvas de ritmo. . . . .	46
5.5.	Vista comparativa Pre-race con las dos filas de piloto lado a lado tras pulsar Calcular (caso Verstappen vs Norris). . . . .	48
5.6.	Vista comparativa Pre-race con las dos filas de piloto lado a lado tras pulsar Calcular (caso Antonelli vs Leclerc). . . . .	48
5.7.	Gráfico de curva de ritmo: líneas por <i>stint</i> coloreadas por compuesto, bandas de ventana de <i>pit</i> y lectura dinámica por vuelta bajo el cursor. . . . .	49
B.1.	Series temporales del error firmado predicho – real por piloto sobre las carreras evaluadas de 2025. En cada panel la franja gris marca el margen $\pm 60$ s aplicado por el motor frente a un evento de <i>safety car</i> . . . . .	67

B.2. Norris: tiempo predicho (círculo blanco) frente al tiempo real (círculo relleno) carrera a carrera, con el error firmado anotado a la derecha.	68
B.3. Piastri: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha. . . . .	69
B.4. Verstappen: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha. . . . .	70
B.5. Leclerc: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha. . . . .	71

# Índice de tablas

2.1. Dominios de información disponibles en la API de OpenF1 y su uso en <i>RaceScope</i> . . . . .	8
4.1. Características continuas de entrada del modelo Transformer de predicción de tiempos de vuelta . . . . .	22
4.2. Resumen comparativo de las tres variantes del Transformer de predicción de ritmo. . . . .	26
4.3. Dispersión del tiempo total esperado entre pilotos al introducir progresivamente las acotaciones. La columna $\Delta T$ es la diferencia entre el máximo y el mínimo entre los 20 pilotos de Sakhir 2023. . .	33
5.1. Métricas de calibración por piloto, MAPE y sesgo evaluados a nivel de tiempo total de carrera; MAE y RMSE evaluados sobre el tiempo de vuelta predicho por el modelo de ritmo; Top-5 indica la proporción de carreras en las que la estrategia real aparece entre las cinco estrategias mejor clasificadas por el motor. . . . .	39
5.2. Cuenta final del campeonato 2025 simulado ( <i>RaceScope</i> ) frente al resultado real, restringido al duelo Verstappen-Norris. Cada carrera asigna 25 puntos al ganador y 18 al segundo. . . . .	42
5.3. Pestañas de la aplicación web y su propósito. . . . .	45



# Capítulo 1

## Introducción

La Fórmula 1 es el máximo campeonato mundial de automovilismo, donde equipos y pilotos compiten en circuitos de todo el mundo para alcanzar victorias y campeonatos. Cada temporada consta de una serie de carreras (Grandes Premios, GPs) en las que ahora participan 11 equipos y 22 pilotos, todos pilotando los monoplazas más rápidos y más avanzados tecnológicamente del mundo. La F1 es una competición de alta velocidad, precisión y estrategia que combina el rendimiento puro del monoplaza con una dimensión táctica de enorme complejidad.

Aunque el talento del piloto y la calidad del coche determinan el techo de rendimiento disponible, la estrategia de carrera es el factor que a menudo decide quién lo aprovecha. Las decisiones tácticas sobre cuándo parar en boxes (pit stop), qué compuesto de neumáticos montar y cómo gestionar el ritmo para controlar la degradación, a menudo significan la diferencia entre ganar o perder una carrera. Esta complejidad se acentúa por factores impredecibles como los coches de seguridad (safety car, SC), las condiciones meteorológicas cambiantes o el tráfico en pista, que alteran continuamente el escenario de competición.

La estrategia de carrera se puede reducir a la optimización del desgaste de los neumáticos durante el evento, y la gestión de los eventos impredecibles que puedan alterar repentinamente el estado de la carrera. Para ello, se requiere una gran cantidad de datos de telemetría del monoplaza, estadísticas sobre cada compuesto de neumático, información sobre el circuito, el asfalto, condiciones meteorológicas, el tráfico en pista, la probabilidad de safety car, etc.

Hasta hace poco, los algoritmos y las aplicaciones para la determinación de estrategias de carrera eran muy complejas de hacer desde la perspectiva de un aficionado o un ingeniero independiente, ya que requerían acceso a una cantidad y un tipo de datos que solo tienen los equipos de F1, y que por su naturaleza son altamente

confidenciales. Sin embargo, la democratización del acceso a estos datos gracias a plataformas como la API OpenF1 [1], sobre la que se construye este proyecto, y el avance en las técnicas de aprendizaje profundo permiten que cualquier persona desarrolle análisis estratégicos que antes requerían la infraestructura y los datos propietarios de los propios equipos.

### 1.1. Motivación

Desde que empecé a ver carreras de F1, siempre me ha fascinado el aspecto estratégico de la competición y cómo la diferencia de una vuelta en la parada en boxes cambia completamente el resultado de una carrera. Además, es la parte del fin de semana donde el trabajo de los analistas y los ingenieros de estrategia sale a relucir, y no se centra toda la actividad en el piloto.

Sin embargo, a pesar de mi interés y fascinación por la estrategia, siempre he sentido que, por la naturaleza compleja del tema, hay mucha gente que desprecia la carreras más estratégicas y se centran solo en la acción en pista y la adrenalina de los adelantamientos. GPs como Mónaco o Barcelona a menudo son criticados por ser “aburridos” o “predecibles”, cuando en realidad esconden una batalla estratégica de fondo que es fascinante de analizar y entender.

Además, el auge de la Inteligencia Artificial (IA) y de las técnicas de aprendizaje automático y profundo en los últimos años hacen de este el momento perfecto para impulsar una herramienta de este tipo. La capacidad de modelar la degradación de los neumáticos con Transformers entrenados de forma específica por piloto, y de simular estrategias bajo incertidumbre con métodos Monte Carlo, hace viable un análisis que hasta hace pocos años requería acceso a datos de telemetría restringidos a los propios equipos.

Por tanto, este proyecto nace del deseo de intentar mostrar esta parte de la competición a gente que no la conoce o no la comprende, y de usar los conocimientos adquiridos durante el grado para crear una herramienta que permita a cualquier persona explorar y entender la estrategia de carrera en F1 de una manera accesible, visual e interactiva.

### 1.2. Objetivos

El objetivo general de este Trabajo Fin de Grado es el diseño, implementación y evaluación de un sistema end-to-end (E2E) de exploración de estrategias de carrera

en Fórmula 1, denominado **RaceScope**, que combina modelos de aprendizaje profundo para la predicción de tiempos de vuelta con un motor de simulación basado en Monte Carlo para la clasificación robusta de estrategias de pit stop.

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

1. Construir un pipeline de ingestión y preprocesamiento de datos de telemetría F1 a partir de la API OpenF1, generando una *feature store* estructurada por temporada, circuito y piloto.
2. Diseñar y entrenar modelos de predicción de tiempos de vuelta específicos por piloto, con un modelo global de respaldo para pilotos con datos insuficientes.
3. Desarrollar un modelo paramétrico de degradación de neumáticos por piloto y compuesto, sensible a las condiciones de temperatura de pista y ambiente.
4. Implementar un motor de estrategia construido sobre tres componentes que colaboran: un perfil lineal por piloto que resuelve la vuelta óptima de parada por break-even cerrado, un Transformer que predice ritmo no lineal y refina las mejores candidatas por simulación Monte Carlo, y un optimizador de cartera estratégica que ordena las propuestas por una combinación media-varianza inspirada en Markowitz.
5. Desplegar el sistema completo como una aplicación web interactiva, con visualizaciones de estrategias, tiempos de vuelta y análisis de sensibilidad.

## 1.3. Estructura del documento

El presente documento se organiza de la siguiente manera:

El **Capítulo 2** fija el contexto del problema y revisa los fundamentos teóricos y los trabajos relacionados sobre los que se apoya el sistema.

El **Capítulo 3** presenta la visión metodológica del sistema y la arquitectura general de sus componentes.

El **Capítulo 4** desarrolla el caso de estudio con las decisiones de diseño y los parámetros concretos del sistema implementado.

El **Capítulo 5** expone los resultados experimentales sobre la temporada 2025 y describe la aplicación web como artefacto de usuario.

El **Capítulo 6** discute los resultados, recoge las conclusiones, las limitaciones del trabajo y las líneas de continuación.

## *CAPÍTULO 1. INTRODUCCIÓN*

---

Los **Anexos** complementan el cuerpo del documento con la alineación del proyecto con los Objetivos de Desarrollo Sostenible (Anexo A), el material gráfico de la evaluación 2025 (Anexo B) y el código fuente completo del proyecto (Anexo C).

# Capítulo 2

## Contexto y Estado del Arte

A lo largo de este capítulo se exponen los pilares teóricos sobre los que se construye *RaceScope*: la estrategia en Fórmula 1 como problema de decisión, la fuente de datos pública sobre la que se sustenta el sistema, la arquitectura Transformer aplicada a la predicción de tiempos de vuelta, los modelos lineales jerárquicos como técnica de regularización para regímenes de datos limitados y la simulación Monte Carlo combinada con un criterio media-varianza para la clasificación de estrategias bajo incertidumbre. La descripción concreta de cómo *RaceScope* instancia cada uno de estos pilares se reserva para el Capítulo 4.

### 2.1. La estrategia de carrera en Fórmula 1

La Fórmula 1 es una competición en la que el resultado final no depende únicamente de la velocidad absoluta del monoplaza ni del talento del piloto: la estrategia de carrera, entendida como el conjunto de decisiones sobre cuándo entrar a boxes, qué compuesto de neumático montar y a qué ritmo gestionar cada stint, puede alterar radicalmente el orden de llegada. Esta dimensión táctica convierte cada Gran Premio en un problema de optimización combinatoria de alta complejidad, en el que los equipos de ingenieros de estrategia deben tomar decisiones en tiempo real con información incompleta y bajo condiciones cambiantes.

#### 2.1.1. El problema de la parada en boxes

El reglamento deportivo de la FIA obliga a los pilotos a utilizar al menos dos especificaciones de neumático distintas durante la carrera [2, Art. B6.3.6]. Esta

restricción convierte la elección del número y momento de las paradas en boxes en la palanca de control estratégica central de cada equipo. La degradación de los neumáticos es la variable que gobierna esta decisión: a medida que el compuesto pierde adherencia, el tiempo de vuelta crece de forma no lineal, y en algún momento la pérdida de rendimiento supera el coste temporal de una parada en boxes, que en F1 oscila entre 20 y 25 segundos incluyendo la trayectoria del pit lane.

La dificultad del problema reside en la combinación de múltiples factores interdependientes. En primer lugar, la degradación del neumático es una función no lineal de la edad del compuesto, la temperatura de la pista, el estilo de conducción del piloto y la carga aerodinámica del vehículo [3, 4]. En segundo lugar, los eventos estocásticos (coches de seguridad, banderas rojas, lluvia repentina) invalidan en segundos cualquier plan estratégico precalculado. En tercer lugar, la decisión de un equipo afecta directamente a los demás: reaccionar a una parada del rival puede o no ser óptimo dependiendo de la diferencia de ritmo real y el tráfico que se encontrará en pista tras la parada. Esta interdependencia estratégica introduce componentes propias de la teoría de juegos en un problema que de otro modo podría tratarse como optimización individual [5].

### 2.1.2. El ritmo de vuelta como variable clave

La predicción del ritmo de vuelta es el componente operativo central de cualquier sistema de análisis estratégico. Consiste en estimar el tiempo que tardará un piloto en completar cada vuelta en función de la edad del neumático, el compuesto montado, las condiciones meteorológicas y el contexto de carrera. Esta capacidad predictiva es la que permite proyectar cuándo el neumático actual habrá degradado lo suficiente como para que sea ventajosa una parada, y qué tiempo de carrera total se espera bajo cada estrategia alternativa.

Los equipos de Fórmula 1 han desarrollado modelos de ritmo propietarios de gran sofisticación, alimentados por sensores de telemetría propios y datos históricos de comportamiento del neumático. *RaceScope* aborda este mismo problema desde una perspectiva open-source y sobre datos públicos; la sección siguiente describe la fuente de datos empleada y las secciones posteriores los pilares teóricos sobre los que se construye su modelo de ritmo y su motor de evaluación de estrategias.

## 2.2. Fuente de Datos: OpenF1

*OpenF1* [1] es una API gratuita y de código abierto que proporciona datos históricos y en tiempo real de Fórmula 1. Expone tiempos de vuelta, telemetría de vehículos, información de pilotos y mensajes de control de carrera en formatos JSON y CSV, a partir de los canales oficiales de la Formula One Management (FOM).

La API registra datos durante las sesiones de carrera a distintas granularidades temporales: los tiempos de vuelta y los cambios de stint se capturan vuelta a vuelta; la telemetría vehicular se muestrea a 10 Hz. El histórico disponible desde la temporada 2023 permite el análisis retrospectivo de carreras y la comparación de rendimiento entre temporadas.

Los datos de telemetría incluyen la posición del acelerador y freno, la activación del DRS, los cambios de marcha y la velocidad instantánea. Esta resolución permite identificar patrones de comportamiento individuales del piloto que el tiempo de vuelta no captura.

La API también integra información estructurada sobre pilotos, equipos y sus afiliaciones, junto con métricas de desempeño histórico que contextualizan el análisis competitivo.

Para el proyecto *RaceScope*, OpenF1 constituye la fuente primaria de datos, permitiendo la ingesta sistemática de características de carrera necesarias para el entrenamiento de modelos de predicción de tiempos de vuelta y la evaluación cuantitativa de estrategias de paradas en boxes. Como alternativa complementaria existe *FastF1* [6], una librería Python que encapsula datos de la API Ergast (deprecada en 2024) y de los canales de telemetría oficiales, aunque su granularidad temporal es inferior a la de OpenF1 para datos en tiempo real.

### 2.2.1. Dominios de datos relevantes para RaceScope

OpenF1 organiza la información por sesión en cuatro dominios que cubren, juntos, todo lo que el sistema necesita para reconstruir el ritmo de una carrera. La tabla 2.1 resume cada dominio con su granularidad temporal, la información principal que aporta y el papel que cumple en *RaceScope*.

Tabla 2.1: Dominios de información disponibles en la API de OpenF1 y su uso en *RaceScope*.

<b>Dominio</b>	<b>Granularidad</b>	<b>Información principal</b>	<b>Uso en <i>RaceScope</i></b>
Telemetría del vehículo	10 Hz continuo	Accelerador, freno, marcha, RPM, velocidad instantánea, estado de DRS	Características de conducción ( <i>push index</i> , agresividad) que alimentan al Transformer como contexto del estilo del piloto.
Vueltas	1 por vuelta	Tiempo de vuelta, tiempos de sector, velocidad de paso, marcas de <i>pit-in</i> y <i>pit-out</i>	Variable objetivo principal del modelo de predicción de ritmo.
Stints y compuestos	1 por stint	Compuesto montado, vuelta de inicio y de fin del stint, edad inicial del neumático	Etiquetado de cada vuelta con su compuesto y su edad de neumático; base para el modelo de degradación paramétrica.
Meteorología	1 muestra por minuto	Temperatura de pista, temperatura del aire, humedad, lluvia, viento	Contexto térmico de la sesión, decisivo para la degradación y la sensibilidad de los compuestos.

Las cuatro fuentes son complementarias en escala temporal. La telemetría aporta resolución submétrica del comportamiento del piloto y del vehículo, mientras que las vueltas y los stints condensan esa información en agregados por vuelta que constituyen el eje central de la ingeniería de características. La meteorología, muestreada al minuto, fija las condiciones de contorno bajo las que se interpreta la degradación de cada compuesto.

El almacenamiento de los datos en formato Apache Parquet [7], un formato columnar orientado a consultas analíticas, permite cargar únicamente las columnas necesarias

en cada fase y mantiene acotado el uso de memoria durante el entrenamiento de modelos y la ejecución del motor de simulación.

## 2.3. Transformers para la predicción de tiempos de vuelta

La predicción del tiempo de vuelta como función de la edad del neumático, el compuesto montado y las condiciones de carrera es un problema de regresión sobre series temporales: cada carrera genera una secuencia ordenada en la que el valor futuro depende del historial reciente. *RaceScope* aborda este problema con un encoder Transformer entrenado de forma específica por piloto.

La arquitectura Transformer, propuesta originalmente por Vaswani *et al.* para tareas de traducción [8], sustituye la recurrencia por capas de atención multi-cabeza, en las que cada elemento de la secuencia atiende a todos los demás con pesos aprendidos dinámicamente. Para un input  $\mathbf{X} \in \mathbb{R}^{T \times d}$  de longitud  $T$  y dimensión  $d$ , una cabeza de atención escala-producto computa:

$$\begin{aligned} \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \mathbf{V}, \\ \mathbf{Q} &= \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V \end{aligned} \tag{2.1}$$

donde las proyecciones  $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$  son parámetros entrenables. Apilando  $H$  cabezas en paralelo y proyectando su concatenación se obtiene la atención multi-cabeza, que se intercala con redes feed-forward en cada bloque del encoder.

A esta operación se le suman tres elementos arquitectónicos centrales para el rendimiento del Transformer en problemas de regresión sobre secuencias temporales:

- **Codificación posicional.** La atención es invariante a permutaciones, de modo que sin información adicional el modelo no distingue el orden de los elementos de la secuencia. La propuesta original añade una codificación posicional sinusoidal a las representaciones de entrada [8]; trabajos posteriores introducen codificaciones aprendidas durante el entrenamiento o variantes relativas, que se ajustan mejor a longitudes de secuencia fijas.
- **Variantes Pre-LN y Post-LN.** Las dos variantes difieren en la ubicación de la normalización por capa dentro de cada bloque del encoder [9]. La variante Pre-LN, que aplica la normalización antes de la atención y de la red

feed-forward, presenta una optimización más estable y permite entrenar redes profundas sin un calentamiento agresivo de la tasa de aprendizaje.

- **Compuertas multiplicativas.** Las compuertas multiplicativas son un mecanismo simple para condicionar la representación interna del modelo a una entrada externa, casi siempre categórica.

La idea consiste en construir un vector  $\mathbf{g} \in (0, 1)^d$  del mismo tamaño que la representación y multiplicarlo componente a componente con cada paso temporal de la secuencia,  $\mathbf{x}_t \leftarrow \mathbf{x}_t \odot \mathbf{g}$ . Cada dimensión queda amplificada cuando su componente en  $\mathbf{g}$  se acerca a 1, mantenida en torno a 0.5 y silenciada cuando se acerca a 0. La inicialización habitual (proyección lineal seguida de normalización por capa y sigmoide) hace que el vector parta de  $\mathbf{g} \approx 0,5$ , equivalente a no modular nada al inicio del entrenamiento; la especialización emerge progresivamente y el coste computacional añadido es despreciable frente al de las capas de atención.

El entrenamiento de un encoder Transformer en regímenes de pocos datos descansa sobre tres decisiones de optimización con base teórica establecida. La pérdida de Huber [10] combina el comportamiento cuadrático de la MSE para errores pequeños con el lineal de la MAE para errores grandes, lo que reduce la influencia de vueltas atípicas (pit-out, safety car) sin perder diferenciabilidad.

El optimizador AdamW [11] desacopla la regularización L2 del paso de actualización, evitando la interacción no deseada entre los términos de momento y de regularización presente en la formulación original de Adam.

El planificador OneCycleLR [12] sustituye un calentamiento manual de la tasa de aprendizaje por una trayectoria triangular (subida, bajada y recocido final) que acelera la convergencia y reduce la sensibilidad al valor inicial.

Sobre esta base, tres propiedades de la familia Transformer justifican su adopción en el problema del ritmo de vuelta:

- **Contexto global por defecto:** cada vuelta de la ventana atiende a todas las demás sin restricciones de orden, lo que captura interacciones a distancia (la vuelta de la última parada influye explícitamente sobre la actual).
- **Aprovechamiento del par (circuito, compuesto):** en cada simulación, los dos identificadores que la compuerta toma como entrada (el circuito y el compuesto montado) son constantes a lo largo de toda la ventana de contexto. El comportamiento de un neumático duro en Mónaco no tiene nada que ver con el de uno duro en Monza, y un sesgo aditivo (sumar un vector a la representación) no puede atenuar dimensiones relevantes en un trazado e

irrelevantes en otro: solo puede desplazarlas. Una compuerta multiplicativa sí puede, y lo hace una sola vez por secuencia, sin gastar capacidad de atención en repropagar los mismos identificadores en cada vuelta.

- **Cabezas auxiliares:** el encoder puede atacar simultáneamente varios objetivos (delta de ritmo respecto al stint, tiempo absoluto, tasa de degradación, coste remanente del stint) compartiendo representación, lo que regulariza el entrenamiento sin necesidad de dropout.

El argumento histórico contra los Transformers, que requieren mucho dato para generalizar bien, se invierte aquí. El objetivo del modelo no es generalizar: es memorizar con precisión el comportamiento de un piloto concreto en el subespacio de combinaciones (circuito, compuesto) que ha rodado de verdad.

Por eso el entrenamiento se hace por piloto y con *dropout* cero, y por eso la compuerta multiplicativa encaja de manera natural en este planteamiento: una **g** específica por cada par equivale a un régimen distinto para cada combinación (Mónaco-HARD, Monza-HARD, etc.), de modo que la red no aprende un único mapa universal de degradación sino un atlas de mapas, uno por par. Esta especificidad es lo que el sistema necesita para predecir el ritmo en regímenes ya observados, no en regímenes hipotéticos para los que el modelo no tiene base empírica.

El Capítulo 4 detalla la configuración concreta de la red ( $d_{\text{model}} = 384$ , 8 capas, contexto de 40 vueltas, 15 features continuos) y la trayectoria de diseño (v1  $\rightarrow$  v2  $\rightarrow$  v3) que llevó a esa configuración.

## 2.4. Modelos lineales jerárquicos para regímenes de datos limitados

Cuando un problema de regresión se desagrega en muchas categorías y el número de observaciones por categoría es bajo, el ajuste independiente por categoría tiende al sobreajuste y produce coeficientes inestables. Los modelos lineales jerárquicos resuelven esta limitación mediante el concepto de *partial pooling*: los coeficientes estimados para un subgrupo con pocos datos se regularizan hacia los del nivel inmediatamente superior de la jerarquía, en lugar de estimarse de forma totalmente independiente [13].

Una implementación práctica habitual sustituye el ajuste bayesiano completo por una jerarquía explícita de niveles con *fallbacks*: si un nivel concreto carece de observaciones suficientes, los parámetros se estiman sobre un nivel más agregado. El Capítulo 4 aplica este principio al perfil paramétrico de degradación de neumáticos,

definiendo cuatro niveles desde el más específico (piloto, circuito y compuesto) hasta el más general (valores de seguridad por defecto).

## 2.5. Simulación Monte Carlo y optimización bajo incertidumbre

Una vez disponible un modelo de ritmo de vuelta, el problema de la estrategia de carrera puede reformularse como la búsqueda de la secuencia de paradas en boxes (número de paradas, vueltas en que se realizan y compuestos a montar) que minimiza el tiempo total de carrera esperado. La dimensionalidad de este espacio de búsqueda es manejable para estrategias de una o dos paradas (del orden de decenas a centenares de candidatos), pero el carácter estocástico del entorno hace que la evaluación determinista de cada candidato sea insuficiente para una toma de decisiones robusta.

### 2.5.1. Fundamentos del método Monte Carlo

El método de Monte Carlo, formalizado por Rubinstein y Kroese, proporciona un marco general para estimar el valor esperado de funciones de variables aleatorias mediante la media empírica de muestras generadas por simulación [14]. Dado un espacio de probabilidad  $(\Omega, \mathcal{F}, \mathbb{P})$  y una función de coste  $f(\mathbf{x}, \xi)$  que depende de la estrategia  $\mathbf{x}$  y de los valores de las variables aleatorias  $\xi$  (ej. duración del pit stop, tiempo de safety car, variabilidad del ritmo), el estimador Monte Carlo de  $\mathbb{E}_\xi[f(\mathbf{x}, \xi)]$  con  $N$  simulaciones es:

$$\hat{F}_N(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N f(\mathbf{x}, \xi^{(n)}), \quad \xi^{(n)} \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_\xi \quad (2.2)$$

La convergencia del estimador es proporcional a  $1/\sqrt{N}$ , lo que implica que reducir el error a la mitad requiere cuadruplicar el número de simulaciones [14].

### 2.5.2. Clasificación de estrategias bajo incertidumbre

En el contexto de la estrategia de carrera, no basta con seleccionar la estrategia de menor tiempo esperado: una estrategia con media ligeramente superior pero varianza mucho menor puede ser preferible en un contexto competitivo donde la

consistencia es tan valiosa como el ritmo absoluto. Este criterio de compromiso entre media y riesgo se formaliza mediante el *risk score*:

$$\rho(\mathbf{x}) = \mathbb{E}[T(\mathbf{x})] + \lambda \cdot \text{Var}[T(\mathbf{x})] \quad (2.3)$$

donde  $T(\mathbf{x})$  es el tiempo total de carrera bajo la estrategia  $\mathbf{x}$ , y  $\lambda \geq 0$  es el parámetro de aversión al riesgo. Esta formulación es análoga al marco de media-varianza de Markowitz en finanzas [15], adaptada al dominio de la optimización de estrategias de carrera. Con  $\lambda = 0$  el criterio colapsa al tiempo esperado puro; valores crecientes de  $\lambda$  penalizan progresivamente la variabilidad de la estrategia. El Capítulo 4 desarrolla cómo este criterio se integra en el motor de estrategia de *RaceScope* y qué valor de  $\lambda$  se adopta como referencia.

## 2.6. Trabajos relacionados

El apoyo a la decisión estratégica en Fórmula 1 ha sido abordado por la literatura desde dos vías que, por construcción, no compiten con *RaceScope*: sistemas privados de los equipos que ejecutan en tiempo real durante la carrera sobre telemetría propietaria, y trabajos académicos sobre simuladores cerrados que no son reproducibles sin recrear el simulador. *RaceScope* ocupa un espacio distinto: pre-carrera, sobre datos públicos (OpenF1), completamente open-source, y con un motor de tres componentes (resolución analítica de la vuelta de pit, refinamiento Monte Carlo con el Transformer y optimizador de cartera estratégica) pensado para una audiencia más amplia que la de los ingenieros de los equipos.

### 2.6.1. Software y herramientas existentes

Las herramientas accesibles que cubren parcial o totalmente el flujo *datos públicos*  $\rightarrow$  *predicción de ritmo*  $\rightarrow$  *recomendación de estrategia* se agrupan en cuatro categorías:

- **Simuladores académicos de código abierto.** El simulador de carrera publicado por Heilmeyer *et al.* (TU Munich) [16, 17] es el referente público más próximo a un motor de estrategia completo. Enumera planes de paradas, modela la degradación con un ajuste paramétrico lineal por compuesto y evalúa cada plan mediante un esquema de Monte Carlo con eventos estocásticos de safety car. La predicción del ritmo descansa en ajustes lineales por compuesto y piloto, sin componente de aprendizaje sobre telemetría. El trabajo previo

de Bekker y Lotz [18] planteó el problema como una simulación de eventos discretos, sin distribuir código abierto reproducible.

- **Librerías de acceso a datos.** *FastF1* [6], mencionada en 2.2 como complemento a *OpenF1*, ofrece descarga normalizada de vueltas, telemetría y eventos, así como utilidades de visualización por sesión. No incorpora motor de estrategia: el usuario obtiene los datos crudos y debe construir su análisis sobre ellos.
- **Herramientas de análisis para aficionados.** *F1 Tempo* [19] y proyectos similares centran su valor en la comparativa interactiva de ritmos por vuelta entre pilotos y compuestos. La parada en boxes se trata como un dato observado a representar, no como una decisión a optimizar.
- **Sistemas de telemetría retransmitida.** *F1 Insights powered by AWS* [20] produce gráficos predictivos integrados en la realización televisiva, como la predicción de la *pit window* o la probabilidad de adelantamiento. El modelo subyacente es propietario y no se distribuye; su salida es información narrativa para la retransmisión, no una recomendación con curvas de ritmo y varianzas explícitas. El predictor de estrategia de Pirelli cumple un papel comparable durante el fin de semana de carrera y se publica como comunicado para medios.

Frente a este panorama, *RaceScope* cubre cuatro funcionalidades que no aparecen reunidas en ninguna herramienta accesible. Primero, un modelo Transformer entrenado por piloto que captura no linealidades de degradación condicionadas al circuito y al compuesto activo (2.3), allí donde los simuladores académicos públicos se limitan a un ajuste lineal por compuesto.

Segundo, una hibridación entre evaluación analítica y refinamiento Monte Carlo: la vuelta óptima de cada parada se resuelve en forma cerrada a partir del perfil paramétrico y el coste de Monte Carlo se reserva para refinar las candidatas mejor posicionadas, mientras que las alternativas open source aplican Monte Carlo sobre la enumeración completa de planes.

Tercero, un ranking sensible al riesgo con criterio media-varianza (2.5.2) que devuelve, junto al tiempo esperado de cada estrategia, una medida explícita de su variabilidad, frente a la salida puntual que ofrecen las herramientas para aficionados y los gráficos de retransmisión. Y finalmente, una distribución *open source* ejecutable sobre un equipo de sobremesa, con entrenamiento y generación de estrategias reproducibles a partir del repositorio.

# Capítulo 3

## Metodología

Este capítulo presenta la visión arquitectónica de *RaceScope*: qué componentes integran el sistema, cómo se relacionan entre sí y con qué criterios de diseño se han combinado para dar respuesta al problema de la estrategia de carrera en Fórmula 1. El desarrollo detallado de cada componente (pipeline de datos, modelos de predicción, motor de estrategia y métricas de evaluación) se aborda en el Capítulo 4.

### 3.1. Visión general del sistema

*RaceScope* es un sistema *end-to-end* de exploración de estrategias de carrera que combina aprendizaje profundo, modelado paramétrico y simulación estocástica para generar y rankear estrategias de pit stop previas a una carrera. La Figura 3.1 muestra los seis bloques funcionales del sistema y los flujos de datos que los conectan, desde la fuente pública de OpenF1 hasta la aplicación web que entrega los resultados al usuario.

Los seis bloques que componen el sistema son los siguientes:

1. **Fuente de datos: *OpenF1*** [1]. API pública que proporciona tiempos de vuelta, stints y compuestos de neumático, telemetría del vehículo y datos meteorológicos por sesión.
2. **Ingestión y preprocesamiento.** Un *pipeline* Python descarga los datos sesión a sesión, los valida y los transforma en una *feature store* columnar en formato Apache Parquet [7].

3. **Perfil paramétrico de piloto.** Modelo de regresión lineal con cuatro coeficientes que captura la degradación de neumáticos por piloto, circuito y compuesto, con jerarquía de *fallback* de cuatro niveles. Alimenta al Motor 1.
4. **Modelo de ritmo Transformer.** Encoder de atención multi-cabeza entrenado por piloto que predice el tiempo de vuelta a partir de una ventana de contexto de 40 vueltas, con un modelo global de respaldo para pilotos con datos insuficientes. Alimenta al Motor 2.
5. **Motor de estrategia con tres motores que colaboran.** El perfil paramétrico resuelve la vuelta óptima de pit por *break-even* cerrado (Motor 1); el Transformer refina las  $K$  mejores candidatas por Monte Carlo (Motor 2); un optimizador de cartera estratégica ordena el conjunto con un criterio media-varianza (Motor 3). Una cota de validez física, derivada de la envoltura empírica del circuito, acota la salida del motor antes de exponerla.
6. **API y aplicación web.** Una API FastAPI expone el resultado del motor a una SPA React + Vite, que lo presenta con un carrusel de estrategias y visualizaciones interactivas de curvas de ritmo y ventanas de pit.

### 3.2. Pipelines offline y online

El sistema opera en torno a dos pipelines bien diferenciados que separan el trabajo computacionalmente intensivo del que debe ejecutarse en tiempo de respuesta.

El **pipeline offline** (entrenamiento) se ejecuta una sola vez por temporada: ingesta los datos de OpenF1, construye la *feature store*, entrena los modelos Transformer por piloto y ajusta los perfiles paramétricos. Este pipeline puede tardar varios minutos y se ejecuta como proceso independiente antes de arrancar el servidor.

El **pipeline online** (inferencia) responde a cada petición de estrategia: carga los modelos entrenados, construye el contexto de carrera, genera las candidatas por *break-even* cerrado a partir del perfil lineal, las ordena con el optimizador de cartera estratégica y refina las  $K$  mejores con el Transformer en simulación Monte Carlo. El diseño de caché multinivel (predicciones del Transformer en memoria con política LRU, curvas de ritmo en disco con TTL de 24 h, respuestas completas en memoria) mantiene la latencia bajo unos pocos segundos en el régimen caliente.

Adicionalmente, el sistema soporta dos modos de datos: **modo snapshot** (por defecto), que sirve el histórico local ya procesado; y **modo live**, que consultaría OpenF1 en tiempo real con caída automática al snapshot si la API no está disponible.

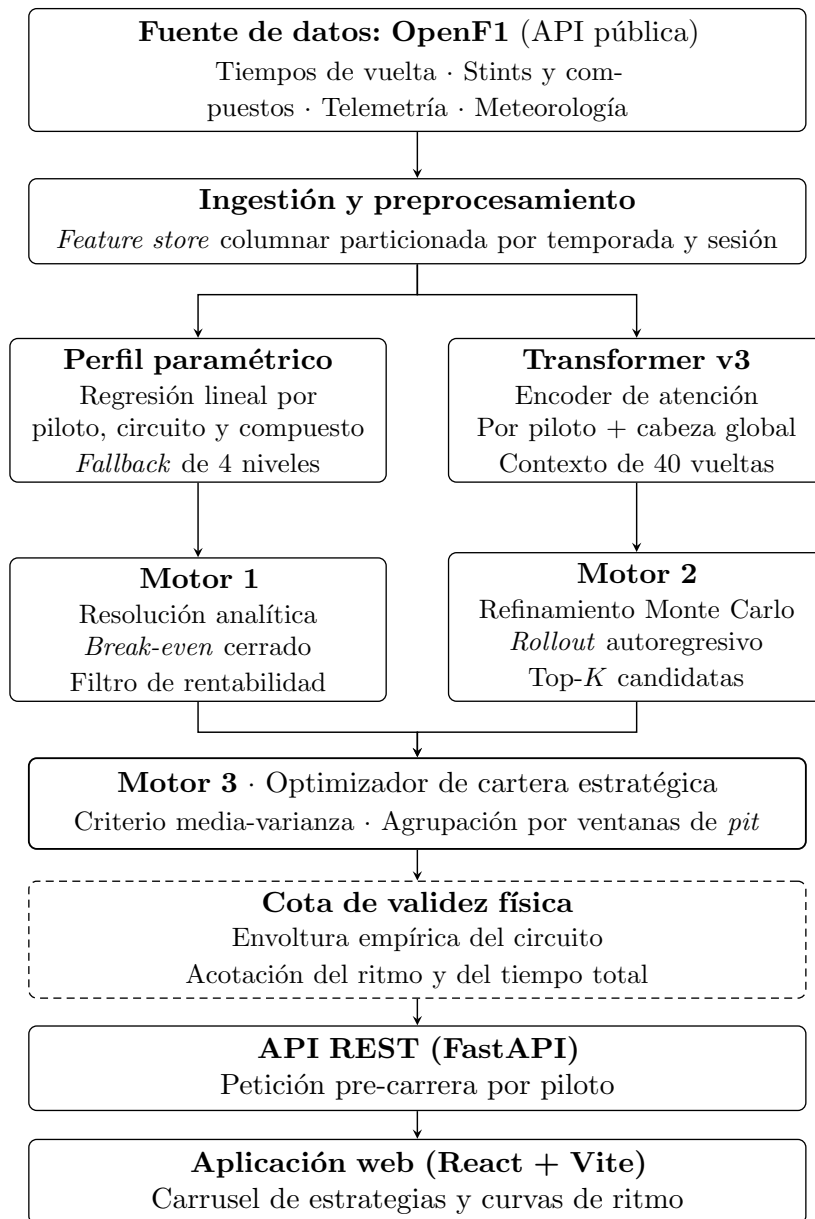


Figura 3.1: Arquitectura de *RaceScope*: flujo de datos desde OpenF1 hasta la API y la aplicación web. Incluye ingesta, dos modelos de ritmo (perfil paramétrico y Transformer), tres motores de estrategia y una cota de validez física. Elaboración propia.



# Capítulo 4

## Caso de Estudio: RaceScope

Este capítulo desarrolla en detalle cada componente del sistema *RaceScope* presentado en el Capítulo 3: el pipeline de datos, el modelo Transformer de predicción de ritmo, el perfil paramétrico por piloto, los tres motores que componen el módulo de estrategia y los criterios de evaluación empleados. El propósito es describir las decisiones de diseño concretas que se adoptaron, con los parámetros específicos del sistema.

### 4.1. Pipeline de datos

El pipeline de datos transforma los registros brutos de OpenF1 en una *feature store* lista para el entrenamiento de modelos. Se articula en dos etapas secuenciales: ingestión y preprocesamiento.

#### 4.1.1. Ingestión de datos

La ingestión descarga, para cada año solicitado, los datos de las sesiones de carrera (Grand Prix), sprint y segundo entrenamiento libre (FP2), que son las que aportan el volumen de vueltas en condiciones de degradación real necesario para el entrenamiento. De cada sesión se recuperan cuatro tipos de información: vueltas, stints y compuestos, pilotos y condiciones meteorológicas.

El cliente HTTP implementa dos mecanismos de control de flujo para operar con la API de OpenF1 dentro de sus límites de tasa. El primero es un **token bucket** que regula el ritmo sostenido a 60 peticiones por minuto con un intervalo mínimo entre

peticiones de 0.167 s. El segundo es un **reintento con *backoff* exponencial**: ante errores transitorios (429 o 5xx), el cliente espera  $2^n$  segundos antes de reintentar, con un máximo de 5 intentos. La ingestión de sesiones ya descargadas se omite en ejecuciones posteriores, lo que permite reanudar ingestiones interrumpidas sin duplicar datos.

Los datos se persisten en disco en una jerarquía particionada por temporada y sesión. El uso del formato Apache Parquet [7], un formato columnar optimizado para consultas analíticas con compresión integrada, permite cargar únicamente las columnas necesarias en cada fase del pipeline, reduciendo el uso de memoria en el preprocesamiento.

### 4.1.2. Preprocesamiento e ingeniería de características

El preprocesamiento unifica los datos de vueltas, stints y meteorología en una única tabla de características por año. Las transformaciones principales son:

**Fusión de vueltas y stints.** Cada vuelta se asocia a su stint correspondiente mediante el par (piloto, número de stint). Si el número de stint no está disponible directamente, se aplica una unión por solapamiento a partir de los rangos de vuelta inicial y final de cada stint.

**Cálculo de la edad del stint.** Para cada vuelta se calcula su posición relativa dentro del stint activo:  $\text{edad\_de\_stint} = \text{vuelta\_actual} - \text{vuelta\_inicial} + 1$ . Esta variable captura directamente el estado de desgaste del neumático en el momento de la vuelta.

**Canonicalización del identificador de circuito.** Los nombres de circuito provistos por OpenF1 pueden variar entre temporadas. El preprocesador aplica un mapa canónico de nombres cortos a identificadores normalizados (p.ej., «sakhir» a «Sakhir»), con *fallback* a conversión en mayúsculas si el circuito no está en el mapa.

**Imputación de temperaturas.** Los valores ausentes de temperatura de pista o de aire se imputan con la media de la sesión correspondiente, preservando la coherencia de la secuencia temporal sin introducir valores extremos.

**Filtrado de compuestos.** Solo se conservan las vueltas con compuesto perteneciente al conjunto {SOFT, MEDIUM, HARD}, descartando vueltas con neumáticos de lluvia o datos de compuesto no disponibles, que presentan una dinámica de degradación radicalmente distinta.

El Transformer recibe quince características continuas por vuelta, tres campos

categoricos por paso temporal (compuesto, tipo de vuelta, tipo de sesión) y dos identificadores estáticos por secuencia (circuito y compuesto activo, que alimentan la compuerta multiplicativa descrita en 4.2.1). La Tabla 4.1 detalla las quince características continuas; todas se normalizan en preprocesamiento de forma que entran al modelo con escala comparable.

Las tres características categóricas (compuesto, tipo de vuelta y tipo de sesión) se proyectan mediante embeddings entrenables de 24, 8 y 8 dimensiones, respectivamente, y se concatenan al vector continuo. El identificador estático de circuito utiliza un embedding de 48 dimensiones replicado a lo largo de la ventana; el del compuesto activo se reserva para la compuerta multiplicativa y aporta 24 dimensiones adicionales.

La variable objetivo principal es el delta normalizado del tiempo de vuelta respecto a la mediana del stint, en *z-score* global. La normalización *z-score* se prefiere a la min-max porque los tiempos de vuelta presentan colas pesadas (pit-out, safety car) que la min-max amplificaría, y porque el entrenamiento es más estable cuando la variable objetivo tiene varianza unitaria.

La tabla de características resultante se mantiene junto con los metadatos de pilotos, equipos y circuitos correspondientes a cada temporada, lista para alimentar el entrenamiento del Transformer y la inferencia del motor de estrategia.

## 4.2. Modelo de predicción de tiempos de vuelta

El modelo de predicción de tiempos de vuelta resuelve un problema de **regresión secuencia a escalar**: dada una ventana de las últimas  $W = 40$  vueltas de un piloto en una sesión, predice el delta del tiempo de vuelta siguiente respecto a la mediana del stint. La formulación secuencial es natural para la degradación de neumáticos: el ritmo de la vuelta actual depende del historial reciente de carga mecánica acumulada, que no es observable directamente pero sí deducible de la secuencia previa.

### 4.2.1. Arquitectura Transformer

El modelo es un encoder Transformer Pre-LN [8, 9] entrenado de forma específica por piloto. Su arquitectura se organiza en cinco bloques:

1. **Embeddings categóricos por paso temporal.** Los tres identificadores discretos del paso temporal (compuesto, tipo de vuelta y tipo de sesión)

Tabla 4.1: Características continuas de entrada del modelo Transformer de predicción de tiempos de vuelta

Característica	Categoría	Descripción y normalización
<code>stint_age_norm</code>	Stint	Vueltas desde el inicio del stint, divididas entre 40
<code>delta_norm</code>	Ritmo	Delta de tiempo respecto al stint, normalizado por su desviación típica global
<code>pace_intent_norm</code>	Ritmo	Diferencia entre la vuelta y la mediana del stint, normalizada por la dispersión global; valores negativos indican ataque
<code>gap_norm</code>	Tráfico	Gap al coche delantero, recortado a 5 s y normalizado
<code>track_temp_norm</code>	Térmico	Temperatura de pista centrada en su referencia por circuito y dividida entre 10
<code>air_temp_norm</code>	Térmico	Temperatura del aire centrada en su referencia y dividida entre 8
<code>throttle_dev</code>	Telemetría	Desviación normalizada del acelerador medio respecto a la media (circuito, compuesto)
<code>brake_dev</code>	Telemetría	Desviación normalizada del freno medio respecto a la media (circuito, compuesto)
<code>speed_dev</code>	Telemetría	Desviación normalizada de la velocidad media respecto a la media (circuito, compuesto)
<code>rpm_norm</code>	Telemetría	Régimen medio del motor centrado en 10 000 y dividido entre 3 000
<code>drs_dev</code>	Telemetría	Desviación normalizada de la fracción de DRS respecto a la media (circuito, compuesto)
<code>push_index</code>	Conducción	Combinación ponderada de <code>throttle_dev</code> , <code>speed_dev</code> , <code>brake_dev</code> y <code>drs_dev</code> , recortada a $[-3, 3]$
<code>race_lap_norm</code>	Carrera	Vuelta de la sesión dividida entre el número esperado de vueltas del circuito
<code>lap_speed_deviation</code>	Carrera	Desviación normalizada de la velocidad máxima de la vuelta respecto a la media del circuito
<code>stint_number_norm</code>	Carrera	Número de stint dentro de la carrera, recortado y normalizado en $[0, 1,5]$

se proyectan a embeddings de 24, 8 y 8 dimensiones, respectivamente. El identificador de circuito, estático por secuencia, aporta un embedding adicional de 48 dimensiones replicado a lo largo de la ventana.

2. **Proyección dual a  $d_{\text{model}}$ .** El vector de 15 características continuas se proyecta linealmente a  $\mathbb{R}^{d_{\text{model}}/2}$ , y el vector concatenado de embeddings categóricos a otro  $\mathbb{R}^{d_{\text{model}}/2}$ . Ambos se concatenan para formar la representación de dimensión completa  $d_{\text{model}} = 384$ .
3. **Compuerta multiplicativa circuito  $\times$  compuesto.** Los embeddings estáticos del circuito y del compuesto activo se concatenan y proyectan a un vector de  $d_{\text{model}}$  dimensiones, sobre el que se aplica una normalización por capa y una sigmoide. El resultado actúa como compuerta multiplicativa que reescala cada dimensión del modelo en función del par activo. La inicialización con Layer-Norm + sigmoide hace que el factor de modulación parta de un valor neutro próximo a 0.5 y se especialice progresivamente durante el entrenamiento.
4. **Codificación posicional y encoder.** Tras una normalización de capa inicial, se añade una codificación posicional aprendida sobre las 40 posiciones, y la secuencia atraviesa un encoder de 8 capas Pre-LN con 8 cabezas de atención y red feed-forward interna de 1 536 unidades.
5. **Cuatro cabezas de salida.** Se toma la representación del último paso temporal  $\mathbf{h}_T \in \mathbb{R}^{d_{\text{model}}}$  y se proyecta linealmente a cuatro escalares: el delta normalizado de la vuelta siguiente (cabeza principal, entrenada con pérdida de Huber), el tiempo absoluto normalizado, la tasa de degradación instantánea y el coste remanente del stint (las tres últimas como cabezas auxiliares, entrenadas con MSE).

La arquitectura resultante acumula aproximadamente  $14,3 \times 10^6$  parámetros, dominados por el bloque de atención multi-cabeza. El conjunto de pesos por piloto se serializa en disco y ocupa del orden de 56 MB.

### 4.2.2. Estrategia de entrenamiento y modelos de fallback

La función de pérdida combina la cabeza principal con las tres auxiliares:

$$\begin{aligned}
 \mathcal{L} = & \mathcal{L}_{\text{Huber}}(\hat{\Delta}, \Delta) \\
 & + 0,10 \text{ MSE}(\hat{t}_{\text{abs}}, t_{\text{abs}}) \\
 & + 0,25 \text{ MSE}(\hat{r}_{\text{deg}}, r_{\text{deg}}) \\
 & + 0,20 \text{ MSE}(\hat{c}_{\text{rest}}, c_{\text{rest}})
 \end{aligned} \tag{4.1}$$

La cabeza principal se entrena con la pérdida de Huber [10], que combina el comportamiento cuadrático cerca del óptimo con el lineal en la cola para reducir la sensibilidad a outliers. El optimizador es AdamW [11] con tasa de aprendizaje base  $\eta = 2 \cdot 10^{-4}$ , sin *weight decay*.

Se aplica un planificador OneCycleLR [12] sobre 30 épocas, con *early stopping* de paciencia 5 monitorizando el MAE de validación, lotes de 64 secuencias y recorte de gradiente a norma máxima 1.0. El dropout se mantiene en cero de forma deliberada: el objetivo del entrenamiento es que el modelo memorice con precisión el comportamiento de un único piloto sobre el conjunto de combinaciones (circuito, compuesto) que ha visto, no generalizar fuera de dominio.

La estrategia sigue una política de **entrenamiento por piloto con fallback global**:

- **Modelo específico por piloto:** se entrena cuando el piloto dispone de un volumen mínimo de vueltas etiquetadas como válidas para entrenamiento.
- **Modelo global de respaldo:** entrenado sobre el conjunto completo de vueltas de todos los pilotos, se utiliza cuando un piloto no supera el umbral de datos o como referencia de calibración para los componentes analíticos del motor de estrategia.

La inferencia denormaliza el delta predicho y lo añade a la mediana de referencia del stint, devolviendo el tiempo de vuelta en segundos.

### 4.2.3. Evolución del modelo: del LSTM al Transformer v3

La configuración descrita en 4.2.1 es la cuarta iteración de una familia de modelos de predicción de ritmo que comenzó con una red recurrente por piloto. Cada salto resuelve una limitación concreta de la versión anterior; las variantes intermedias se conservan como ruta de respaldo para los modelos serializados de versiones anteriores.

- **LSTM (línea base).** Red recurrente Long Short-Term Memory [21] entrenada por piloto sobre el mismo *feature store*. Capturaba la pendiente media de degradación, pero el estado oculto se saturaba en stints largos y apenas reaccionaba al embedding categórico de compuesto. Se conserva como ruta de respaldo para modelos serializados de versiones anteriores.
- **Transformer v1** ( $d_{\text{model}} = 32$ , 2 capas, 4 cabezas, ventana de 15 vueltas, una cabeza de salida,  $\approx 26\,500$  parámetros). Prueba de concepto que validó la viabilidad de un encoder de atención sobre el volumen de datos disponible.

El modelo aprendía la curva de degradación media pero no distinguía entre pilotos: el embedding categórico no se traducía en pendientes de degradación diferenciadas.

- **Transformer v2** ( $d_{\text{model}} = 256$ , 6 capas, 8 cabezas, ventana de 25 vueltas, tres cabezas de salida y término aditivo de circuito,  $\approx 4,76 \times 10^6$  parámetros). Las cabezas auxiliares de tiempo absoluto y tasa de degradación regularizan la representación. El término de circuito se proyecta desde su embedding estático a  $\mathbb{R}^{d_{\text{model}}}$  y se suma a cada paso temporal antes del encoder,  $\mathbf{x}_t \leftarrow \mathbf{x}_t + \mathbf{b}_{\text{circ}}$ , sesgando el delta predicho hacia el ritmo típico de cada trazado. El MAE de validación del modelo común se situó en 0,107 sobre el conjunto de entrenamiento de mayo de 2026. La inyección aditiva resultó insuficiente frente a la interacción real circuito  $\times$  compuesto: un desplazamiento constante de la representación no permite atenuar dimensiones irrelevantes para una combinación dada, y la diferencia entre un compuesto duro y uno blando no admite una traducción lineal entre trazados.
- **Transformer v3** ( $d_{\text{model}} = 384$ , 8 capas, 8 cabezas, ventana de 40 vueltas, cuatro cabezas de salida,  $\approx 14,3 \times 10^6$  parámetros). La iteración más ambiciosa. Sustituye el término aditivo por la compuerta multiplicativa descrita en 4.2.1, que reescala la representación mediante un vector sigmoidal aprendido del par (circuito, compuesto) y permite amplificar o atenuar dimensiones de forma independiente. Amplía el contexto al rango habitual de un stint de carrera, incorpora la característica `stint_number_norm` para distinguir el comportamiento entre stints sucesivos y añade la cabeza de coste remanente, que el Motor 1 reutiliza en el filtro de rentabilidad por parada. El *dropout* se fija en cero de forma deliberada: la generalización se obtiene por separación de modelos entre pilotos, no por regularización dentro de cada uno.

En paralelo a la transición de v2 a v3, el *rollout* autoregresivo del Motor 2 pasó de procesar las simulaciones de Monte Carlo de manera secuencial a tratarlas como un *batch* compartido en el encoder. La implementación de PyTorch del encoder Transformer admite una dimensión de *batch* arbitraria sin necesidad de modificar la arquitectura. La latencia por petición de estrategia pasó del orden de minutos al de segundos, lo que permitió asumir el coste de la arquitectura más grande sin penalizar el tiempo de respuesta del API.

La Tabla 4.2 resume las dimensiones comparables de las tres variantes del Transformer. La v3 es la configuración vigente en *RaceScope*: minimiza el MAE de validación sobre el conjunto de pilotos con datos suficientes y es la que se asume en el resto del capítulo.

Tabla 4.2: Resumen comparativo de las tres variantes del Transformer de predicción de ritmo.

	<b>v1</b>	<b>v2</b>	<b>v3</b>
$d_{\text{model}}$	32	256	384
Capas del encoder	2	6	8
Cabezas de atención	4	8	8
Ventana de contexto (vueltas)	15	25	40
Características continuas	13	14	15
Cabezas de salida	1	3	4
Modulación de circuito	—	aditiva	multiplicativa
Parámetros	$\approx 26\,500$	$\approx 4,76 \times 10^6$	$\approx 14,3 \times 10^6$

### 4.3. Modelo paramétrico de perfil de piloto

En paralelo al Transformer, *RaceScope* mantiene un modelo paramétrico de perfil de piloto que captura la evolución del tiempo de vuelta como función lineal de la antigüedad del stint y de las condiciones térmicas de la sesión. Este perfil cumple dos funciones: alimenta como prior estructural al motor de estrategia y, sobre todo, permite resolver de forma cerrada la vuelta óptima de pit. Sobre su formulación se construye el Motor 1 del módulo de estrategia descrito en 4.4.

#### 4.3.1. Formulación del modelo

El ritmo de vuelta en la vuelta  $k$  de un stint se modela mediante:

$$t_k = \beta_0 + \beta_1 \cdot (k - 1) + \beta_2 \cdot (T_{\text{pista}} - T_{\text{ref,pista}}) + \beta_3 \cdot (T_{\text{aire}} - T_{\text{ref,aire}}) + \varepsilon_k \quad (4.2)$$

donde  $\beta_0$  es el tiempo base del primer stint,  $\beta_1$  la tasa de degradación lineal por vuelta (s/vuelta), y  $\beta_2, \beta_3$  los coeficientes de sensibilidad térmica. Las referencias son  $T_{\text{ref,pista}} = 30 \text{ °C}$  y  $T_{\text{ref,aire}} = 22 \text{ °C}$ , valores representativos de condiciones de seco en F1. Los coeficientes se estiman mediante mínimos cuadrados ordinarios.

La elección de un modelo lineal, frente a modelos de mayor orden o de espacio de estados, responde al principio de parsimonia estadística: en el régimen de datos disponible (decenas a centenares de vueltas por combinación piloto-circuito-compuesto), un modelo con cuatro parámetros presenta menor riesgo de sobreajuste.

### 4.3.2. Jerarquía de fallback de cuatro niveles

Para cada solicitud de ritmo, el sistema resuelve los parámetros del perfil siguiendo una jerarquía de cuatro niveles, del más específico al más general:

1. **Circuito + compuesto:** perfil del piloto en ese circuito y compuesto, si hay al menos 40 vueltas disponibles.
2. **Compuesto del piloto:** perfil sobre todas las vueltas del piloto con ese compuesto (todos los circuitos), si hay al menos 120 vueltas.
3. **Compuesto global:** perfil sobre todas las vueltas de todos los pilotos con ese compuesto, como referencia del comportamiento genérico del compuesto.
4. **Valores de seguridad:**  $\beta_0 = 90$  s,  $\beta_1 = 0,05$  s/vuelta,  $\beta_2 = \beta_3 = 0$ . Este nivel garantiza que el motor nunca produce errores numéricos por falta de datos.

Esta estructura es análoga a los modelos lineales jerárquicos: el perfil global actúa como prior implícito del comportamiento de un piloto genérico [13]. El entrenamiento de perfiles requiere un mínimo de 160 vueltas por piloto.

El nivel 4 admite, de forma opcional, una sobreescritura del ritmo base con la mediana de las vueltas de carrera del circuito para el compuesto solicitado (ecuación 4.6). Cuando está disponible, sustituye al literal  $\beta_0 = 90$  s por un valor coherente con la dinámica real del trazado. Este mecanismo se activa únicamente cuando el dataset no contiene muestra suficiente para entrenar perfiles a los niveles 1 a 3; en los circuitos presentes en el conjunto actual no se utiliza, pero queda disponible ante la incorporación futura de trazados nuevos o pilotos con historial corto.

## 4.4. Motor de estrategia: tres motores que colaboran

El motor de estrategia recibe el contexto de una carrera (año, circuito, piloto) y produce una lista rankeada de estrategias de pit stop. Aplicar Monte Carlo a todas las candidatas sería computacionalmente prohibitivo; por ello, el módulo se organiza como tres motores que se ejecutan en cascada y especializan cada paso de la decisión. El **Motor 1** resuelve de forma cerrada la vuelta óptima de cada parada a partir del perfil paramétrico, y descarta de entrada las candidatas cuya ganancia por neumático nuevo no compensa el tiempo perdido en pit lane. El **Motor 2** refina las mejores candidatas por Monte Carlo con el Transformer, que

predice las curvas de ritmo no lineales y cuantifica la incertidumbre. El **Motor 3**, el *optimizador de cartera estratégica*, ordena el conjunto con un criterio media-varianza inspirado en Markowitz y aplica los filtros de presentación (exclusión de arranques con HARD, agrupación por ventanas de pit, sesgo suave hacia MEDIUM como primer compuesto).

#### 4.4.1. Contexto de carrera

Antes de la evaluación, el motor construye un contexto de carrera con los parámetros empíricos de la sesión: número total de vueltas (máximo histórico observado), temperaturas medias de pista y aire (media de los registros meteorológicos de la sesión), pérdida de pit stop y probabilidad de safety car.

La **pérdida de pit stop** se estima como la diferencia media entre el tiempo de la primera vuelta de un stint y el tiempo esperado sin parada, acotada entre 15 y 45 s. El *fallback* cuando no hay datos suficientes es 22.5 s, coherente con los tiempos históricos en F1.

La **probabilidad de safety car** se estima como la fracción de vueltas en las que el tiempo de vuelta supera en más de un 35 % el tiempo típico de la sesión. El *fallback* es  $P_{SC} = 0,20$ , acotado entre 0.05 y 0.55.

#### 4.4.2. Motor 1: Resolución analítica de la vuelta óptima

El Motor 1 sustituye la enumeración exhaustiva por la solución cerrada del problema de break-even. Sea una estrategia de una parada con compuestos  $A \rightarrow B$  y longitud total  $L$  vueltas. Si  $\text{pace}_A$ ,  $\text{pace}_B$  son los ritmos base del perfil paramétrico para cada compuesto y  $\text{deg}_A$ ,  $\text{deg}_B$  sus tasas lineales de degradación, el tiempo total como función de la vuelta de parada  $s$  se descompone en la suma analítica de dos stints más la pérdida de pit. Derivando e igualando a cero se obtiene:

$$s^* = \frac{(\text{pace}_B - \text{pace}_A) + (\text{deg}_A - \text{deg}_B)/2 + \text{deg}_B \cdot L}{\text{deg}_A + \text{deg}_B} \quad (4.3)$$

Para una estrategia de dos paradas  $A \rightarrow B \rightarrow C$ , el problema admite una formulación análoga en dos variables  $(s_1, s_2)$  que se resuelve como un sistema lineal  $2 \times 2$ :

$$\begin{bmatrix} \text{deg}_A + \text{deg}_B & -\text{deg}_B \\ -\text{deg}_B & \text{deg}_B + \text{deg}_C \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} (\text{pace}_B - \text{pace}_A) + \frac{\text{deg}_A - \text{deg}_B}{2} \\ (\text{pace}_C - \text{pace}_B) + \frac{\text{deg}_B - \text{deg}_C}{2} \\ + (\text{deg}_C - \text{deg}_B) L \end{bmatrix} \quad (4.4)$$

La solución óptima se acota a las ventanas físicamente posibles de cada compuesto (longitudes mínimas de 5 vueltas, máximos derivados del histórico) y se redondea a vuelta entera. Si el sistema lineal es singular, se devuelve la repartición uniforme  $L/3, 2L/3$  como respaldo numérico.

A continuación, el Motor 1 aplica un **filtro de rentabilidad**: para cada parada de la estrategia se calcula la ganancia por neumático nuevo (diferencia entre el tiempo proyectado del compuesto viejo y el del fresco) menos la pérdida de pit. Si alguna de las paradas tiene rentabilidad negativa, la estrategia se descarta antes de pasar a los motores posteriores. Este filtro reduce el espacio de búsqueda a candidatas en las que cada decisión, vista de forma aislada, está justificada.

### 4.4.3. Motor 2: Refinamiento Monte Carlo con el Transformer

Las  $K = 3$  mejores candidatas según el orden analítico se someten a una simulación Monte Carlo con  $N_{\text{sim}} = 100$  rollouts por candidata. Cada rollout simula una realización aleatoria de la carrera con cuatro componentes estocásticas:

1. **Distribución de intención de ritmo.** Se muestrean dos valores, ritmo de ataque y ritmo de conservación, de la distribución empírica construida a partir de los entrenamientos libres del piloto, y se interpolan vuelta a vuelta para alimentar al Transformer.
2. **Evento de safety car.** Variable Bernoulli con probabilidad  $P_{\text{SC}}$  derivada del contexto; si se activa, la vuelta del SC se muestrea uniformemente en el intervalo central de la carrera.
3. **Reducción del pit loss por SC.** Si la parada planificada cae a  $\pm 2$  vueltas del SC, la pérdida efectiva se recorta en hasta 8 s, modelando el aprovechamiento de la neutralización para parar a coste reducido.
4. **Ruido de tráfico.** A cada vuelta del rollout se le suma un componente gaussiano  $\mathcal{N}(\mu_{\text{tráfico}}, \sigma_{\text{tráfico}}^2)$  que representa la variabilidad por adelantamientos y defensa.

El Transformer ejecuta cada rollout en modo autoregresivo: dada la ventana de contexto de 40 vueltas, predice el delta de la siguiente, lo desnormaliza, lo añade al historial y desplaza la ventana. La implementación trata las  $N_{\text{sim}}$  réplicas como un *batch* y comparte el coste de las capas de atención entre todas. El error del estimador empírico de media y varianza decae como  $1/\sqrt{N_{\text{sim}}}$  [14].

#### 4.4.4. Motor 3: Optimizador de cartera estratégica

El Motor 3 produce el orden final entregado al usuario. Sobre cada candidata, tanto las refinadas por Monte Carlo como las restantes con su puntuación analítica, se evalúa el criterio media-varianza:

$$r(\mathbf{x}) = \mathbb{E}[T(\mathbf{x})] + \lambda \cdot \text{Var}[T(\mathbf{x})] \quad (4.5)$$

con  $\lambda = 0,15$  por defecto. La formulación es la adaptación directa del marco de cartera de Markowitz [15] al dominio de la estrategia de carrera: la varianza se trata como riesgo a controlar, no como ruido residual a ignorar.

Antes de devolver el resultado al usuario, el Motor 3 aplica tres operaciones de presentación:

- **Agrupación por ventanas de pit.** Las vueltas de pit se discretizan en cubos de 5 vueltas. Si dos candidatas comparten cubo de pit en todas sus paradas, se considera que representan la misma decisión estratégica y solo se conserva la de menor *risk score*.
- **Exclusión de arranques con HARD.** Las estrategias que arrancan con compuesto duro se filtran del resultado final. La práctica competitiva en F1 hace que ese arranque sea inviable salvo en circunstancias específicas que el sistema no modela.
- **Sesgo suave hacia MEDIUM.** Las estrategias que arrancan con compuesto medio reciben una penalización aditiva de 1 s en su puntuación final, suficiente para no aparecer como primera opción salvo cuando la ventaja analítica es clara.

Tras estas operaciones se devuelven las  $N_{\text{strat}} = 5$  mejores estrategias, cada una acompañada de sus stints, vueltas de pit, ventanas de pit, tiempo esperado, varianza, *risk score* y curvas de ritmo proyectadas por el Transformer. Una semilla aleatoria fija garantiza la reproducibilidad del refinamiento Monte Carlo.

#### 4.4.5. Cota de validez física: envoltura del circuito y acotaciones de seguridad

El modelo paramétrico de 4.3 y el Transformer de 4.2 se entrenan a partir de datos de telemetría que mezclan vueltas de carrera limpias con vueltas in/out, neutralizaciones residuales y sesiones de práctica. Cuando el ajuste por mínimos

cuadrados se aplica a una combinación piloto-circuito-compuesto con muestra reducida o con presencia de ruido estructural, los coeficientes resultantes pueden caer fuera del rango físicamente admisible. Las consecuencias se propagan, sin contención adicional, al motor de estrategia.

### Diagnóstico inicial

Para cuantificar el efecto sobre el sistema en producción se construyó una rutina diagnóstica que, para una temporada y circuito dados, recorre la totalidad de pilotos en parrilla, ejecuta el pipeline completo de generación de estrategias y registra el tiempo total esperado de la estrategia ganadora junto con los parámetros del perfil utilizado. Sobre el Gran Premio de Baréin de 2023 (Sakhir,  $L = 57$  vueltas), las observaciones agregadas mostraron:

- Rango de duraciones totales entre pilotos: [57:42, 3:16:45], equivalente a una dispersión de 139 min (8343s) en una carrera donde el intervalo de tiempos físicamente esperable, en presencia de pilotos doblados, no supera las decenas de segundos.
- Coeficientes térmicos por OLS con magnitudes incompatibles con la sensibilidad reportada para neumáticos de F1: el pico de agarre se sitúa en torno a 105 °C y la pérdida es de aproximadamente un 1% por cada 5 a 10 °C fuera de esa ventana, equivalente a 0,03-0,06 s°C<sup>-1</sup> por vuelta [22]. En 53 de 60 combinaciones piloto-compuesto la corrección térmica efectiva  $\beta_2 \cdot \Delta T_{\text{pista}} + \beta_3 \cdot \Delta T_{\text{aire}}$  excedía los  $\pm 3$  s.
- Intercepts  $\beta_0$  distribuidos en el intervalo [84, 188] s para los compuestos SOFT y MEDIUM, frente a un valor de referencia próximo a 90 s.

### Envoltura empírica del circuito

La solución adoptada deriva una cota de validez directamente de la distribución observada de tiempos de vuelta de carrera del circuito, independiente de los perfiles paramétricos. Para cada compuesto  $c \in \{\text{SOFT}, \text{MEDIUM}, \text{HARD}\}$  se extraen los percentiles

$$\mathcal{E}_c(\text{circuit}) = (t_{\text{fast}}^{(c)}, t_{\text{median}}^{(c)}, t_{\text{slow}}^{(c)}) = (Q_{0,05}, Q_{0,50}, Q_{0,90}) \quad (4.6)$$

calculados sobre el conjunto de vueltas de la sesión de carrera del circuito y del año solicitados, tras un recorte de saneamiento  $0,5 \cdot \tilde{t} < t < 2,0 \cdot \tilde{t}$  (siendo  $\tilde{t}$  la

mediana sin filtrar) que elimina vueltas in/out, periodos de safety car y otros valores estructuralmente atípicos. Si la muestra resultante es menor que 400 vueltas se extiende la búsqueda a todos los años disponibles para el mismo circuito; el respaldo final es una tabla de valores promedio por compuesto.

### Puntos de aplicación

La envoltura se aplica como un límite estricto en cuatro puntos del pipeline, organizados como capas concéntricas de defensa:

1. **Corrección térmica acotada.** Cada uno de los términos  $\beta_2 \cdot \Delta T_{\text{pista}}$  y  $\beta_3 \cdot \Delta T_{\text{aire}}$  se acota individualmente al intervalo  $[-\Delta_T, +\Delta_T]$  con  $\Delta_T = 3$  s. La cota se sitúa por encima del rango fisiológico del compuesto pero por debajo del orden de magnitud de los coeficientes sobreajustados detectados en el diagnóstico. El acotado se aplica tanto en la generación analítica del Motor 1 como en la precomputación de las curvas de ritmo, lo que evita que ambos caminos del motor diverjan en su tratamiento de la temperatura.
2. **Acotación del ritmo base en el Motor 1.** El ritmo  $\text{pace}_{\text{base}}$  resultante para cada compuesto se acota al intervalo

$$[t_{\text{fast}}^{(c)} \cdot \alpha_{\text{lo}}, t_{\text{median}}^{(c)} \cdot \alpha_{\text{hi}}] \quad (4.7)$$

con  $\alpha_{\text{lo}} = 0,99$  y  $\alpha_{\text{hi}} = 1,02$ . El extremo inferior impide que un perfil produzca un ritmo más rápido que el quinto percentil histórico del circuito; el extremo superior limita los excesos del intercept a un 2% sobre la mediana de carrera, margen suficiente para acomodar a los pilotos legítimamente lentos del campo.

3. **Acotación de la curva de ritmo por vuelta.** Se aplica una acotación por vuelta al intervalo

$$[t_{\text{fast}}^{(c)} \cdot \gamma_{\text{lo}}, t_{\text{slow}}^{(c)} \cdot \gamma_{\text{hi}}] \quad (4.8)$$

con  $\gamma_{\text{lo}} = 0,98$  y  $\gamma_{\text{hi}} = 1,10$ , tanto sobre la serie base que sirve de contexto al Transformer como sobre los tiempos que devuelve el modelo. La misma acotación se aplica dentro del Motor 2 a los tiempos producidos por los *rollouts* de las tres variantes del Transformer (v1, v2 y v3), antes de su agregación. Este paso protege la suma  $\sum_k t_k$  que alimenta a la evaluación analítica y a la simulación de estrategias frente a salidas anómalas del modelo en contextos poco representados durante el entrenamiento.

4. **Envoltura del tiempo total.** Antes de añadir cada estrategia al resultado final, el Motor 3 contrasta el tiempo total esperado contra la cota

$$[A \cdot \eta_{\text{lo}} + n_{\text{stops}} \cdot p_{\text{loss}} \cdot 0,85, A \cdot \eta_{\text{hi}} + n_{\text{stops}} \cdot p_{\text{loss}} \cdot 1,15 + \Delta_{\text{SC}}] \quad (4.9)$$

siendo  $A = L \cdot \text{median}_c\{t_{\text{median}}^{(c)}\}$  el anclaje a la mediana de los compuestos del circuito,  $\eta_{lo} = 0,985$ ,  $\eta_{hi} = 1,04$  y  $\Delta_{SC} = 60$  s un margen para neutralizaciones. Cuando el valor cae fuera de la cota se acota al extremo correspondiente y se marca la estrategia como acotada en el resultado, junto con un aviso en el registro de eventos que recoge el contexto completo de la candidata. Si las capas anteriores funcionan, este paso no debería activarse de manera frecuente y los avisos sirven como instrumentación para detectar regresiones de modelo o de datos.

### Validación cuantitativa

La incorporación progresiva de los cuatro mecanismos se midió sobre el mismo escenario de referencia (Sakhir 2023, 20 pilotos) mediante reejecuciones del diagnóstico de 4.4.5. La tabla 4.3 resume la reducción de la dispersión  $\Delta T = \max_i T_i - \min_i T_i$  entre pilotos para la estrategia ganadora de cada uno:

Tabla 4.3: Dispersión del tiempo total esperado entre pilotos al introducir progresivamente las acotaciones. La columna  $\Delta T$  es la diferencia entre el máximo y el mínimo entre los 20 pilotos de Sakhir 2023.

Configuración	$\Delta T$ (s)	Reducción frente a línea base
Línea base (sin acotaciones)	8 343	ref.
+ Corrección térmica acotada	3 630	-56 %
+ Envoltura del Motor 1 y curvas	3 029	-64 %
+ Acotación en <i>rollouts</i> MC	895	-89 %
+ Envoltura del tiempo total	375	-95,5 %

La configuración final sitúa el rango de duraciones de carrera en [1:32:18, 1:38:33], con mediana 1:35:28 y ritmo medio 100,5 s/vuelta, valores coherentes con la duración real del Gran Premio de Baréin de 2023 (1:33:56 para el ganador, datos oficiales FIA 2023).

## 4.5. Métricas de evaluación

La evaluación del sistema se articula en tres niveles: los modelos de pace de manera aislada, el motor de estrategia en el contexto de carreras históricas, y el sistema completo como herramienta desplegada.

### 4.5.1. Evaluación de los modelos de predicción de ritmo

El modelo Transformer y los perfiles paramétricos se evalúan sobre un conjunto de validación temporal (vueltas de las últimas sesiones disponibles para cada piloto-circuito) mediante:

- **Error Absoluto Medio (MAE):**  $\frac{1}{N} \sum |\hat{t}_i - t_i|$ , interpretable directamente en segundos de tiempo de vuelta.
- **Raíz del Error Cuadrático Medio (RMSE):**  $\sqrt{\frac{1}{N} \sum (\hat{t}_i - t_i)^2}$ , más sensible a errores grandes (vueltas de seguridad no detectadas).

Ambas métricas se calculan en el espacio original de segundos (tras denormalización).

### 4.5.2. Evaluación del motor de estrategia

El motor se valida comparando sus recomendaciones con las estrategias empleadas en carreras históricas. Las métricas de interés son:

- **Concordancia estratégica:** fracción de carreras en que la estrategia real coincide con la mejor recomendada por el motor.
- **Ranking de la estrategia real:** posición media de la estrategia efectivamente usada dentro del ranking de las  $N_{\text{strat}} = 5$  generadas. Una posición cercana a 1 indica buen ordenamiento.
- **Análisis de sensibilidad:** variación de los rankings ante perturbaciones en  $\lambda$ , la pérdida de pit y  $P_{\text{SC}}$ .

### 4.5.3. Evaluación del sistema: latencia

Se mide la latencia del endpoint principal de estrategia bajo tres regímenes:

- **Latencia fría (*cold*):** primera petición tras el arranque del servidor, incluyendo la carga de modelos y la construcción de curvas de ritmo desde cero.
- **Latencia tibia (*warm*):** petición con modelos ya cargados en memoria pero curvas de ritmo no cacheadas.
- **Latencia caliente (*hot*):** petición con modelos y curvas de ritmo en caché, tanto en memoria como en disco.

El objetivo de diseño es una latencia de respuesta inferior a 2 s en el régimen caliente para peticiones de estrategia de un piloto.



# Capítulo 5

## Resultados y Evaluación

Este capítulo responde dos preguntas sobre *RaceScope* en la temporada 2025: 1) cómo de cerca está el tiempo total predicho del tiempo real, piloto a piloto, a lo largo del calendario? 2) Si *RaceScope* hubiera elegido las estrategias de Verstappen y Norris en cada carrera, ¿quién habría ganado el duelo directo? El apartado 5.1 presenta la configuración, 5.2 evalúa la magnitud del error, 5.3 simula el duelo VER–NOR y 5.4 describe la interfaz. Los resultados y limitaciones se discuten en el capítulo 6.

### 5.1. Configuración experimental

La herramienta de computación es un portátil MacBook Pro con chip Apple M4 Pro (arquitectura arm64, 12 núcleos de CPU), 24 GB de memoria unificada y macOS 26.5. La integración Metal Performance Shaders [23] del chip está disponible pero el código no la activa: tanto el entrenamiento como la inferencia del Transformer se ejecutan en CPU, para mantener la trazabilidad numérica del experimento y la portabilidad del entorno entre máquinas sin GPU dedicada.

El entorno de software fija Python 3.11.11, PyTorch 2.4 para el modelo de ritmo, FastAPI 0.115 para el servicio HTTP, React 18 y Vite 5 (sobre Node.js 24) para el cliente, y pandas 2.2 con numpy 1.26 para el procesamiento de datos tabulares. Todos los procesos aleatorios del sistema (*rollout* Monte Carlo, muestreos de la distribución de práctica) se inicializan con una misma semilla fija para garantizar la reproducibilidad.

El conjunto de entrenamiento usa las temporadas 2023 y 2024, descargadas de OpenF1 y procesadas con la *pipeline* de ingesta y preprocesamiento descrita en el

capítulo 4. La temporada 2025 se reserva íntegramente como conjunto *held-out*: ningún piloto, sesión o vuelta de 2025 entra en el entrenamiento. Para la evaluación se utilizan los modelos por defecto del sistema: **Transformer v3** entrenado por piloto con cabeza global de respaldo, y perfiles paramétricos lineales por piloto-circuito-compuesto.

## 5.2. Evaluación sobre la temporada 2025

La evaluación empieza por la pregunta de magnitud: ¿cuánto se acerca el tiempo total de carrera predicho por el motor al tiempo real, piloto a piloto, sobre la temporada 2025 completa? El conjunto evaluado son las 24 carreras del calendario para Norris (NOR), Piastri (PIA), Verstappen (VER) y Leclerc (LEC), con 94 emparejamientos válidos tras descartar las filas claramente afectadas por retirada (tiempo real inferior al 60 % de la mediana del circuito). La pregunta complementaria de clasificación piloto contra piloto, restringida al duelo Verstappen-Norris, se aborda en el apartado 5.3 y reutiliza los tiempos esperados obtenidos en esta evaluación como entrada.

### 5.2.1. Métricas y procedimiento

Para cada par (piloto, carrera) se compara el tiempo total de la estrategia mejor clasificada por el motor con el tiempo real agregado a partir de las vueltas de carrera limpias del registro almacenado. El filtro de vueltas excluye las entradas y salidas de *pit* y las vueltas tras un evento de *safety car*, que distorsionarían la comparación. De esta pareja de tiempos se derivan dos métricas estándar de evaluación:

- **MAPE** (*Mean Absolute Percentage Error*): media del error absoluto relativo al tiempo real, expresada en porcentaje. MAPE es una métrica estándar de evaluación de pronósticos [24]. Cuantifica la precisión sin sesgo de signo.
- **Sesgo medio**: media del error firmado predicho – real en segundos. Su signo indica si el motor sobreestima (positivo) o subestima (negativo) el tiempo de carrera.

### 5.2.2. Resultados agregados

La tabla 5.1 resume las dos métricas principales por piloto, junto con el MAE y RMSE del modelo de ritmo a nivel de vuelta y la tasa de aciertos del Top-5 (proporción de carreras en las que la estrategia real aparece entre las cinco estrategias mejor clasificadas por el motor). El MAPE se mantiene entre el 13,5 % y el 14,6 % en los cuatro casos, con Leclerc como mejor calibrado y Piastri como el más alejado, separados por poco más de un punto porcentual.

El sesgo medio es positivo y de magnitud similar en los cuatro pilotos, entre +637s y +692s: el motor sobreestima el tiempo total de carrera de manera sistemática, en una franja que para un Gran Premio de unos 5400s representa entre el 11,8 % y el 12,8 % del tiempo real. La coincidencia en signo y orden de magnitud apunta a un sesgo del propio motor, no a ruido del *rollout* Monte Carlo: si fuese ruido, los cuatro pilotos no compartirían la misma dirección y la misma magnitud.

Tabla 5.1: Métricas de calibración por piloto, MAPE y sesgo evaluados a nivel de tiempo total de carrera; MAE y RMSE evaluados sobre el tiempo de vuelta predicho por el modelo de ritmo; Top-5 indica la proporción de carreras en las que la estrategia real aparece entre las cinco estrategias mejor clasificadas por el motor.

Piloto	Equipo	N	MAPE (%)	Sesgo (s)	MAE / RMSE (s)	Top-5 (%)
Leclerc (LEC)	Ferrari	23	13,53	+637	9,73 / 11,67	21,7
Norris (NOR)	McLaren	24	14,45	+682	11,89 / 13,72	33,3
Piastri (PIA)	McLaren	23	14,57	+688	12,71 / 14,66	17,4
Verstappen (VER)	Red Bull	23	14,56	+692	11,90 / 14,32	30,4

La figura 5.2 desglosa el error firmado carrera a carrera, con un panel por piloto y los Grandes Premios ordenados según el calendario. La franja gris marca el margen  $\pm 60$ s con el que el motor protege sus predicciones frente a un evento de *safety car*; un error por encima de la banda indica que el sobrecoste predicho excede el margen previsto.

La forma de las cuatro series es casi idéntica entre pilotos: el sesgo positivo se mantiene a lo largo del calendario y los picos coinciden con los mismos circuitos. Melbourne y Spa concentran los errores más grandes, con valores que superan los 1500s; son carreras con incidencias mayores en pista (bandera roja en Australia, lluvia intermitente en Spa) que reducen el tiempo real muy por debajo de la duración nominal y que el motor, diseñado bajo la hipótesis de carrera completa, no puede anticipar clasificaciones oficiales FIA, temporada 2025. Fuera de esos casos extremos, el error firmado se estabiliza por encima del cero pero debajo del techo del margen SC en la mayoría de las carreras.

## CAPÍTULO 5. RESULTADOS Y EVALUACIÓN

Resumen de calibración por piloto (2025)

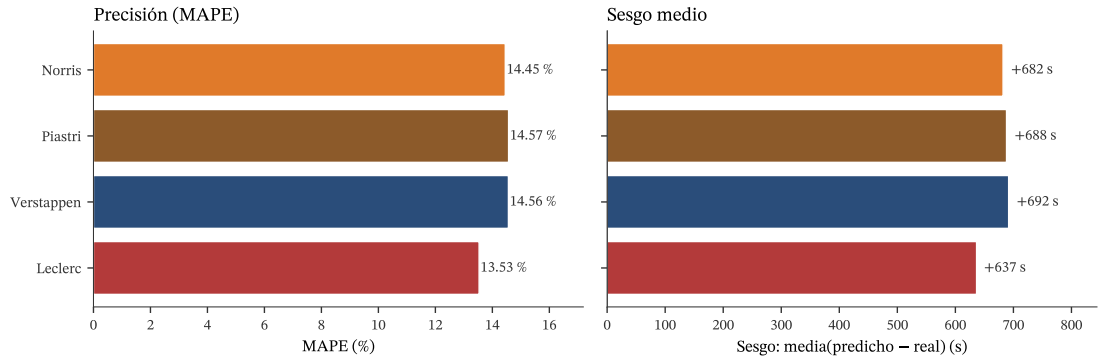


Figura 5.1: MAPE y sesgo medio del motor de predicción por piloto sobre las carreras evaluadas de 2025. Izquierda: precisión global. Derecha: sesgo calculado como  $\text{media}(\text{predicho} - \text{real})$  en segundos.

Error firmado (predicho – real) por carrera — temporada 2025

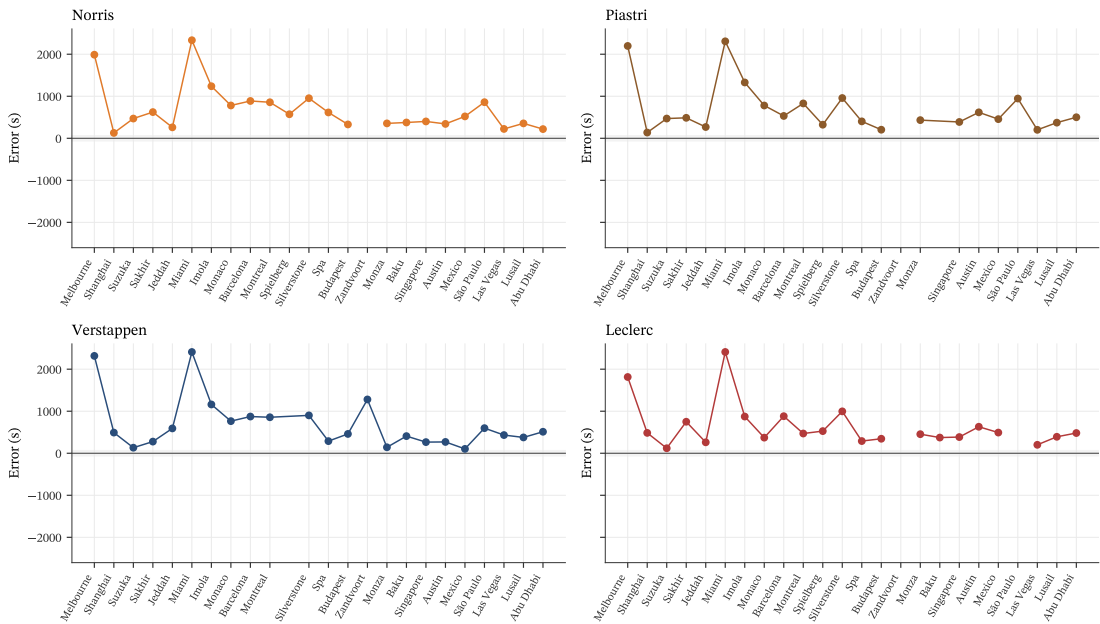


Figura 5.2: Error predicho – real carrera a carrera para Norris, Piastri, Verstappen y Leclerc. Comparten el eje vertical. La franja gris marca el margen  $\pm 60$  s aplicado por el motor frente a un evento de *safety car*.

La concordancia de los cuatro pilotos en magnitud y signo confirma que el sesgo es atribuible al motor y no al piloto evaluado. Tres componentes plausibles lo explican: el margen de 60s aplicado como protección frente a un *safety car*, la elección del tiempo esperado como cuantil alto de la distribución Monte Carlo, y el filtro de vueltas limpias en el lado real, que descarta las entradas y salidas de *pit* y las vueltas neutralizadas y rebaja por construcción la referencia. Diagnosticar la contribución relativa de cada componente queda fuera del alcance de este capítulo y se retoma en el apartado 6.1 del capítulo 6.

Las gráficas individuales por piloto, en formato *dumbbell* de predicho frente a real y serie temporal del error firmado, se recogen en el anexo B. Permiten contrastar el sesgo global con la evolución de un piloto concreto a lo largo del calendario y localizar los Grandes Premios en los que el motor se aleja del tiempo real.

## 5.3. Simulación del campeonato 2025: Verstappen vs Norris

Tras la lectura de magnitud del apartado anterior, esta sección reduce el alcance a una pregunta binaria: a partir de los tiempos esperados predichos por *RaceScope* para los dos contendientes naturales del campeonato 2025, ¿el motor declara al mismo campeón que la temporada real? El sesgo común detectado en el apartado 5.2 se cancela parcialmente al comparar dos pilotos: como afecta a ambos en magnitud y signo similares, lo que importa para la clasificación H2H no es cuánto se aleja cada predicción del tiempo real, sino cuánto se separa una predicción de la otra.

### 5.3.1. Metodología

Para cada Gran Premio de 2025 con datos disponibles se invoca el generador de estrategias sobre dos pilotos, Max Verstappen (VER) y Lando Norris (NOR), y se registra el tiempo esperado de la mejor estrategia para cada uno. Sobre estos tiempos se aplica la siguiente regla:

- En cada carrera, el piloto con menor tiempo total recibe **25 puntos** (la asignación de F1 al ganador) y el otro recibe **18 puntos** (la asignación a la segunda posición), siguiendo el reglamento deportivo FIA, sistema de puntos por Gran Premio [25, Art. A2.2.1].
- Se construyen dos tablas paralelas: la simulada, donde el tiempo es el predicho por *RaceScope*; y la real, donde el tiempo es el agregado de las vueltas de

carrera observadas en la sesión oficial.

- Los puntos se acumulan a lo largo del calendario y la diferencia final declara al campeón en cada uno de los dos universos.

Esta regla mantiene la comparación justa: el sistema de puntos es idéntico en ambos lados, y la única variable que cambia entre la tabla simulada y la real es el origen del tiempo. Carreras con *DNF* (tiempo real inferior a 2000s, indicador de retirada temprana) se descartan del cómputo.

### 5.3.2. Resultados predichos y oficiales

La tabla 5.2 resume la cuenta final del campeonato bajo las dos perspectivas. La columna *Concordancia H2H* reporta el porcentaje de carreras en las que el ganador predicho coincide con el real.

Tabla 5.2: Cuenta final del campeonato 2025 simulado (*RaceScope*) frente al resultado real, restringido al duelo Verstappen-Norris. Cada carrera asigna 25 puntos al ganador y 18 al segundo.

Escenario	Campeón	VER	NOR	$\Delta$	VER-NOR
Simulación <i>RaceScope</i>	VER	508	438	70	16-6
Resultado real (H2H)	Empate (NOR)	473	473	0	11-11

El motor declara a Verstappen vencedor con 70 puntos de margen y un balance de 16 victorias frente a 6 sobre Norris. El balance real entre los dos pilotos es un empate exacto: 11 victorias cada uno y 473 puntos, escenario en el que el sistema de puntos H2H aquí adoptado no permite separar a un campeón sin un criterio adicional. Sobre las 22 carreras evaluadas, el ganador predicho por *RaceScope* coincide con el ganador real del duelo en 11 de ellas, una concordancia del 50% indistinguible del lanzamiento de una moneda.

Conviene situar la comparación en su contexto: no captura, ni pretende capturar, todo lo que decide una temporada real. La predicción de *RaceScope* es *a priori* y aislada por piloto; ignora los accidentes, las banderas rojas, la lluvia y la presencia de los otros 18 monoplazas, que en cada carrera fijan la posición real, los *undercuts* forzados como táctica de adelantamiento mediante parada anticipada [17] y la pérdida de tiempo por tráfico.

El motor sí modela explícitamente la incertidumbre de seguridad en pista: el *rollout* Monte Carlo del Motor 2 sorteja un evento de *safety car* mediante una Bernoulli

con probabilidad  $P_{SC}$  derivada del contexto del Gran Premio y, cuando la parada planificada cae a menos de dos vueltas del evento, recorta hasta 8 s la pérdida efectiva de pit (4.4.3). Es un modelado parcial pero no nulo, y la concordancia del 50 % debe leerse con esta delimitación en mente: el sistema acierta sobre las decisiones estratégicas que dependen del ritmo y de la degradación, no sobre la resolución final del Gran Premio bajo el conjunto completo de incidencias.

Las figuras 5.3b y 5.3a muestran la evolución acumulada de los puntos de VER y NOR carrera por carrera. El eje horizontal sigue el orden cronológico real del calendario 2025, desde Melbourne (carrera 1) hasta Abu Dhabi (carrera 22), recuperado a partir de la fecha de inicio de cada Gran Premio en los metadatos de sesión.

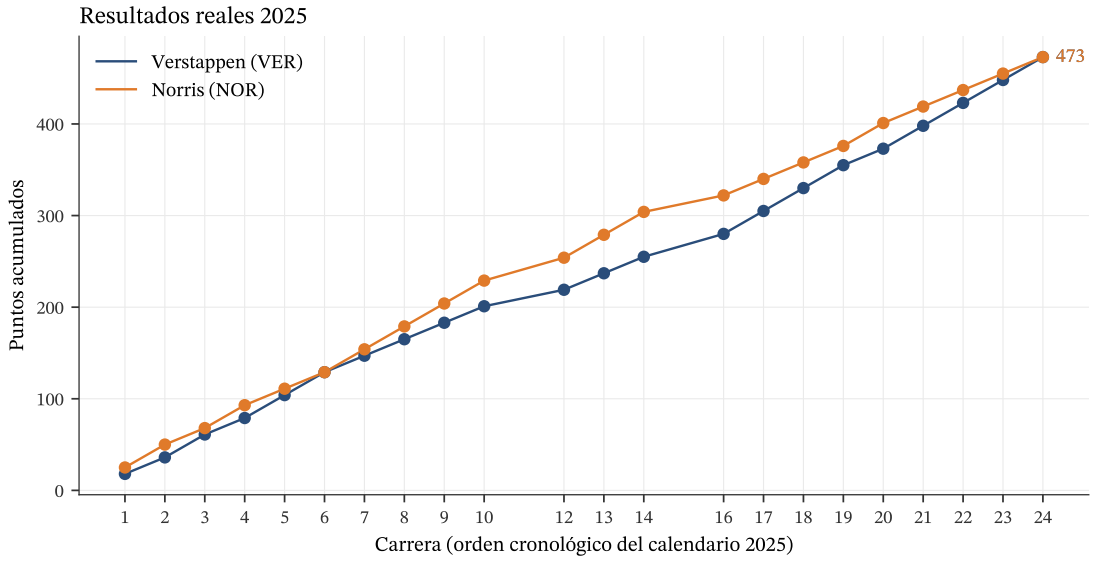
La diferencia entre ambas trayectorias señala los Grandes Premios en los que el motor predice un ganador distinto del real. La discusión en el apartado 6.1 interpreta el solapamiento global y los principales puntos de divergencia.

## 5.4. Aplicación web

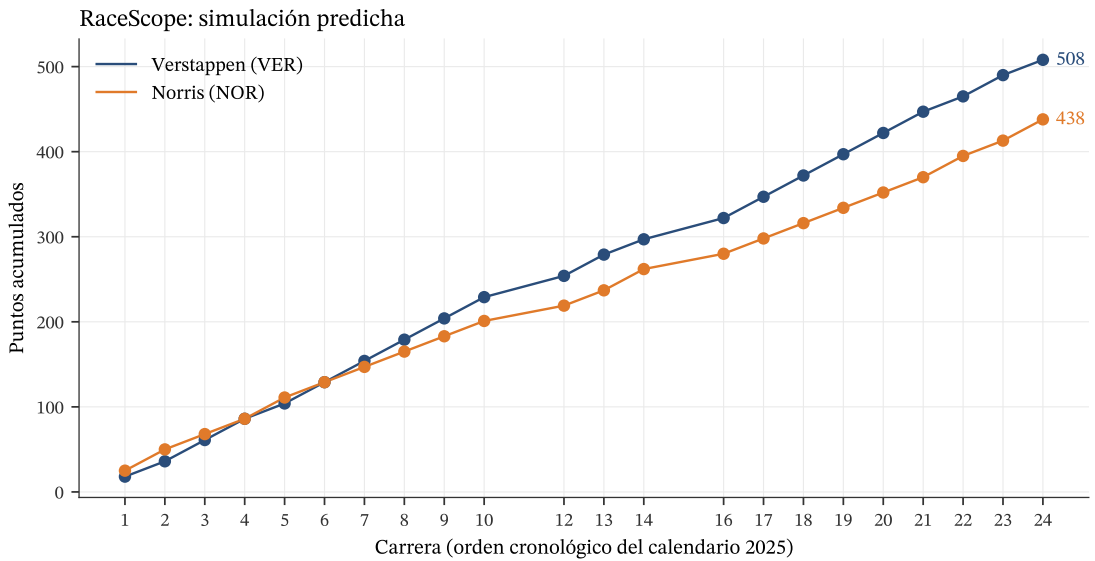
La aplicación web es la cara visible de *RaceScope*, la capa donde las decisiones del motor se convierten en una experiencia consultable por una persona. Su diseño parte de una premisa: el usuario no quiere ver el modelo, quiere ver una estrategia y comparar dos pilotos sobre la misma carrera. Todo lo demás se ha procurado mantener fuera de su camino.

### 5.4.1. Estructura de navegación

La cabecera presenta cuatro pestañas ordenadas según la temporalidad de la información que ofrecen sobre un Gran Premio concreto: **Home** como punto de entrada, **Pre-race** para la decisión antes del semáforo verde, **Live** para el seguimiento durante la carrera y **Rewatch** para la lectura post-mortem. Solo las dos primeras están implementadas en el prototipo actual (tabla 5.3); las otras dos se reservan como continuación natural del trabajo. Exponerlas como espacios pendientes, en lugar de esconderlas bajo descripciones futuras, comunica el alcance previsto del producto sin disfrazarlo.



(a) Resultados reales 2025: misma regla 25/18 aplicada al tiempo real agregado a partir de los tiempos de carrera observados.



(b) Simulación predicha por *RaceScope*: la regla 25/18 se aplica al tiempo esperado de la mejor estrategia generada para cada piloto.

Figura 5.3: Evolución acumulada de los puntos de Verstappen y Norris bajo el sistema de puntos 25/18 por carrera. Eje horizontal: índice de carrera en orden cronológico del calendario 2025.

Tabla 5.3: Pestañas de la aplicación web y su propósito.

Pestaña	Estado	Propósito
Home	Implementada	Punto de entrada y presentación del sistema.
Pre-race	Implementada	Análisis pre-carrera para uno o dos pilotos.
Live	Pendiente	Análisis en directo durante la carrera.
Rewatch	Pendiente	Reproducción analítica post-carrera.

### 5.4.2. Diseño visual

La interfaz adopta el modo oscuro por defecto. Es la convención dominante en herramientas de análisis y de monitorización deportiva: reduce la fatiga ocular en sesiones largas de mirada fija, deja respirar a los datos saturados de color (compuestos rojos, amarillos, grises claros) y evoca el muro de boxes real, donde los paneles se mantienen en baja luminancia para no interferir con la visión durante carreras nocturnas o crepusculares. Un único color de acento, un naranja saturado tomado de la identidad de *RaceScope*, aparece reservado para la acción primaria (el botón **Calcular**) y los marcadores que el usuario debe encontrar de un vistazo.

Los compuestos de neumático adoptan los colores que el aficionado a la Fórmula 1 ya reconoce: rojo para el blando, amarillo para el medio y gris claro para el duro. Esta convención compartida con la propia FIA reduce a cero la carga cognitiva del aprendizaje de la leyenda. La misma paleta se replica entre la tira de estrategias y el gráfico de curvas de ritmo, de modo que el lazo visual entre la estrategia y su curva es inmediato. El principio que opera aquí es el de coherencia: un mismo concepto debe presentarse con el mismo signo visual en cualquier lugar de la interfaz, para que el ojo no tenga que traducir.

El usuario puede conmutar al tema claro con un control en la cabecera, y la aplicación recuerda la preferencia entre sesiones. La internacionalización es funcional, con español como idioma por defecto y un conmutador a inglés en la misma zona; la elección queda preservada también entre visitas. Los dos controles ocupan posiciones idénticas y vecinas, comunicando que pertenecen al mismo registro: ajustes de presentación, no de contenido.

### 5.4.3. Recorrido del usuario

La pestaña Pre-race concentra el caso de uso principal y se estructura como una secuencia lineal de cuatro momentos.



Figura 5.4: *Landing page* de *RaceScope*, con titular, llamada a la acción y vista previa animada de curvas de ritmo.

**Primero, el contexto.** Año, circuito y selección de pilotos se sitúan en una zona aireada en la parte superior, donde la mirada cae de forma natural siguiendo el patrón F de lectura occidental [26]. Un único botón, **Calcular**, actúa como disparador. Siendo éste el único control destacado de la pantalla en estado inicial, evitando la ambigüedad de un panel con varios botones de aspecto comparable.

**Segundo, la espera.** Mientras el motor procesa la petición, las filas de piloto adoptan una micro-animación de carga sin reorganizar la estructura final. El usuario sabe a dónde llegará el resultado antes de que llegue. Esta continuidad espacial es lo que distingue una espera tolerable de una espera ansiosa: la incertidumbre se reduce a la dimensión del tiempo, no a la del lugar.

**Tercero, el resultado primario.** Cada piloto presenta, como máximo, sus cinco estrategias como un carrusel horizontal de píldoras ordenadas por tiempo esperado. Cada píldora se condensa a tres unidades visuales: el tipo de plan, los compuestos como puntos de color en orden cronológico y el delta de tiempo respecto a la mejor opción. Tres datos suficientes para una decisión informada sin obligar al usuario a leer texto.

**Cuarto, el detalle profundo.** Al pulsar una píldora, un gráfico de curva de ritmo despliega vuelta a vuelta lo que la estrategia significa en pista. Los *stints* aparecen como líneas continuas coloreadas por compuesto, y las ventanas de *pit* se proyectan como bandas verticales semitransparentes. Un indicador dinámico aparece bajo el cursor con la lectura precisa de la vuelta enfocada.

La doble fila de filas de piloto, dispuesta lado a lado, materializa el caso de uso *head-to-head*: dos pilotos en idéntico contexto, ocupando porciones simétricas del *viewport*, con cada análisis sirviendo de referencia visual para el otro. El paralelismo espacial refuerza el paralelismo conceptual y permite la comparación directa sin que la mirada tenga que recordar lo que vio un segundo antes.

## CAPÍTULO 5. RESULTADOS Y EVALUACIÓN

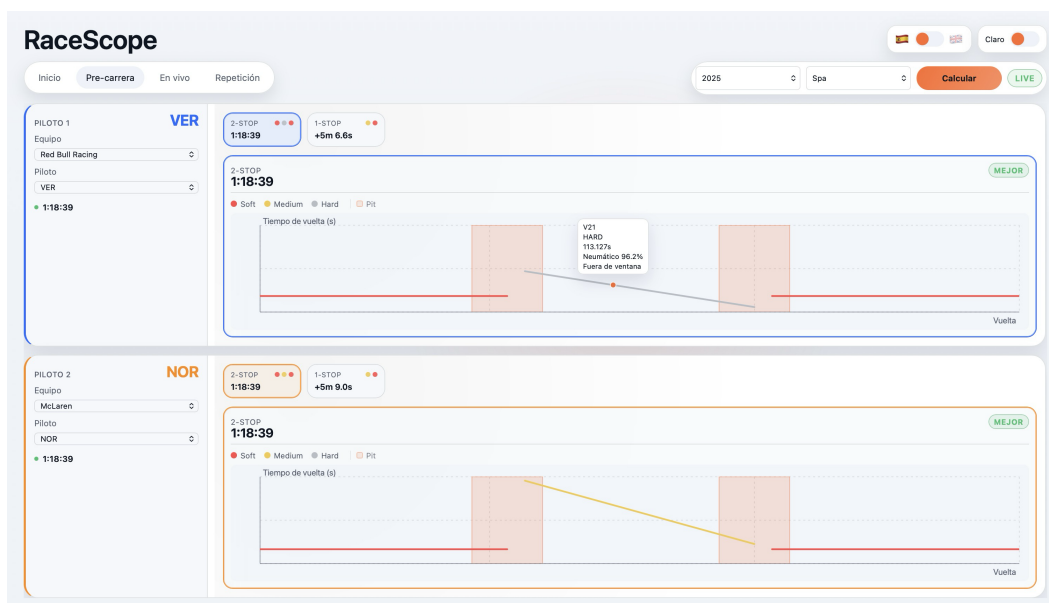


Figura 5.5: Vista comparativa Pre-race con las dos filas de piloto lado a lado tras pulsar Calcular (caso Verstappen vs Norris).

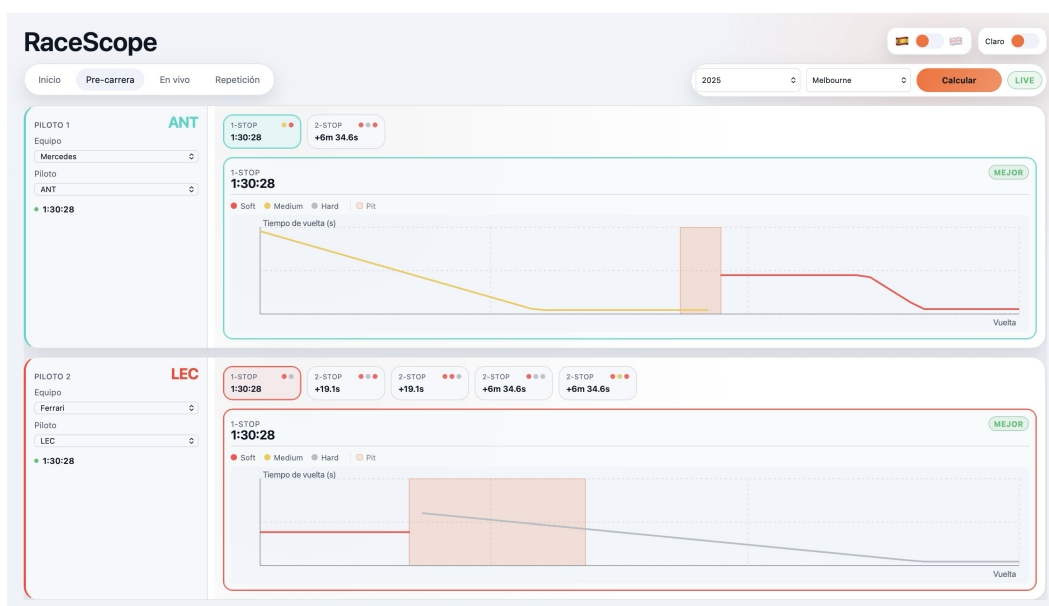


Figura 5.6: Vista comparativa Pre-race con las dos filas de piloto lado a lado tras pulsar Calcular (caso Antonelli vs Leclerc).

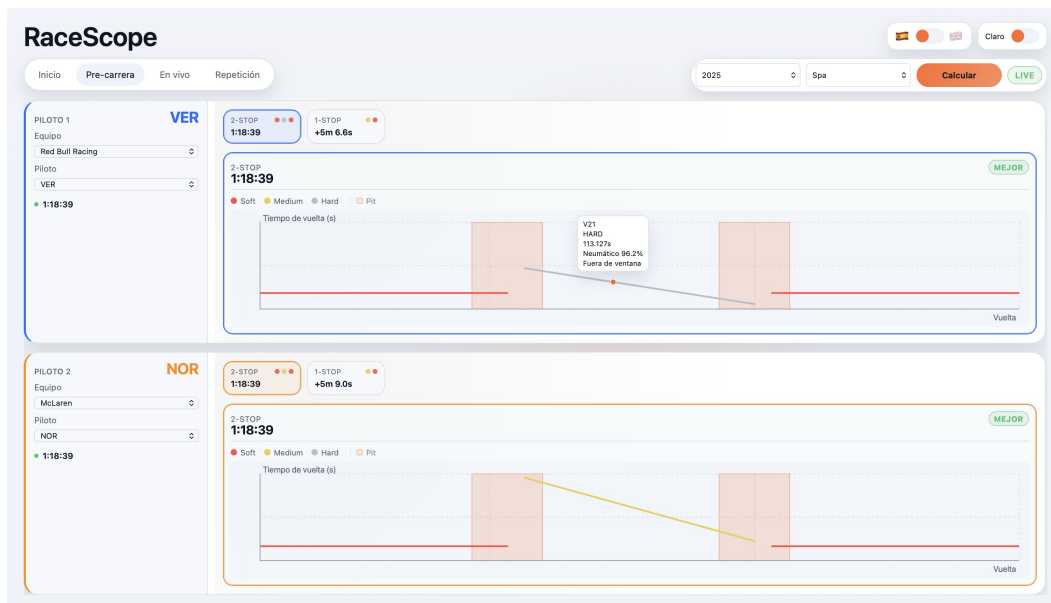


Figura 5.7: Gráfico de curva de ritmo: líneas por *stint* coloreadas por compuesto, bandas de ventana de *pit* y lectura dinámica por vuelta bajo el cursor.



# Capítulo 6

## Discusión, Conclusiones y Trabajo Futuro

Este capítulo cierra el trabajo en tres planos. El apartado 6.1 lee de manera conjunta la evaluación de calibración sobre la temporada 2025, la simulación del campeonato Verstappen-Norris y el comportamiento de la aplicación web, fijando hasta dónde llega el experimento y dónde se queda corto. El apartado 6.2 extrae las conclusiones contrastándolas con los objetivos planteados en el capítulo 1. Los apartados 6.3 y 6.4 enumeran las limitaciones reconocidas y las líneas de continuación a corto y a medio plazo.

### 6.1. Discusión de los resultados

La evaluación de calibración sobre la temporada 2025 mantenida íntegramente como *held-out* ofrece la primera lectura cuantitativa del motor (5.2). El MAPE se mantiene entre el 13,5 % y el 14,6 % en los cuatro pilotos evaluados, una franja estrecha de poco más de un punto porcentual, con Leclerc como mejor calibrado y Piastri como el más alejado.

Lo verdaderamente significativo de la lectura es el sesgo medio: positivo y de magnitud similar entre +636 s y +692 s en los cuatro pilotos, sobre un Gran Premio que dura del orden de 5400 s. La coincidencia en signo y orden de magnitud descarta el ruido aleatorio como explicación; si lo fuera, los cuatro pilotos no compartirían ni dirección ni amplitud. El sesgo es por tanto atribuible al motor y se reparte entre tres mecanismos de protección que actúan en cascada: el margen de 60 s aplicado para protegerse frente a un *safety car*, la elección del tiempo esperado

como cuantil alto de la distribución Monte Carlo, y el filtro de vueltas limpias en el lado real, que rebaja por construcción la referencia. Los picos del error firmado se concentran en Melbourne y Spa por causas externas a la decisión estratégica (bandera roja en Australia, lluvia intermitente en Bélgica) que reducen el tiempo real muy por debajo de la duración nominal y que el motor, planteado bajo la hipótesis de carrera completa, no puede anticipar.

La simulación del campeonato entre Verstappen y Norris cambia la pregunta: de cuánto se aleja cada predicción del tiempo real a quién gana el duelo directo carrera a carrera. El sesgo común detectado en la calibración se cancela en buena medida al comparar dos pilotos afectados por la misma magnitud y signo, de modo que la cifra absoluta importa menos que la diferencia entre las dos predicciones.

Pese a esta ventaja estructural, el resultado sigue siendo poco favorable: *RaceScope* declara a Verstappen vencedor por 70 puntos cuando el duelo real entre ambos termina empatado a 11-11, una distancia de un campeón completo entre la simulación y la realidad. La concordancia del 50 % en el ganador carrera a carrera es la cifra que mejor resume esta brecha y coincide con la probabilidad de un sorteo aleatorio entre dos pilotos de élite.

Las dos limitaciones estructurales ya identificadas en el capítulo 4 completan el cuadro y explican gran parte del comportamiento observado en los dos apartados anteriores: la deriva temporal entre las temporadas de entrenamiento (2023-2024) y el conjunto evaluado (2025), que la acotación de la envoltura por circuito amplifica al impedir predicciones más rápidas que la cota empírica anterior, y la ausencia de información de carrera en tiempo real (rivales, tráfico, decisiones reactivas del muro) que en la realidad gobierna parte de las posiciones finales.

El comportamiento del sistema sobre el caso de uso principal, el análisis Pre-race con comparativa dual, sí cumple su función: la aplicación web entrega resultados estables en cuestión de segundos sobre una interfaz coherente, con tema dual e internacionalización funcional. Las pestañas Live y Rewatch quedan reservadas para futuras iteraciones del prototipo. La cota de validez física descrita en el capítulo 4 mitiga los modos de fallo numérico más graves del modelo paramétrico, aunque su efecto secundario, la imposibilidad de adaptarse a un régimen más rápido que el observado en el conjunto de entrenamiento, queda señalado por esta misma evaluación como la siguiente limitación a resolver.

## 6.2. Conclusiones

Las conclusiones del trabajo se ordenan en torno al objetivo general y a los cinco objetivos específicos planteados en 1.2. Para cada uno se indica el grado de consecución y el resultado clave del documento que lo respalda.

El **objetivo general** consistía en diseñar, implementar y evaluar un sistema *end-to-end* de exploración de estrategias de carrera en Fórmula 1 que combinara aprendizaje profundo para la predicción de tiempos de vuelta con simulación Monte Carlo para la clasificación de estrategias de *pit stop*. El sistema está construido como una pieza única que va desde la ingesta de datos públicos hasta la entrega de estrategias rankeadas al usuario, se ha evaluado sobre la temporada 2025 mantenida íntegramente como conjunto *held-out*, y se materializa en la aplicación web descrita en 5.4.

El alcance del prototipo entregado se restringe al análisis pre-carrera; las vistas en directo y de reproducción posterior, anunciadas en el diseño original, quedan reservadas como continuación natural del trabajo en 6.4.3.

Sobre los **cinco objetivos específicos** la valoración es la siguiente.

1. **Pipeline de ingestión y preprocesamiento de datos de telemetría F1 (alcanzado).** El sistema descarga las temporadas solicitadas de la API *OpenF1* con un cliente que respeta los límites de tasa mediante *token bucket* y reintento con *backoff* exponencial, y produce una *feature store* estructurada por temporada, circuito y piloto sobre la que se construyen los modelos (4.1). Las quince características continuas, las tres categóricas y los dos identificadores estáticos que alimentan el Transformer se enumeran en la Tabla 4.1.
2. **Modelos de predicción de tiempos de vuelta por piloto, con respaldo global (alcanzado).** La cuarta iteración del modelo, **Transformer v3** con  $\approx 14,3 \times 10^6$  parámetros, se entrena de forma específica por piloto y conserva una cabeza global como respaldo para pilotos con datos insuficientes (4.2). La evaluación carrera a carrera sobre 2025 (5.2) reporta un MAPE entre el 13,5 % y el 14,6 % en los cuatro pilotos evaluados, cifra que cuantifica el comportamiento del modelo sobre datos no vistos y permite identificar el sesgo común al alza discutido en 6.1.
3. **Modelo paramétrico de degradación de neumáticos sensible a las condiciones térmicas (alcanzado).** El perfil lineal por piloto-circuito-compuesto descrito en 4.3 captura la evolución del tiempo de vuelta como función de la antigüedad del *stint* y de las desviaciones térmicas de pista y

aire respecto a la referencia del circuito. La envoltura empírica del circuito (4.4.5) acota los coeficientes OLS dentro del rango físicamente admisible y reduce la dispersión entre pilotos del tiempo total esperado de 8343s a 375s sobre el caso de referencia de Sakhir 2023.

4. **Motor de estrategia construido sobre tres componentes que colaboran (alcanzado).** El motor opera tal como se planteó: el Motor 1 resuelve de forma cerrada la vuelta óptima de cada parada a partir del perfil lineal; el Motor 2 refina por Monte Carlo las mejores candidatas con el Transformer y cuantifica su incertidumbre; el Motor 3 ordena el conjunto con un criterio media-varianza inspirado en el marco de Markowitz (4.4). El uso del motor sobre las 22 carreras del duelo Verstappen-Norris en 2025 (5.3) demuestra que la cadena completa opera de manera estable y produce resultados accionables sobre datos no vistos durante el entrenamiento.
5. **Despliegue como aplicación web interactiva (parcialmente alcanzado).** El cliente está desplegado y operativo en su caso de uso principal, la pestaña *Pre-race* con análisis comparativo de dos pilotos sobre el mismo Gran Premio, tema dual claro/oscuro e internacionalización funcional (5.4). Las pestañas *Live* y *Rewatch* están presentes en la barra de navegación pero sin contenido funcional, y la vista de análisis de sensibilidad anunciada en el objetivo no se ha materializado en el prototipo. El cierre de estas tres piezas constituye el bloque principal de trabajo futuro a largo plazo recogido en 6.4.3.

En conjunto, cuatro de los cinco objetivos específicos se cumplen íntegramente y el quinto se cumple parcialmente en su componente más visible: la aplicación entregada cubre el caso de uso central pero deja abiertas las vistas en directo y de reproducción y el análisis de sensibilidad, que se trasladan a la sección 6.4.

Desde un punto de vista más personal y cercano a la motivación inicial del proyecto, *RaceScope* consigue acercar al público general el detalle y los aspectos fundamentales para entender y valorar la estrategia en el contexto de un Gran Premio de Fórmula 1. Mediante la predicción de las estrategias previas a la carrera, el usuario es capaz de comparar y contrastar las posibilidades de su piloto favorito con las de sus rivales, y de anticipar el desarrollo de la carrera a partir de las paradas previstas. La aplicación web, con su interfaz y diseño enfocado en el usuario, hace accesible esta información a un público amplio, más allá de los expertos en análisis de datos o en estrategia deportiva.

Añadido a esto, si bien el la motivación inicial viene de acercar este mundo a aficionados menos técnicos, *RaceScope* aporta mucho valor a los aficionados más avanzados, que pueden disfrutar de esta información de manera interactiva y visual,

y que pueden profundizar en los detalles de la estrategia y su evolución a lo largo de la carrera. La posibilidad de comparar estrategias entre pilotos y de analizar cómo las condiciones de la carrera afectan a estas estrategias añade una capa adicional de comprensión y disfrute para los aficionados más comprometidos. *RaceScope* hace posible que el aficionado sienta que el muro de ingenieros está en su casa.

En definitiva, el sistema no solo cumple con sus objetivos técnicos, sino que también logra su propósito de hacer que la estrategia de carrera sea más comprensible y atractiva.

### 6.3. Limitaciones

Las limitaciones se ordenan de mayor a menor impacto sobre la calidad de la predicción y cada una cierra apuntando a la subsección de 6.4 que la aborda, de modo que ambas listas puedan leerse en paralelo.

1. **Deriva temporal entre temporadas.** Los modelos se entrenan con 2023-2024 y se evalúan sobre 2025, en el que los tiempos de carrera son consistentemente más rápidos. La envoltura del circuito impide predecir por debajo de la cota histórica y produce el sesgo sistemático al alza reportado en el capítulo 5. Se aborda en 6.4.2 con la incorporación incremental de nuevas temporadas y una capa de cuantificación de incertidumbre.
2. **Imposibilidad de incorporar circuitos nuevos sin reentrenamiento.** El sistema está, por construcción, planteado para predecir el ritmo en regímenes ya observados: el circuito es un *embedding* categórico aprendido del histórico, los perfiles paramétricos se ajustan por piloto-circuito-compuesto y la envoltura empírica (4.4.5) se deriva del propio circuito. Un trazado nuevo deja sin anclaje a los tres componentes y exige acumular vueltas y reentrenar. Las arquitecturas que regularizan mejor el régimen de muestra reducida y la cuantificación de incertidumbre recogidas en 6.4.2 atacan el mismo problema desde el lado del modelo.
3. **Ausencia de información de carrera en tiempo real.** El motor opera *a priori*, con la información previa al semáforo verde, y no reacciona a la entrada efectiva del *safety car* (más allá de la implementación del Motor 2), a las banderas amarillas ni al comportamiento real de los rivales. La solución estructural pasa por la pestaña *Live* recogida en 6.4.3.
4. **Sobreajuste residual de los perfiles paramétricos.** Los coeficientes OLS del perfil de degradación pueden caer fuera del rango físicamente admisible en

combinaciones piloto-circuito-compuesto poco representadas. La envoltura de validez (4.4.5) los acota a posteriori, con efecto medible (la dispersión entre pilotos del tiempo total cae de 8343 s a 375 s en Sakhir 2023), pero no resuelve el sobreajuste de raíz. 6.4.2 lo ataca con arquitecturas que regularizan mejor el régimen de muestra reducida.

5. **Dinámica de tráfico modelada como ruido.** La interacción competitiva entre pilotos (*undercut*, *overcut*, lucha por DRS, defensa) se reduce hoy a un componente de ruido gaussiano sobre el *rollout*, y las pérdidas reales por tráfico no se capturan como tales. Su modelado explícito desde teoría de juegos y aprendizaje por refuerzo multiagente queda recogido en 6.4.3.
6. **Pestañas Live y Rewatch no implementadas.** El prototipo cubre solo el caso de uso *Pre-race*. Las dos pestañas restantes aparecen en la barra de navegación como espacios reservados; su cierre constituye la primera línea de trabajo futuro a largo plazo descrita en 6.4.3.
7. **Ingestión bloqueante y caché sin políticas de expulsión.** El cliente HTTP de *OpenF1* es síncrono y la caché en memoria del API no implementa expulsión por tamaño. No resulta problemático en el rango de datos manejado, pero limita la operación a escala de campeonato continuo; el paso a primitivas asíncronas y políticas de expulsión por TTL y tamaño se recoge en 6.4.1.

## 6.4. Trabajo futuro

### 6.4.1. Mejoras a corto plazo

A corto plazo, varias mejoras incrementales cierran las limitaciones operacionales identificadas en 6.3. La estimación de la pérdida de *pit* y de la probabilidad de *safety car* debe derivarse de manera más fina a partir de los tiempos reales de parada y de los mensajes de dirección de carrera, en lugar de los respaldos numéricos actualmente acotados. El cliente HTTP de ingesta debe migrar a primitivas asíncronas para evitar el bloqueo del proceso durante descargas largas. La caché en memoria debe ganar políticas de expulsión por TTL y por tamaño, y el sistema de trazas debe pasar a un formato estructurado que facilite la observabilidad en despliegues más exigentes.

### 6.4.2. Extensiones del modelo

A medio plazo, el modelo de ritmo admite tres direcciones de expansión. Primero, incorporar más temporadas a medida que estén disponibles, con especial atención al solapamiento entre años para amortiguar la deriva temporal observada en la evaluación de 2025. Con esto se deberá tener mucho cuidado, ya que el cambio de normativa técnica de F1 a partir de 2026 [27] hace que se empiece prácticamente de cero en términos de datos históricos, y que la ventana de entrenamiento se reduzca drásticamente. En parte por esto se ha decidido no incluir datos posteriores a 2025, y abarcar únicamente la generación de monoplazas de Fórmula 1 de efecto suelo (*ground effect*) vigente desde 2022 hasta 2025 reglamento técnico FIA [27].

Segundo, explorar variantes más ligeras de la arquitectura Transformer y de modelos especializados en series temporales (N-BEATS [28], PatchTST [29]), así como la integración de modelos fundacionales tipo TimesFM [30] como base para *transfer learning*. Tercero, añadir una capa explícita de cuantificación de incertidumbre, mediante *dropout* bayesiano [31] o un *ensemble* de cabezas, que ofrezca al usuario no solo la estrategia recomendada sino una banda de confianza sobre el tiempo esperado.

### 6.4.3. Evolución del sistema

A más largo plazo, el sistema completo admite tres líneas de desarrollo. La primera es la implementación de las pestañas Live y Rewatch del cliente, que cierra el alcance del producto previsto en el diseño original. Lo ideal sería tener un dashboard, similar a lo que vemos en la pestaña Pre-race, pero con información de carrera en tiempo real (posiciones, tiempos de vuelta, mensajes de dirección de carrera) y un elemento que indique si en ese momento se debe parar, según el contexto de la carrera (lluvia, safety car, tráfico) y el estado del piloto (tiempo de vuelta, ritmo, degradación).

La segunda es el modelado explícito de las interacciones estratégicas entre pilotos. Desde la teoría de juegos [32], cada equipo decide paradas con información parcial sobre rivales (estado de neumático, combustible, tráfico) y la utilidad depende de las decisiones ajenas. Para duelos uno a uno puede usarse un juego bayesiano [33]; para líder y perseguidor, un enfoque tipo Stackelberg [34] resulta más adecuado. A escala de campeonato, el juego se repite y las reputaciones de los equipos pueden integrarse como priors. En la práctica esto se traduce en modelos de tráfico que cuantifican la pérdida por seguir a un rival o, de forma escalable, en aprendizaje por refuerzo multiagente [35] para obtener políticas de equilibrio en simulaciones completas.

La tercera es el despliegue del sistema sobre infraestructura en la nube con orquestación de contenedores, y la incorporación de datos de otras categorías de monoplazas (F2, Fórmula E) como base de *transfer learning* hacia la propia Fórmula 1.

# Bibliografía

- [1] OpenF1 Project. *OpenF1: The Open Source API for Formula 1 Data*. Accessed March 2026. 2023. URL: <https://openf1.org/>.
- [2] Fédération Internationale de l'Automobile. *2026 Formula 1 Sporting Regulations — Section B*. Reglamento Deportivo Issue 06. Aprobado por el WMSC el 28/04/2026. FIA, abr. de 2026. URL: <https://www.fia.com/regulation/category/110>.
- [3] Hans B. Pacejka. *Tyre and Vehicle Dynamics*. 2nd. Butterworth-Heinemann, 2006. ISBN: 978-0-7506-6918-4.
- [4] Flavio Farroni et al. «TRT EVO: Advances in Real-Time Thermodynamic Tire Modeling for Vehicle Dynamics Simulations». En: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 233.1 (2019), págs. 121-135. DOI: 10.1177/0954407018808992.
- [5] Felipe Aguad y Charles Thraves. «Optimizing Pit Stop Strategies in Formula 1 with Dynamic Programming and Game Theory». En: *European Journal of Operational Research* 319.3 (2024), págs. 908-919. DOI: 10.1016/j.ejor.2024.07.011.
- [6] Philipp Schaefer. *FastF1: Python Package for F1 Data Analysis*. GitHub: @theOehrly. Accessed March 2026. 2021. URL: <https://docs.fastf1.dev/>.
- [7] Apache Software Foundation. *Apache Parquet: Columnar Storage Format*. Accessed March 2026. 2013. URL: <https://parquet.apache.org/>.
- [8] Ashish Vaswani et al. «Attention Is All You Need». En: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017, págs. 5998-6008. URL: <https://arxiv.org/abs/1706.03762>.
- [9] Ruibin Xiong et al. «On Layer Normalization in the Transformer Architecture». En: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 119. PMLR. 2020. URL: <https://arxiv.org/abs/2002.04745>.
- [10] Wikipedia contributors. *Huber loss*. Wikipedia, The Free Encyclopedia. Accessed June 2026. 2026. URL: [https://en.wikipedia.org/wiki/Huber\\_loss](https://en.wikipedia.org/wiki/Huber_loss).

- [11] Yufeng Hao et al. *AdamW*. Cornell University Computational Optimization Open Textbook — Optimization Wiki. Accessed June 2026. 2024. URL: <https://optimization.cbe.cornell.edu/index.php?title=AdamW>.
- [12] PyTorch Contributors. *torch.optim.lr\_scheduler.OneCycleLR*. PyTorch Documentation, v2.12. Accessed June 2026. 2026. URL: [https://docs.pytorch.org/docs/2.12/generated/torch.optim.lr\\_scheduler.OneCycleLR.html](https://docs.pytorch.org/docs/2.12/generated/torch.optim.lr_scheduler.OneCycleLR.html).
- [13] Stephen W. Raudenbush y Anthony S. Bryk. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd. Sage Publications, 2002. ISBN: 978-0-7619-1904-9.
- [14] Reuven Y. Rubinstein y Dirk P. Kroese. *Simulation and the Monte Carlo Method*. 3rd. Wiley, 2016. ISBN: 978-1-118-63216-1.
- [15] Harry M. Markowitz. «Portfolio Selection». En: *The Journal of Finance* 7.1 (1952), págs. 77-91. DOI: 10.1111/j.1540-6261.1952.tb01525.x.
- [16] Alexander Heilmeyer et al. «Application of Monte Carlo Methods to Consider Probabilistic Effects in a Race Simulation for Circuit Motorsport». En: *Applied Sciences* 10.12 (2020), pág. 4229. DOI: 10.3390/app10124229.
- [17] Alexander Heilmeyer et al. «Virtual Strategy Engineer: Using Artificial Neural Networks for Making Race Strategy Decisions in Circuit Motorsport». En: *Applied Sciences* 10.21 (2020), pág. 7805. DOI: 10.3390/app10217805.
- [18] James Bekker y Wikus Lotz. «Planning Formula One Race Strategies Using Discrete-Event Simulation». En: *Journal of the Operational Research Society* 60.7 (2009), págs. 952-961. DOI: 10.1057/palgrave.jors.2602626.
- [19] GP Tempo. *GP Tempo: Formula 1 Lap Time Comparison*. Herramienta de análisis en línea. Accessed June 2026. 2026. URL: <https://www.gp-tempo.com>.
- [20] Amazon Web Services. *F1 Insights Powered by AWS*. F1 on AWS. Accessed June 2026. 2026. URL: <https://aws.amazon.com/sports/f1/>.
- [21] Sepp Hochreiter y Jürgen Schmidhuber. «Long Short-Term Memory». En: *Neural Computation* 9.8 (1997), págs. 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.
- [22] Pat Symonds. *Por qué la temperatura de los neumáticos es un gran problema en la F1*. Motorsport.com (publicación original en GP Racing). Publicado 17 diciembre 2024. Accessed June 2026. 2024. URL: <https://es.motorsport.com/f1/news/por-que-temperatura-neumaticos-gran-problema/10683165/>.
- [23] Apple Inc. *Metal Performance Shaders*. Apple Developer Documentation. Accessed June 2026. 2026. URL: <https://developer.apple.com/documentation/metalperformanceshaders>.

- 
- [24] Rob J. Hyndman y Anne B. Koehler. «Another Look at Measures of Forecast Accuracy». En: *International Journal of Forecasting* 22.4 (2006), págs. 679-688. DOI: 10.1016/j.ijforecast.2006.03.001.
- [25] Fédération Internationale de l'Automobile. *2026 Formula 1 General Regulatory Provisions — Section A*. Reglamento General Issue 02. Aprobado por el WMSC el 27/02/2026. FIA, feb. de 2026. URL: <https://www.fia.com/regulation/category/110>.
- [26] Kara Pernice. *F-Shaped Pattern of Reading on the Web: Misunderstood, But Still Relevant (Even on Mobile)*. Nielsen Norman Group. Accessed June 2026. Nov. de 2017. URL: <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>.
- [27] Fédération Internationale de l'Automobile. *2026 Formula 1 Technical Regulations — Section C*. Reglamento Técnico Issue 18. Aprobado por el WMSC el 07/05/2026. FIA, mayo de 2026. URL: <https://www.fia.com/regulation/category/110>.
- [28] Boris N. Oreshkin et al. «N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting». En: *International Conference on Learning Representations (ICLR)*. 2020. URL: <https://arxiv.org/abs/1905.10437>.
- [29] Yuqi Nie et al. «A Time Series is Worth 64 Words: Long-term Forecasting with Transformers». En: *International Conference on Learning Representations (ICLR)*. 2023. URL: <https://arxiv.org/abs/2211.14730>.
- [30] Abhimanyu Das et al. «A Decoder-Only Foundation Model for Time-Series Forecasting». En: *International Conference on Machine Learning (ICML)*. 2024. URL: <https://arxiv.org/abs/2310.10688>.
- [31] Yarín Gal y Zoubin Ghahramani. «A Theoretically Grounded Application of Dropout in Recurrent Neural Networks». En: *Advances in Neural Information Processing Systems (NeurIPS)*. 2016. URL: <https://arxiv.org/abs/1512.05287>.
- [32] Wikipedia contributors. *Teoría de juegos*. Wikipedia, La enciclopedia libre. Accessed June 2026. 2026. URL: [https://es.wikipedia.org/wiki/Teor%C3%ADa\\_de\\_juegos](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_juegos).
- [33] Wikipedia contributors. *Juego bayesiano*. Wikipedia, La enciclopedia libre. Accessed June 2026. 2026. URL: [https://es.wikipedia.org/wiki/Juego\\_bayesiano](https://es.wikipedia.org/wiki/Juego_bayesiano).
- [34] Giona Fieni et al. «Game Theory in Formula 1: From Physical to Strategic Interactions». En: *arXiv preprint arXiv:2503.05421* (2025). URL: <https://arxiv.org/abs/2503.05421>.

## BIBLIOGRAFÍA

---

- [35] Wikipedia contributors. *Aprendizaje por refuerzo multiagente*. Wikipedia, La enciclopedia libre. Accessed June 2026. 2026. URL: [https://es.wikipedia.org/wiki/Aprendizaje\\_por\\_refuerzo\\_multiagente](https://es.wikipedia.org/wiki/Aprendizaje_por_refuerzo_multiagente).

# Apéndice A

## Alineación con los Objetivos de Desarrollo Sostenible

Los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas constituyen el marco global de referencia para el desarrollo sostenible hasta 2030. Este anexo analiza la contribución de *RaceScope* a los tres ODS sobre los que el sistema tiene un efecto identificable (9, 17 y 4) y cierra con una estimación de su huella computacional.

### ODS 9 — Industria, innovación e infraestructura

*RaceScope* se inscribe en el ODS 9 como una pieza de analítica aplicada al automovilismo construida sobre datos públicos y código abierto. La combinación de Transformers entrenados por piloto y simulación Monte Carlo, distribuida íntegramente como software libre, trasladada al exterior del entorno de los equipos de Fórmula 1 una capacidad de apoyo a la decisión estratégica que hasta ahora dependía de soluciones propietarias y de canales de telemetría restringidos.

### ODS 17 — Alianzas para lograr los objetivos

El proyecto se construye íntegramente sobre infraestructura y datos abiertos: la API *OpenF1* como fuente gratuita de datos de Fórmula 1, librerías de uso libre para el modelo de ritmo (PyTorch), el servicio HTTP (FastAPI) y el cliente (React), y el formato *Apache Parquet* como estándar de almacenamiento columnar para la

## APÉNDICE A. ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

---

*feature store*. Esta elección reduce a cero la dependencia de plataformas propietarias y permite que la comunidad académica, los investigadores independientes y los aficionados al automovilismo reproduzcan, modifiquen y extiendan el sistema sin barreras de licencia. La contribución del proyecto al ODS 17 se sitúa en este plano: cada componente puede reutilizarse en proyectos derivados, y los datos y los modelos quedan accesibles a quien quiera continuar el trabajo.

### ODS 4 — Educación de calidad

La reproducibilidad del sistema (pipeline documentado, código abierto y arquitectura modular) lo habilita como recurso educativo en cursos de ciencia de datos, ingeniería y analítica de negocios. La separación por componentes facilita además su uso parcial: un curso de aprendizaje profundo puede tomar prestado el bloque del Transformer; uno de simulación, el motor Monte Carlo; uno de desarrollo de aplicaciones web, la arquitectura cliente-servidor montada sobre FastAPI y React. La complejidad del dominio (degradación no lineal, incertidumbre estocástica, decisión bajo restricciones reglamentarias) aporta un caso de estudio realista para enseñar la cadena completa que conecta datos crudos, modelos y decisión.

### Impacto medioambiental del proyecto

El entrenamiento del Transformer se ejecuta sobre el portátil descrito en 5.1, sin GPU dedicada, con tiempos por piloto del orden de minutos. La huella computacional del sistema se mantiene varios órdenes de magnitud por debajo de la de los modelos fundacionales de gran escala. La inferencia también se ejecuta en CPU y devuelve la estrategia en cuestión de segundos, sin requerir aceleración dedicada en producción.

# Apéndice B

## Evaluación de calibración: gráficas por piloto

Este anexo complementa la sección 5.2 con la lectura individual de cada piloto sobre las carreras evaluadas de 2025. Primero se ofrecen los cuatro comentarios cualitativos por piloto; a continuación, la rejilla común con las series temporales del error firmado (figura B.1); y, finalmente, un *dumbbell* a página completa por piloto, en orden Norris, Piastri, Verstappen y Leclerc, con el tiempo total predicho (círculo blanco) y el tiempo real (círculo relleno) carrera a carrera, unidos por una línea fina del color del piloto y con el error firmado anotado a la derecha. La lectura agregada (MAPE, sesgo, distribución del error absoluto) se reserva al capítulo 5; aquí el foco es el comportamiento del motor en cada Gran Premio concreto.

### Norris

Norris cierra la temporada con un MAPE del 14,45 % y un sesgo medio de +682s sobre 24 carreras. El *dumbbell* a página completa (figura B.2) permite localizar las dos divergencias mayores (Melbourne y Spa), donde el motor predice un tiempo cercano al nominal mientras que la carrera real se acorta por bandera roja o lluvia. La serie temporal del error firmado en el orden del calendario aparece en el primer panel de la figura B.1.

## Piastrri

Piastrri es el piloto con mayor MAPE de los cuatro (14,57 %) y un sesgo medio de +688s sobre 23 carreras válidas. Su patrón temporal (segundo panel de la figura B.1) es prácticamente paralelo al de su compañero de equipo Norris, lo que sugiere que el sesgo procede del motor y no de un perfil paramétrico del piloto. El *dumbbell* correspondiente es la figura B.3.

## Verstappen

Verstappen presenta un MAPE del 14,56 % y un sesgo medio de +692s sobre 23 carreras. El *dumbbell* (figura B.4) confirma que la sobreestimación se reparte de manera homogénea por el calendario, sin un circuito concreto que aporte un peso desproporcionado al sesgo global más allá de los casos extremos comunes a los cuatro pilotos. La serie temporal aparece en el tercer panel de la figura B.1.

## Leclerc

Leclerc es el piloto mejor calibrado de los cuatro, con un MAPE del 13,53 % y un sesgo medio de +637s sobre 24 carreras. La mediana del error firmado se sitúa en +476s, la más baja del grupo, lo que se traduce en una franja de error más estrecha en el cuarto panel de la figura B.1. El *dumbbell* correspondiente es la figura B.5.

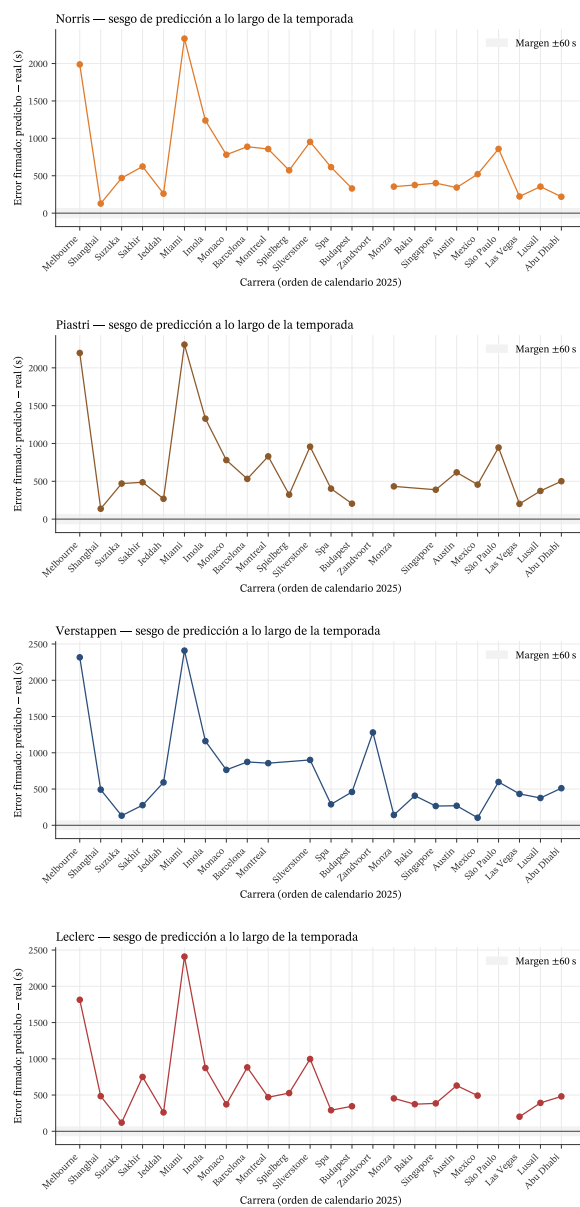


Figura B.1: Series temporales del error firmado predicho – real por piloto sobre las carreras evaluadas de 2025. En cada panel la franja gris marca el margen  $\pm 60$  s aplicado por el motor frente a un evento de *safety car*.

APÉNDICE B. EVALUACIÓN DE CALIBRACIÓN: GRÁFICAS POR PILOTO

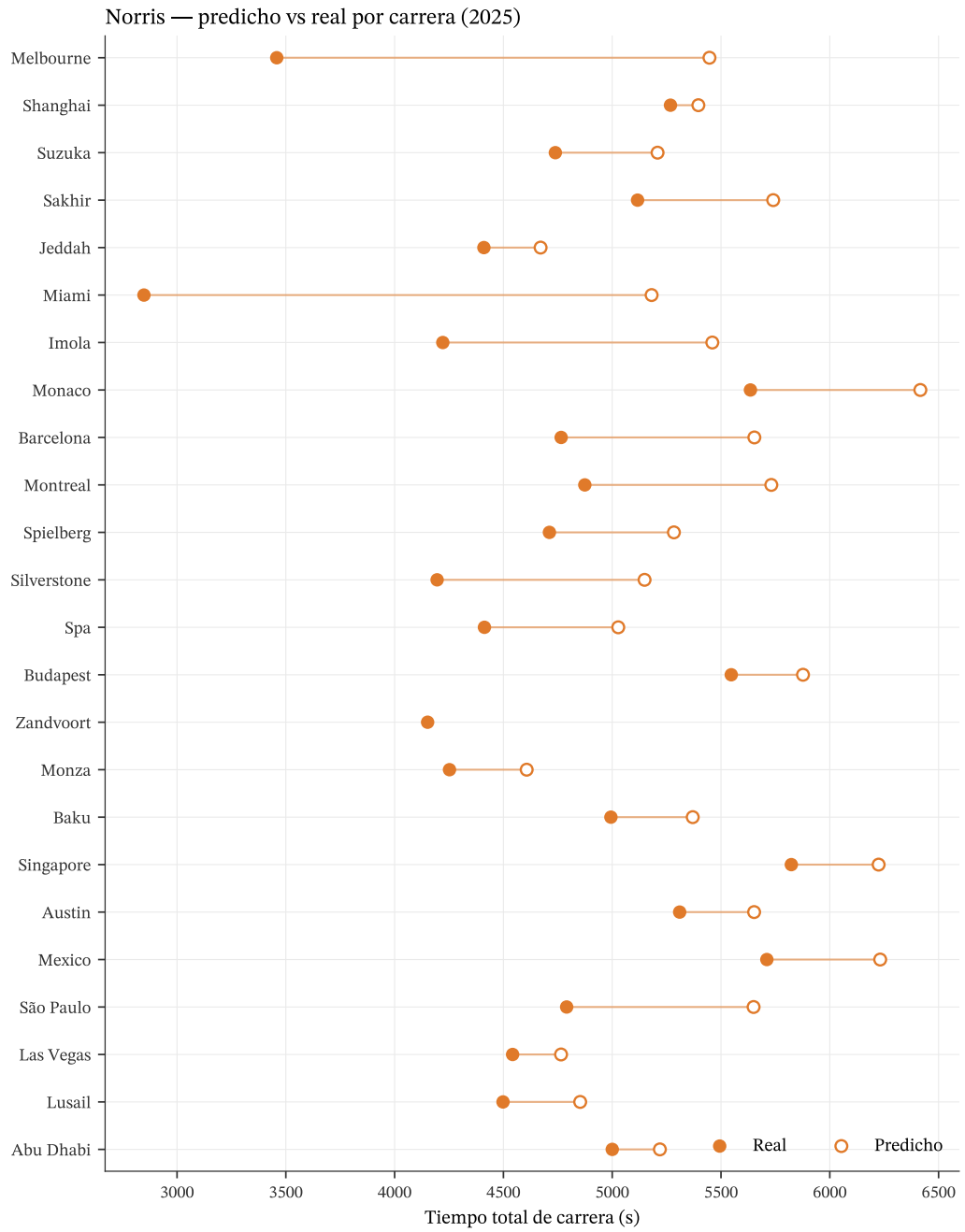


Figura B.2: Norris: tiempo predicho (círculo blanco) frente al tiempo real (círculo relleno) carrera a carrera, con el error firmado anotado a la derecha.

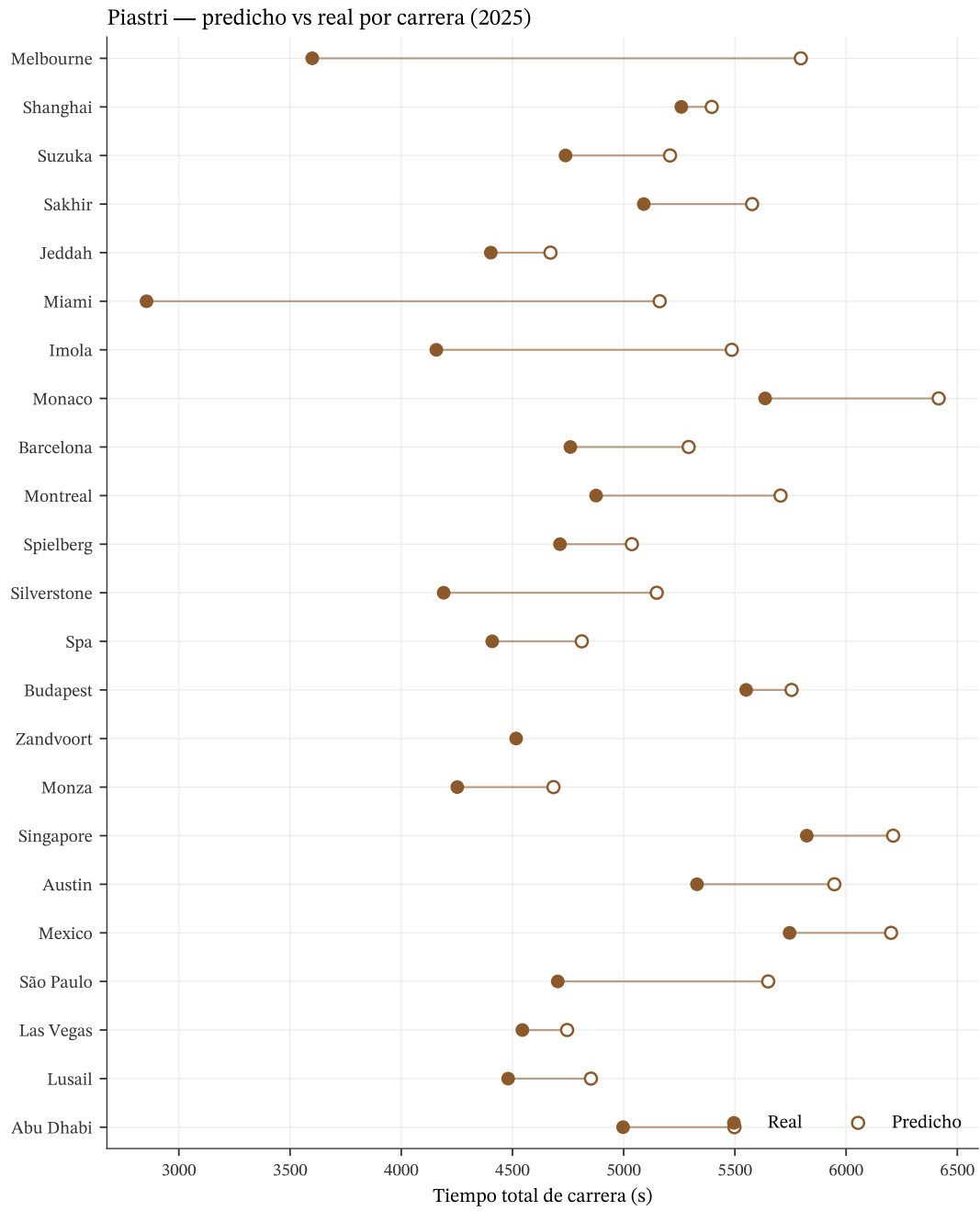


Figura B.3: Piastri: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha.

APÉNDICE B. EVALUACIÓN DE CALIBRACIÓN: GRÁFICAS POR PILOTO

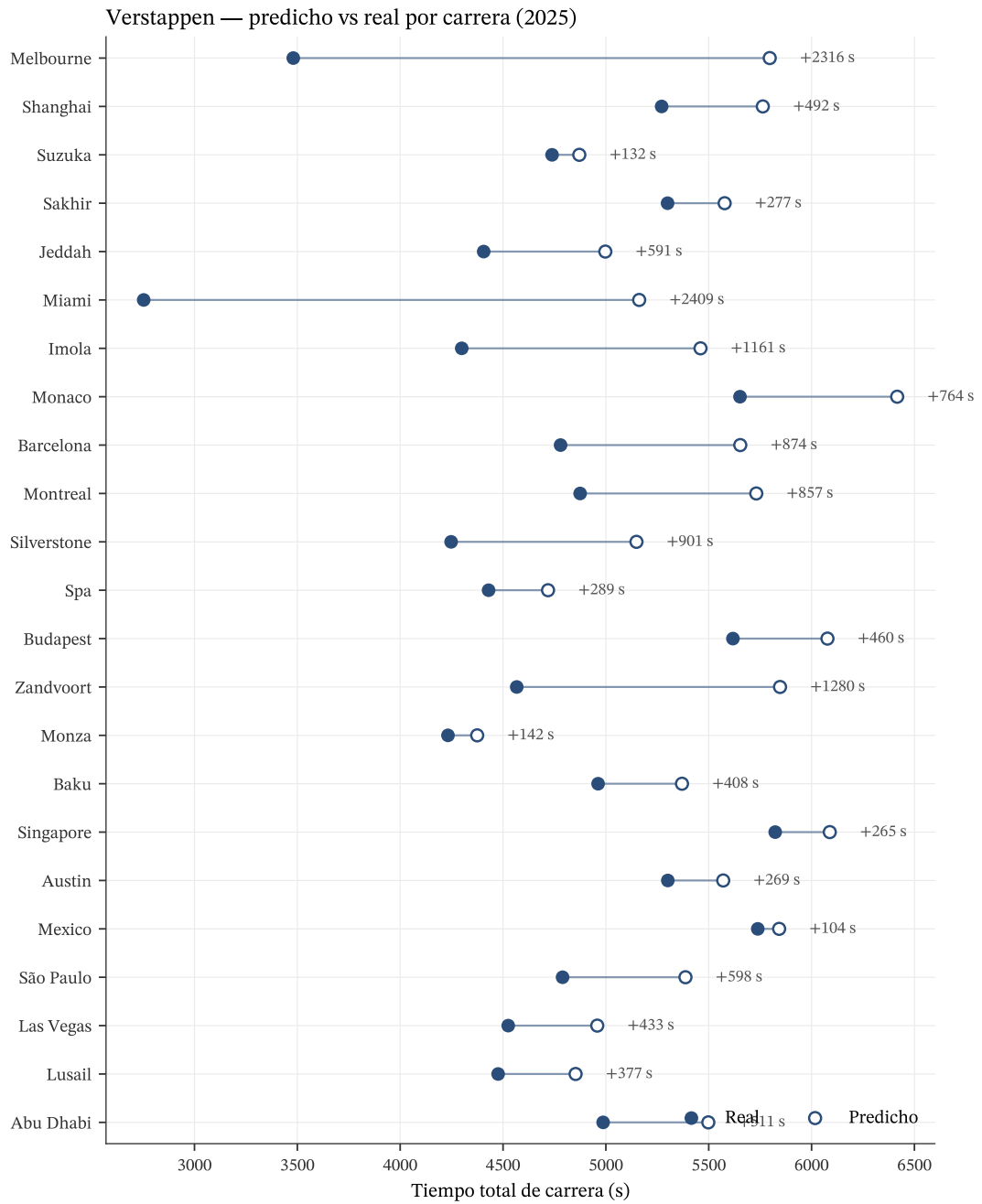


Figura B.4: Verstappen: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha.

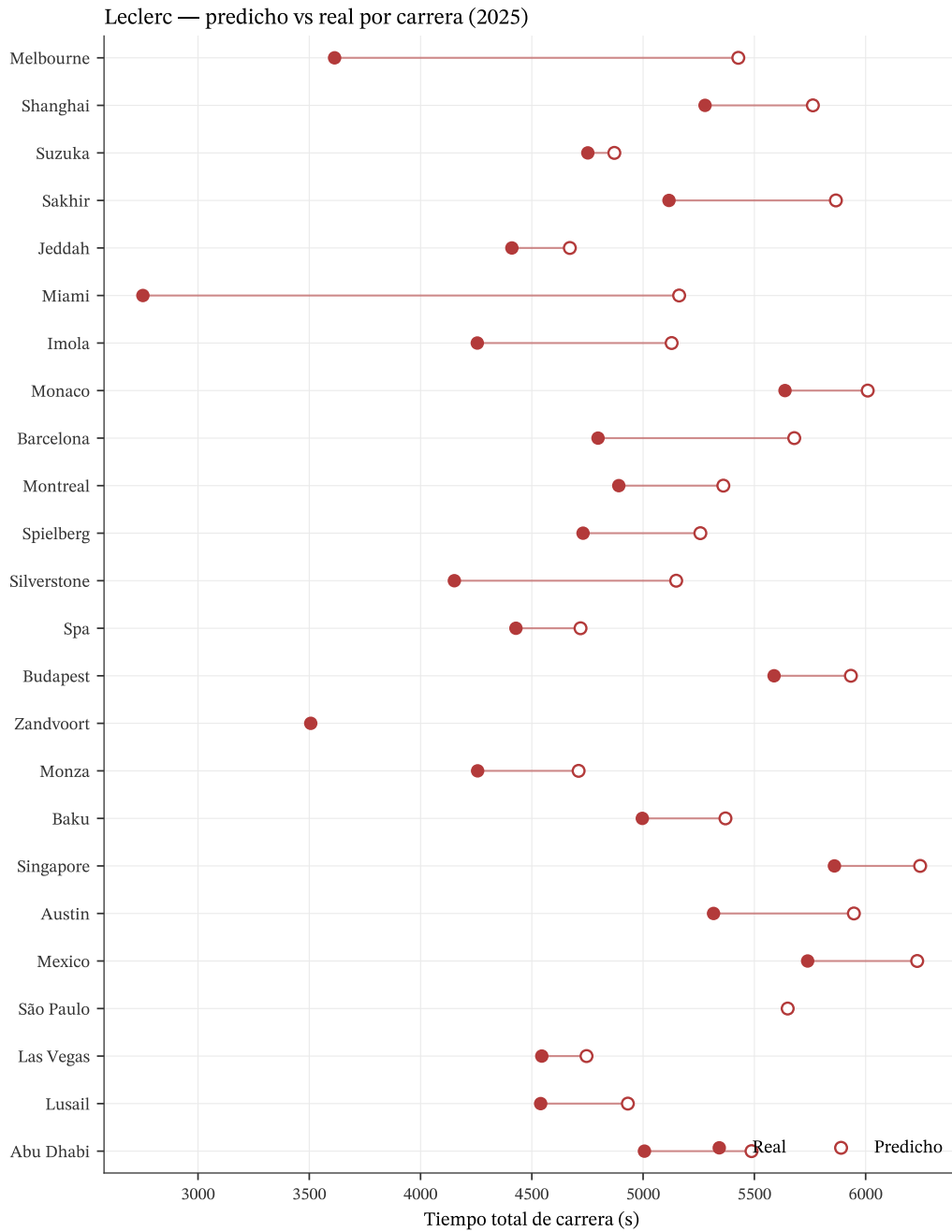


Figura B.5: Leclerc: tiempo predicho frente al tiempo real carrera a carrera, con el error firmado anotado a la derecha.

APÉNDICE B. EVALUACIÓN DE CALIBRACIÓN: GRÁFICAS POR PILOTO

# Apéndice C

## Código fuente del proyecto

El código fuente completo de *RaceScope* se distribuye en abierto en el siguiente repositorio público de GitHub:

`https://github.com/AlvaroGonzalez05/RaceScope`

El repositorio contiene tanto el backend (pipeline de datos OpenF1, modelos Transformer y paramétrico, motor de estrategia y API FastAPI) como el frontend (aplicación web en React + Vite), las fuentes L<sup>A</sup>T<sub>E</sub>X de esta memoria, la declaración de uso de herramientas de Inteligencia Artificial Generativa y la documentación operativa para reproducir el sistema.

Las temporadas de entrenamiento (2023-2024) y el conjunto *held-out* de 2025 se regeneran localmente a partir de la API pública *OpenF1* ejecutando los *scripts* de ingesta y preprocesamiento descritos en el documento pertinente; los artefactos resultantes (parquets, modelos entrenados, cachés) no se versionan para evitar sobrecargar el repositorio.

