

*Universidad Politécnica de Madrid*

*Departamento de Lenguajes y Sistemas Informáticos e  
Ingeniería de Software (DLSIS)*

*Escuela técnica superior de ingenieros informáticos*

# **Modelización**

## **Optimización de Estrategias de Carrera usando Programación por Restricciones**



CAMPUS  
DE EXCELENCIA  
INTERNACIONAL

# Práctica Modelización

## **Autores Grupo 1:**

*Raúl Muñoz Dávila 20M063*

*Alfonso Mateos Vicente 20M041*

*Aaron Moyano Alcacer 20M020*

## **Resumen:**

*Este proyecto presenta el desarrollo de un simulador de carreras de Fórmula 1 utilizando programación por restricciones. El sistema, implementado en Minizinc y Python, utiliza dos componentes fundamentales llamados "cerebros": uno dedicado a las estrategias previas a la carrera y otro que opera durante la misma. Además, se identifican áreas de desarrollo futuro con el objetivo de convertirse en un sistema de apoyo real para la toma de decisiones por parte de las escuderías de la Fórmula 1. Este enfoque promete mejorar la capacidad de análisis y planificación estratégica en el contexto de las carreras automovilísticas de alto rendimiento.*

## **Fecha:**

*13 de junio de 2023, Campus Montegancedo (Boadilla del Monte), Universidad Politécnica de Madrid.*

# Índice

<i>1. Conocimientos Previos</i>	<i>4</i>
<i>2. Descripción del problema a modelizar (Ejemplo Sencillo)</i>	<i>6</i>
<i>3. Sistema de Ayuda a la Decisión</i>	<i>8</i>
<i>4. Modelo</i>	<i>10</i>
<i>5. Cerebro Pre</i>	<i>10</i>
<i>6. Cerebro Race</i>	<i>11</i>
<i>7. Modelos MiniZinc</i>	<i>13</i>
<i>8. Simulador</i>	<i>34</i>
<i>9. Conclusiones</i>	<i>35</i>
<i>10. Líneas futuras</i>	<i>37</i>

## 1. Conocimientos Previos

Primero de todo, véase un enfoque en cuanto al contexto e historia de la Fórmula 1.

La Fórmula 1 (F1) ha sido durante mucho tiempo el epítome de las carreras de automóviles a nivel mundial. Se ha creado un deporte de élite que combina velocidad, habilidad y tecnología de vanguardia. Desde su inicio oficial en 1950, la F1 ha cautivado a millones de espectadores en todo el mundo.

Las carreras de la F1 se organizan en una serie de Grandes Premios, cada uno de los cuales se lleva a cabo en un circuito específico. Estos circuitos pueden ser pistas permanentes, semipermanentes o callejeras, cada una de las cuales presenta su propio conjunto de desafíos.

Los eventos de un Gran Premio generalmente se extienden durante un fin de semana e incluyen prácticas libres, clasificación y la carrera. Las sesiones de práctica libre brindan a los equipos la oportunidad de probar diferentes configuraciones de coches y estrategias. Las sesiones de clasificación determinan la parrilla de salida para la carrera, con el coche más rápido comenzando desde la primera posición, conocida como la "pole position". Finalmente, la carrera se lleva a cabo y los puntos se otorgan a los primeros diez finalistas.

Ahora, véase un enfoque más relacionado con la mecánica y la estrategia de la Fórmula 1.

Los coches de F1 son obras maestras de la ingeniería. Propulsados por motores híbridos complejos, son capaces de alcanzar velocidades de hasta 375 km/h. Sin embargo, a pesar de su aparente simplicidad en el exterior, la F1 es un deporte increíblemente técnico y estratégico.

En el corazón de esta estrategia se encuentra la elección de los neumáticos. Hay cinco tipos diferentes de neumáticos disponibles para los equipos de F1: ultra blandos, super blandos, blandos, medios y duros. Cada uno ofrece diferentes niveles de agarre y resistencia al desgaste, lo que significa que ofrecen diferentes velocidades y tiempos de vida útil. Además, hay dos tipos de neumáticos para condiciones de lluvia: los intermedios para lluvia ligera y los de lluvia extrema para condiciones de mucha agua en pista.



*Tipos de neumáticos en la Fórmula 1*

El desgaste de los neumáticos es un factor crítico en la F1. Con cada vuelta, los neumáticos se desgastan, lo que reduce su rendimiento. Este desgaste puede variar dependiendo del tipo de neumático, las condiciones de la pista y el estilo de conducción del piloto.

La decisión de cuándo cambiar los neumáticos es fundamental en la estrategia de carrera. Un cambio de neumáticos requiere una parada en boxes, lo que implica tiempo y puede alterar la posición de un coche en la carrera. Sin embargo, continuar corriendo con neumáticos desgastados puede resultar en un rendimiento reducido o incluso un pinchazo.

En conclusión, la Fórmula 1 es mucho más que simplemente coches que corren en un circuito. Se trata de una combinación de habilidad de conducción, ingeniería de precisión y estrategia inteligente. Desde las decisiones tomadas en la pista hasta las tomadas en los boxes, cada elemento puede marcar la diferencia entre la victoria y la derrota.

Ahora, se procede a explicar conceptos de estrategia relacionados con la Fórmula 1.

La Fórmula 1 se distingue por su destacada velocidad, sus circuitos que plantean desafíos únicos y sus pilotos de talla mundial. Cada uno de los Grandes Premios es un evento independiente que se desarrolla en un circuito específico. Las condiciones de las carreras varían, pueden desarrollarse bajo condiciones secas o mojadas, dependiendo de las condiciones climáticas prevalecientes.

Como en cualquier competencia, la estrategia juega un papel esencial en la F1. Esta puede entenderse como un plan de acción concebido para conseguir los mejores resultados posibles. En el marco de la F1, esto supone una serie de decisiones tomadas por el equipo y el piloto para optimizar su rendimiento durante una carrera o a lo largo de toda una temporada.

Las decisiones estratégicas en la F1 abarcan desde aspectos tan sencillos como cuándo adelantar a un rival, hasta otros tan complejos como cuándo realizar una parada en boxes para cambiar los neumáticos o reabastecer el combustible. Los neumáticos, de hecho, representan uno de los aspectos más cruciales en la estrategia de la F1. Los neumáticos constituyen el único punto de contacto entre el vehículo y la pista, y distintos tipos de neumáticos ofrecen diferentes niveles de agarre y rendimiento en una variedad de condiciones.

El desgaste de los neumáticos es otro aspecto de gran relevancia. Conforme los neumáticos se utilizan, su rendimiento disminuye, lo cual puede afectar la velocidad y la capacidad de maniobra del vehículo. Por tanto, la decisión de cuándo cambiar los neumáticos durante una carrera puede tener un impacto significativo en el resultado final.

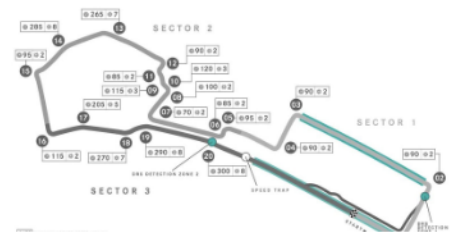
Además de esto, los reglamentos de la FIA (Federación Internacional del Automóvil), que es la entidad encargada de regular la F1, estipulan que los equipos deben utilizar al menos dos tipos de neumáticos en cada carrera que se desarrolle en condiciones secas. Esta regla añade un elemento adicional a la estrategia, ya que los equipos deben considerar cuándo y cómo cambiar los neumáticos durante la carrera para maximizar su rendimiento.



*Símbolo de la FIA*

## 2. Descripción del problema a modelizar (Ejemplo Sencillo)

Se plantea un escenario en el que una serie de variables y factores han sido predefinidos: el piloto es Fernando Alonso, el monoplaza proviene de la escudería Aston Martin, y el lugar de la competición es el desafiante circuito urbano de Azerbaiyán. En el seno de este marco, el objetivo es diseñar la estrategia de carrera óptima para que Alonso consiga la mejor clasificación posible.



*Fernando Alonso, el coche de Aston Martin y el circuito de Azerbaiyán*

Las estrategias de carrera en la Fórmula 1, como se ha explicado anteriormente, son una amalgama de decisiones que requieren análisis y consideración cuidadosa en una amplia gama de factores durante el transcurso de la competición. Estos elementos incluyen, entre otros, la gestión de los neumáticos (incluyendo cuándo cambiarlos y qué tipo de compuesto utilizar), el número y el tiempo de las paradas en boxes, la administración del combustible y la adaptación a la posición en pista.

Para Alonso y su escudería Aston Martin, se ha establecido el desafío de manejar todas estas variables en su intento por maximizar el rendimiento en el tortuoso circuito urbano de Azerbaiyán. A continuación, se tratará de desglosar esta complejidad en componentes más manejables para comprender mejor cómo se orquesta una estrategia de carrera exitosa.

El primer elemento a considerar es la gestión de los neumáticos. Se ha mencionado antes que los neumáticos son vitales en una carrera de F1 debido a su papel en el agarre y rendimiento del vehículo. En un circuito urbano como el de Azerbaiyán, donde las rectas largas se mezclan con giros estrechos y cerrados, la elección del compuesto de neumáticos puede ser decisiva. Además, las condiciones de la pista y el clima también desempeñan un papel crucial en esta decisión. Si se espera que la pista esté seca durante toda la carrera, entonces se deben seleccionar los neumáticos de compuesto seco más adecuados para las condiciones de la pista y las características del monoplaza de Aston Martin.

El siguiente aspecto es la cantidad y el momento de las paradas en boxes. En la F1, cada parada en boxes consume un tiempo precioso que puede alterar la posición de un piloto en la carrera. Por lo tanto, se deben considerar cuidadosamente las circunstancias que justifiquen una parada en boxes, como la necesidad de cambiar de neumáticos, la necesidad de reabastecer de combustible o la necesidad de reparar algún daño en el monoplaza.

Además, la gestión del combustible es otra variable que podría tener un impacto significativo en el rendimiento del monoplaza. En la Fórmula 1, se requiere una cantidad de combustible suficiente para completar la carrera, pero tener demasiado combustible puede hacer que el monoplaza sea más pesado y, por lo tanto, más lento. Es por ello que se debe establecer una estrategia de combustible eficaz, que asegure que el monoplaza tenga suficiente combustible para terminar la carrera, pero sin cargarlo innecesariamente.

Por último, la adaptación a la posición en pista es otro factor que debe considerarse en la estrategia de carrera. En este sentido, se debe tener en cuenta cómo responder a las acciones de los otros pilotos, cómo adelantar o defender la posición y cómo manejar las banderas amarillas o las situaciones de coche de seguridad.

En conclusión, la estrategia de carrera en la Fórmula 1 es un intrincado puzzle que requiere la toma de decisiones rápidas y bien informadas en una multitud de factores. Sin embargo, cuando se lleva a cabo con éxito, puede conducir a la victoria y al éxito, no solo para el piloto, sino también para la escudería. En el caso de Fernando Alonso y Aston Martin, la habilidad para combinar todos estos elementos y tomar las decisiones correctas en el momento adecuado en el circuito de Azerbaiyán será la clave para obtener la mejor clasificación posible.

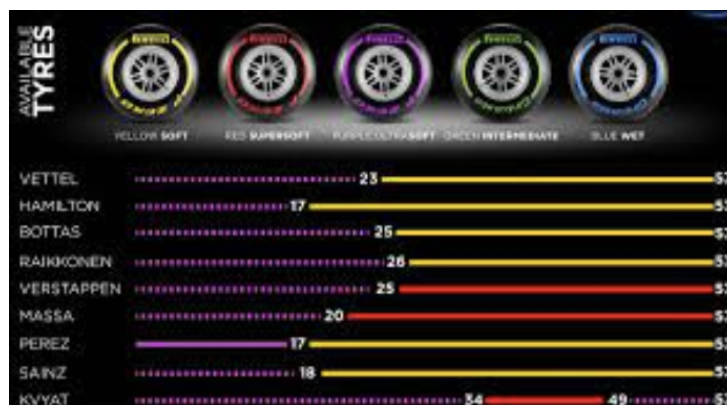
### 3. Sistema de Ayuda a la Decisión

El "Sistema de Ayuda a la Decisión", conocido en la jerga de la Fórmula 1 como "pit stop", es una pieza esencial en la estrategia de carrera en este deporte. Este sistema se encarga de coordinar aspectos críticos como cuándo realizar una parada en boxes y qué acciones llevar a cabo durante la misma. Estas actividades pueden variar desde el cambio de neumáticos y el reabastecimiento de combustible, hasta realizar ajustes técnicos en el monoplaza. Un manejo eficaz de este sistema puede marcar la diferencia entre ganar o perder una carrera.

Para maximizar la efectividad del sistema de ayuda a la decisión, se han incorporado simulaciones de dos "cerebros" distintos en su estructura. El primero, denominado "PRE-RACE", se concentra en la estrategia antes de la carrera. Este "cerebro" se encarga de planificar todas las posibles situaciones y acciones que pueden ocurrir durante la carrera. Estudia variables como las condiciones meteorológicas, el rendimiento histórico del monoplaza y del piloto en el circuito, la degradación esperada de los neumáticos, y la estrategia de combustible, entre otras.

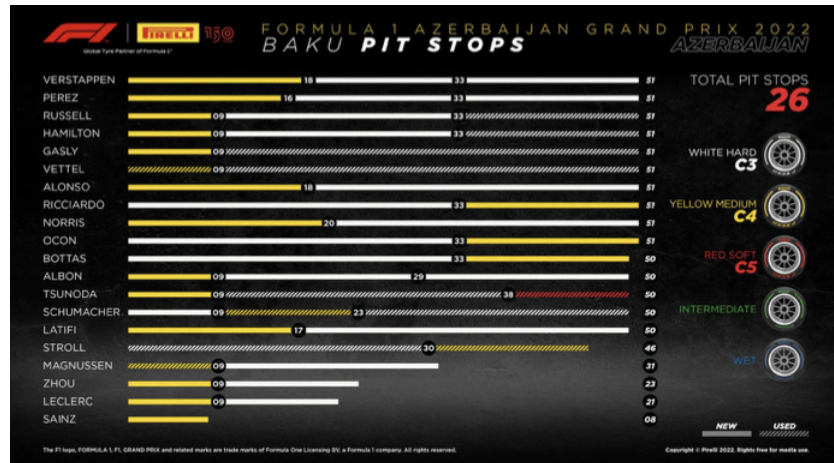
El segundo "cerebro", identificado como "RACE", se ocupa de las decisiones en tiempo real que deben tomarse durante la carrera. Esta parte del sistema es vital para adaptarse a las circunstancias cambiantes de la carrera, como cambios en las condiciones meteorológicas y el rendimiento de los competidores. El "cerebro" RACE evalúa continuamente la situación de la carrera y toma decisiones basadas en la información disponible en tiempo real.

Estos dos "cerebros" trabajan juntos para proporcionar un enfoque integral y adaptable a la estrategia de carrera. Mientras que el "cerebro" PRE-RACE se encarga de establecer un plan general basado en el análisis previo a la carrera, el "cerebro" RACE está allí para adaptar y ajustar ese plan según se desarrolle la carrera. Juntos, se encargan de abordar tanto la planificación anticipada como las respuestas dinámicas a las circunstancias cambiantes de la carrera.



*Estrategia Pre-Carrera para cada piloto*





### *Estrategia Carrera para cada piloto*

Para que este sistema de ayuda a la decisión funcione eficazmente, debe ser alimentado con una gran cantidad de información. Esto incluye datos sobre el rendimiento del monoplaza, las condiciones de la pista, las tácticas de los otros equipos, y mucho más. Con esta información, el sistema puede realizar cálculos rápidos y precisos para ayudar al equipo a tomar las decisiones correctas en el momento adecuado.

El sistema de ayuda a la decisión es, por lo tanto, una herramienta vital en el arsenal de una escudería de Fórmula 1. Si se utiliza correctamente, puede proporcionar a los equipos una ventaja significativa en la pista, ayudándoles a navegar a través de la complejidad de una carrera de Fórmula 1 y a emerger victoriosos.

## 4. Modelo

El modelo se divide en Cerebro Race y Cerebro Pre que son dos módulos que trabajan en conjunto para optimizar la estrategia de neumáticos y conducción en una carrera automovilística.

Cerebro Pre se encarga de determinar una estrategia preliminar basada en la información previa disponible, como el número total de vueltas, las condiciones climáticas, la habilidad del piloto y la posición inicial. Su objetivo principal es establecer una estrategia inicial que maximice el rendimiento y minimice el tiempo total de la carrera. Utilizando algoritmos y reglas predefinidas, Cerebro Pre determina el tipo de neumáticos a utilizar en la primera vuelta y la estrategia de conducción correspondiente.

Por otro lado, Cerebro Race entra en juego durante la carrera en tiempo real. Su objetivo es adaptar la estrategia de neumáticos y conducción en función de la información actualizada que se va obteniendo vuelta a vuelta. Teniendo en cuenta el desgaste de los neumáticos, las condiciones climáticas cambiantes y el rendimiento del piloto, Cerebro Race toma decisiones sobre el tipo de neumáticos a utilizar en cada vuelta y la actitud de conducción más adecuada. Su objetivo principal es optimizar el tiempo por vuelta y minimizar el tiempo total de la carrera, permitiendo así una mayor competitividad y eficiencia en la pista.

En resumen, mientras que Cerebro Pre establece una estrategia preliminar antes de la carrera, Cerebro Race se encarga de ajustar y adaptar esa estrategia en tiempo real durante la competencia. Ambos modelos trabajan en conjunto para lograr un desempeño óptimo en términos de elección de neumáticos y conducción, permitiendo al equipo de carreras tomar decisiones informadas y maximizar sus posibilidades de éxito en la pista.

## 5. Cerebro Pre

Como ya se ha mencionado antes, el objetivo en la pre carrera es minimizar el tiempo total, pues es la única manera de a priori conseguir una mejor posición, ya que no se puede predecir a ciencia cierta si habrá la posibilidad de realizar adelantamientos durante la carrera.

Por tanto la función objetivo en Cerebro Pre será:

$n^{\circ}$  pilotos

$$\sum_{i=1}^{n^{\circ} \text{ pilotos}} T(D, N, C, P, V_{max}, V_{min}, H, T_{parada}) =$$
$$D \times (6 - N) + C + P \times T_{parada} + (V_{max} - V_{min}) \times H + \frac{(V_{max} - V_{min})}{2}$$

La expresión  $T(D, N, C, P, V_{max}, V_{min}, H, T_{parada})$  calcula el tiempo de vuelta de un piloto en función de varios factores. A continuación, se explica cada uno de los términos de la expresión:

- D: Desgaste de neumáticos. Es un valor numérico entre 0 y 100 que representa el desgaste de los neumáticos del vehículo. Cuanto mayor sea este valor, más desgastados estarán los neumáticos, lo que puede afectar el rendimiento y la adherencia del vehículo en la pista. La fórmula utiliza este valor multiplicado por (6-N), donde N representa el tipo de neumático utilizado. Esto implica que a medida que el desgaste de los neumáticos aumenta, el tiempo de vuelta también aumenta, y el efecto se ve agravado por el tipo de neumático utilizado.
- C: Clima. Es un valor numérico que representa las condiciones climáticas durante la carrera. Se utiliza un sistema de puntuación, donde 4 indica lluvia extrema, 2 indica lluvia ligera y 0 indica no lluvia. La expresión simplemente suma este valor al tiempo de vuelta. En general, las condiciones de lluvia pueden afectar la adherencia de los neumáticos y la visibilidad, lo que podría ralentizar el tiempo de vuelta.
- P: PitStop. Es un valor binario que indica si el piloto realiza o no una parada en boxes durante la carrera. Si P es igual a 1, significa que se realiza un PitStop, y si es igual a 0, significa que no se realiza. El tiempo de parada en boxes, representado por  $T_{parada}$ , se multiplica por este valor. Si se realiza un PitStop, esto aumentará el tiempo total de vuelta.
- $(V_{max}-V_{min})/2$ : Esta parte de la expresión simplemente toma la diferencia entre las velocidades máxima y mínima y la divide por 2. Esto se suma al tiempo de vuelta y representa un factor adicional relacionado con la variabilidad de la velocidad del vehículo durante la vuelta.

## 6. Cerebro Race

En cuanto al cerebro race el objetivo es maximizar la ganancia de posición en cada vuelta, para definir la función objetivo “ganancia”  $G_i$ , definimos primero la función ritmo:

$$R_i(H, V_{max}, V_{min}, A, N, D, T_{parada}, P) = D + (6 - N) + P \times T_{parada} + (V_{max} + \frac{(V_{max} - V_{min})}{2}) + H + V_{max} + A$$

La expresión  $R(H, V_{max}, V_{min}, A, N, D, T_{parada}, P)$  calcula el ritmo de carrera de un piloto, teniendo en cuenta diversos factores. A continuación, se explica cada uno de los términos de la expresión:

- D: Desgaste de neumáticos. Es un valor numérico entre 0 y 100 que representa el desgaste de los neumáticos del vehículo. Cuanto mayor sea este valor, más

desgastados estarán los neumáticos, lo que puede afectar el ritmo de carrera del piloto. En la expresión, simplemente se suma este valor.

- (6-N): Representa el tipo de neumático utilizado durante la carrera. N es un número entre 1 y 5. A medida que el número de tipo de neumático disminuye, es decir, a medida que se utiliza un neumático de mayor rendimiento, el ritmo de carrera puede aumentar. Por lo tanto, se resta N del valor 6 y se suma al ritmo de carrera.
- P: PitStop. Es un valor binario que indica si el piloto realiza o no una parada en boxes durante la carrera. Si P es igual a 1, significa que se realiza un PitStop, y si es igual a 0, significa que no se realiza. El tiempo de parada en boxes, representado por  $T_{parada}$ , se multiplica por este valor y se suma al ritmo de carrera. Si se realiza un PitStop, esto puede afectar el ritmo de carrera en función del tiempo perdido en boxes.
- Vmax y Vmin: Representan la velocidad máxima y la velocidad mínima del vehículo durante la carrera, respectivamente. La diferencia entre estas dos velocidades, más la mitad de la velocidad máxima, se suma al ritmo de carrera. A medida que aumenta la diferencia entre las velocidades máxima y mínima, y a medida que aumenta la velocidad máxima en sí misma, el ritmo de carrera puede aumentar.
- H: Habilidad del piloto. Es un valor numérico entre 0 y 100 que representa la habilidad del piloto. A mayor habilidad, es decir, a medida que H aumenta, se suma al ritmo de carrera.
- A: Actitud del piloto. Es un número que representa la actitud del piloto durante la carrera. Se utiliza un sistema de puntuación donde 3 indica una actitud agresiva, 2 indica una actitud normal y 1 indica una actitud conservadora. La expresión simplemente suma este valor al ritmo de carrera. La actitud del piloto puede afectar su estilo de conducción y su enfoque en la carrera.

La función posibilidad  $P_{ij}$ :

$$P_{ij} = R_i(H_i, V_{max}, V_{min}, A_i, N_i, D_i, T_{parada}, P_i) - R_j(H_j, V_{max}, V_{min}, A_j, N_j, D_j, T_{parada}, P_j) - d(i, j)$$

La expresión  $P_{ij}$  calcula la posibilidad de que el piloto i adelante al piloto j en función de su ritmo de carrera y la distancia en segundos entre ellos. A continuación, se explica cada término de la expresión:

- $R_i$  y  $R_j$  representan los ritmos de carrera de los pilotos i y j.
- $d(i, j)$ : Es la distancia en segundos entre el piloto i y el piloto j. Esta distancia se resta de la diferencia entre los ritmos de carrera de los pilotos i y j. Cuanto mayor sea la distancia entre los pilotos, más difícil será para el piloto i adelantar al piloto j.

La expresión  $P_{ij}$  se obtiene al restar el ritmo de carrera del piloto j al ritmo de carrera del piloto i y luego restar la distancia entre ellos. Esta expresión representa la diferencia de

rendimiento entre los pilotos y cómo se ve afectada por la distancia entre ellos. Cuanto mayor sea el resultado de la expresión, mayor será la posibilidad de que el piloto  $i$  adelante al piloto  $j$ .

Por último, la función objetivo, ganancia:

$$G_i = P_{i(i+1)} - P_{(i-1)i}$$

La expresión  $G_i$  se obtiene al restar la posibilidad de adelantamiento del piloto anterior al piloto  $i$  de la posibilidad de adelantamiento del piloto  $i$  al siguiente piloto. Esta expresión muestra la diferencia en términos de posibilidad de adelantamiento entre los dos pilotos vecinos.

La expresión  $G_i$  puede utilizarse para evaluar el rendimiento relativo del piloto  $i$  en relación con los pilotos que lo rodean. Si  $G_i$  es positivo, significa que el piloto  $i$  tiene una mayor posibilidad de adelantar al siguiente piloto en comparación con ser adelantado por el piloto anterior. Por el contrario, si  $G_i$  es negativo, indica que el piloto  $i$  tiene una menor posibilidad de adelantar al siguiente piloto en comparación con ser adelantado por el piloto anterior.

## 7. Modelos MiniZinc

Con los objetivos claros en ambos módulos, ya se puede desarrollar el modelo:

Se definen las siguientes constantes, comunes a ambos modelos, que representan diferentes variables y estados utilizados en las restricciones:

```
% NEUMATICOS
int: DEFAULT = 0;
int: SOFT = 1;
int: MEDIUM = 2;
int: HARD = 3;
int: INTERMEDIUM_WET = 4;
int: EXTREME_WET = 5;
% ESTADO DEL CIRCUITO
int: SECO = 0;
int: MOJADO = 1;
int: MUY_MOJADO = 2;
% CONDUCCIÓN
int: NORMAL = 2;
int: AGRESIVO = 3;
int: CONSERVADOR = 1;
% WEATHER
int: CLEAR = 0;
int: LIGHT_RAIN = 1;
int: HEAVY_RAIN = 2;
% ESTADO DE NEUMÁTICOS
int: REVENTADOS = 0;
int: MUY_GASTADOS = 1;
int: GASTADOS = 2;
```

```

int: USADOS = 3;
int: POCO_USADOS = 4;
int: NUEVOS = 5;
% SHOULD_PIT
int: YES = 1;
int: NO_PIT_STOP = 0;

```

- **NEUMATICOS:** Representa los tipos de neumáticos disponibles. Las constantes DEFAULT, SOFT, MEDIUM, HARD, INTERMEDIUM\_WET y EXTREME\_WET tienen asignados valores enteros del 0 al 5 respectivamente.

- **ESTADO DEL CIRCUITO:** Representa el estado del circuito. Las constantes SECO, MOJADO y MUY\_MOJADO tienen asignados valores enteros del 0 al 2 respectivamente.

- **CONDUCCIÓN:** Representa los niveles de conducción. Las constantes DEFAULT, NORMAL, AGRESIVO y CONSERVADOR tienen asignados valores enteros del 2 al 3 respectivamente.

- **WEATHER:** Representa las condiciones climáticas. Las constantes CLEAR, LIGHT\_RAIN y HEAVY\_RAIN tienen asignados valores enteros del 0 al 2 respectivamente.

- **ESTADO DE NEUMATICOS:** Representa el estado de los neumáticos. Las constantes REVENTADOS, MUY\_GASTADOS, GASTADOS, USADOS, POCO\_USADOS y NUEVOS tienen asignados valores enteros del 0 al 5 respectivamente.

- **SHOULD\_PIT:** Representa si se debe realizar una parada en boxes o no. Las constantes YES y NO\_PIT\_STOP tienen asignados valores enteros 1 y 0 respectivamente.

Estas constantes se utilizan en las restricciones para definir las condiciones y reglas de la estrategia de carrera.

## CEREBRO PRE

Estos son los parámetros de entrada utilizados en el modelo:

```

int: total_laps;
int: weather_index;
int: pilot_hability;
int: posicion;
int: min_speed;
int: max_speed;
int: med_speed;
int: pit_stop_time;

```

- **total\_laps**: Representa el número total de vueltas en la carrera.
- **weather\_index**: Es el índice del clima actual. Es un parámetro auxiliar que indica el estado del clima en la pista.
- **pilot\_hability**: Representa la habilidad del piloto. Es un valor que indica la destreza y experiencia del piloto en la carrera.
- **posicion**: Representa la posición actual del piloto en la carrera.
- **min\_speed**: Es la velocidad mínima permitida en la pista.
- **max\_speed**: Es la velocidad máxima permitida en la pista.
- **med\_speed**: Es la velocidad media en la pista.
- **pit\_stop\_time**: Representa el tiempo necesario para realizar una parada en boxes.

Estos parámetros se utilizan para calcular diferentes aspectos relacionados con la estrategia de carrera y la toma de decisiones en cada vuelta.

Estas son las variables de toma de decisión utilizadas en el modelo:

```
array[1..total_laps] of var 0..5: neumaticos_index;  
var 1..3: conduccion_index;
```

- **neumaticos\_index**: Es un array que representa el tipo de neumáticos a utilizar en cada vuelta de la carrera. El índice del array indica la vuelta correspondiente, y el valor representa el tipo de neumáticos elegido. Los valores posibles van desde 0 hasta 5, donde 0 representa el tipo de neumáticos por defecto (no hay cambio de neumático en esa vuelta) y los valores de 1 a 5 corresponden a los diferentes tipos de neumáticos disponibles.

- **conduccion\_index**: Es una variable que representa la estrategia de conducción a utilizar en la primera vuelta de la carrera. El valor de la variable va desde 1 hasta 3, donde 1 representa una conducción conservadora, 2 una conducción normal y 3 una conducción agresiva.

Estas variables de toma de decisión son utilizadas para determinar la elección de neumáticos y la actitud de conducción en cada vuelta de la carrera.

Estas son las variables auxiliares utilizadas en el modelo:

```
array[1..total_laps] of var 0..5: neumaticos_state;  
array[1..total_laps] of var 0..1: pit_stop;  
array[1..total_laps] of var int: time_per_lap;  
array[1..total_laps] of var 0..100: tire_accumulation;  
array[0..5, 1..3] of int: tire_degradation = array2d(0..5, 1..3,  
  [ % NORMAL, AGRESIVO, CONSERVADOR  
    7, 9, 5, % DEFAULT  
    6, 13, 5, % SOFT  
    6, 7, 5, % MEDIUM  
    7, 8, 5, % HARD  
    6, 13, 3, % INTERMEDIUM_WET  
    8, 17, 4 % EXTREME_WET  
  ];  
);
```

- **neumaticos\_state**: Es un array que representa el estado de los neumáticos a medida que avanzan las vueltas de la carrera. El índice del array indica la vuelta correspondiente, y el valor representa el estado de los neumáticos. Los valores posibles van desde 0 hasta 5, donde 0 representa el estado por defecto de los neumáticos y los valores de 1 a 5 corresponden a diferentes estados de desgaste.

- **pit\_stop**: Es un array que indica si se debe realizar una parada en boxes en la vuelta correspondiente. El índice del array representa la vuelta de la carrera, y el valor de la variable puede ser 0 o 1, donde 0 indica que no se debe realizar una parada en boxes y 1 indica que se debe realizar una parada.

- **time\_per\_lap**: Es un array que representa el tiempo por vuelta en la carrera. El índice del array indica la vuelta correspondiente, y el valor representa el tiempo estimado para completar esa vuelta.

- **tire\_accumulation**: Es un array que representa el nivel de desgaste acumulado de los neumáticos a medida que avanzan las vueltas de la carrera. El índice del array indica la vuelta correspondiente, y el valor representa el nivel de desgaste en una escala del 0 al 100.

-**tire\_degradation**: Se define una matriz que representa la degradación de los neumáticos según el tipo de neumático y la estrategia de conducción.

Estas variables auxiliares son utilizadas para realizar cálculos y seguimiento del estado de los neumáticos, tiempos por vuelta y desgaste acumulado durante la carrera.

Estas son las restricciones y la función objetivo definidas en el modelo:

- La acumulación de desgaste en la primera vuelta debe ser 0.



```
constraint tire_accumulation[1] = 0;
```

- Se establece que se debe realizar una parada en boxes cuando la acumulación de desgaste alcanza un umbral determinado.

```
constraint forall(i in 2..total_laps) (  
  if tire_accumulation[i] >= 60 then  
    pit_stop[i] = YES  
  else  
    pit_stop[i] = NO_PIT_STOP  
  endif  
);
```

- Se establece que si se realiza una parada en boxes, la acumulación de desgaste de los neumáticos se reinicia a 0.

```
constraint forall(i in 2..total_laps) (  
  if pit_stop[i-1] == YES then  
    tire_accumulation[i] = 0  
  else  
    tire_accumulation[i] = tire_accumulation[i-1] +  
tire_degradation[neumaticos_index[i-1], conduccion_index]  
  endif  
);
```

- Se establece que los neumáticos deben estar en estado "NUEVOS" si se realiza una parada en boxes, de lo contrario, estarán en estado "USADOS".

```
constraint forall(i in 2..total_laps) (  
  if pit_stop[i] == YES then  
    neumaticos_state[i] = NUEVOS  
  else  
    neumaticos_state[i] = USADOS  
  endif  
);
```

- Se establece la estrategia de conducción en la primera vuelta, considerando la habilidad del piloto y las condiciones climáticas.

```
constraint (  
  if posicion <= 7 then  
    if pilot_hability <= 90 then
```

```

        if weather_index == CLEAR then
            conduccion_index = AGRESIVO
        elseif weather_index == LIGHT_RAIN then
            conduccion_index = AGRESIVO
        else
            conduccion_index = CONSERVADOR
        endif
    else
        conduccion_index = CONSERVADOR
    endif
else
    if pilot_hability >= 90 then
        if weather_index == CLEAR then
            conduccion_index = AGRESIVO
        elseif weather_index == LIGHT_RAIN then
            conduccion_index = AGRESIVO
        else
            conduccion_index = CONSERVADOR
        endif
    else
        conduccion_index = AGRESIVO
    endif
endif
);

```

- Se establece el tipo de neumático a utilizar en cada vuelta, considerando la estrategia de paradas en boxes y las condiciones climáticas.

```

constraint forall(i in 2..total_laps) (
    if pit_stop[i] == YES then
        if weather_index == CLEAR then
            if neumaticos_index[i-1] == SOFT then
                neumaticos_index[i] == MEDIUM
            elseif neumaticos_index[i-1] == MEDIUM then
                neumaticos_index[i] = SOFT
            elseif neumaticos_index[i-1] == MEDIUM then
                neumaticos_index[i] == MEDIUM
            else
                neumaticos_index[i] == SOFT
            endif
        elseif weather_index == LIGHT_RAIN then
            if neumaticos_index[i-1] == INTERMEDIUM_WET then
                neumaticos_index[i] == EXTREME_WET
            else
                neumaticos_index[i] == INTERMEDIUM_WET
            endif
        else
            if neumaticos_index[i-1] == EXTREME_WET then
                neumaticos_index[i] == SOFT /\ neumaticos_index[i] = INTERMEDIUM_WET
            end
        end
    end
);

```

```

        else
            neumaticos_index[i] == EXTREME_WET
        endif
    endif
    else
        neumaticos_index[i] == neumaticos_index[i-1]
    endif
end;

```

- Se establece el tipo de neumático a utilizar en la primera vuelta, considerando las condiciones climáticas y la posición del piloto.

```

constraint (
    if weather_index == CLEAR then
        if posicion >= 7 then
            neumaticos_index[1] = SOFT
        else
            neumaticos_index[1] = MEDIUM
        endif
    elseif weather_index == LIGHT_RAIN then
        if posicion >= 7 then
            neumaticos_index[1] = MEDIUM
        else
            neumaticos_index[1] = INTERMEDIUM_WET
        endif
    else
        neumaticos_index[1] = EXTREME_WET
    endif
end;

```

- Se establece que se debe realizar una parada en boxes y cambiar los neumáticos si están desgastados y las condiciones climáticas son adversas o si están muy desgastados.

```

constraint forall(i in 2..total_laps) (
    if (neumaticos_state[i-1] == REVENTADOS /\ (weather_index == LIGHT_RAIN \/ weather_index ==
HEAVY_RAIN)) \/ neumaticos_state[i-1] == DEFAULT then
        pit_stop[i-1] = YES /\ neumaticos_state[i] = NUEVOS
    else
        true
    endif
end;

```

- Se establece una parada obligatoria en la penúltima vuelta si no se ha realizado ninguna parada en boxes.

```

constraint (

```

```

if sum(i in 1..total_laps - 1) (pit_stop[i] = YES) = 0 then
    pit_stop[total_laps - 1] = YES
else
    true
endif
);

```

- Se establece que no se puede cambiar al mismo tipo de neumático en la parada en boxes.

```

constraint forall(i in 2..total_laps) (
    if pit_stop[i] == YES then
        neumaticos_index[i] != neumaticos_index[i-1]
    endif
);

```

- Se calcula el tiempo por vuelta en función de las condiciones, considerando el desgaste de los neumáticos, las condiciones climáticas, la habilidad del piloto, el rango de velocidad y el tiempo de la parada en boxes.

```

constraint forall(i in 1..total_laps) (
    let {
        var int: tire_factor = neumaticos_index[i] * 2 * (100 - tire_accumulation[i]) div 100,
        var int: weather_factor = (weather_index = HEAVY_RAIN) * 4 + (weather_index = LIGHT_RAIN) *
2,
        var int: pilot_factor = (pilot_hability - 60) div 40,
        var int: speed_range = max_speed - min_speed,
        var int: pit_stop_factor = if pit_stop[i] == YES then pit_stop_time div 1000 else 0 endif
    } in (
        time_per_lap[i] = med_speed + tire_factor + weather_factor + pilot_factor * speed_range +
pit_stop_factor
    )
);

```

- Objetivo: Se minimiza el tiempo total sumando los tiempos por vuelta.

```

var int: total_time = sum(time_per_lap);
solve minimize total_time;

```

Estas restricciones y la función objetivo permiten modelar y resolver el problema de optimización para determinar la estrategia de neumáticos y conducción óptima en una carrera a priori.

## CEREBRO RACE

Para cerebro race se utilizan los siguientes parametros de entrada:

```
int: remaining_laps;
int: vueltas_sigiente_parada;
int: current_position;
int: weather_index;
int: estado_circuito;
array[1..20] of int: degradacion;
array[1..20] of int: conduccion_actual;
array[1..20] of int: neumaticos_actual;
array[1..20] of int: seconds_between;
array[1..20] of int: habilidad_piloto;
array[1..20] of int: velocidad_media;
array[1..20] of int: velocidad_maxima;
array[1..20] of int: velocidad_minima;
array[1..20] of int: equipo;
array[1..20] of int: tiempo_pitstop;
int: n_paradas;
array[1..5] of int: neumaticos_usados;
```

- **remaining\_laps**: Representa el número de vueltas restantes en la carrera.
- **vueltas\_sigiente\_parada**: Indica el número de vueltas hasta la próxima parada en boxes.
- **current\_position**: Representa la posición actual del piloto en la carrera.
- **weather\_index**: Es un índice que indica el estado del clima actual.
- **estado\_circuito**: Representa el estado del circuito en términos de condiciones de la pista.
- **degradacion**: Es un array que almacena el nivel de degradación de los neumáticos para cada piloto.
- **conduccion\_actual**: Es un array que indica el tipo de conducción actual de cada piloto.
- **neumaticos\_actual**: Es un array que indica los neumáticos actuales utilizados por cada piloto.
- **seconds\_between**: Es un array que almacena el tiempo en segundos entre cada piloto en relación con nuestro piloto actual.
- **habilidad\_piloto**: Es un array que representa la habilidad de cada piloto en la carrera.
- **velocidad\_Media**: Es un array que indica la velocidad media de cada piloto.
- **velocidad\_maxima**: Es un array que indica la velocidad máxima alcanzada por cada piloto.
- **velocidad\_minima**: Es un array que indica la velocidad mínima alcanzada por cada piloto.
- **equipo**: Es un array que representa el equipo al que pertenece cada piloto.
- **tiempo\_pitstop**: Es un array que almacena el tiempo de parada en boxes de cada piloto.
- **n\_paradas**: Indica el número de paradas en boxes realizadas hasta el momento.
- **neumaticos\_usados**: Es un array que registra la cantidad de neumáticos usados por tipo.

Estos parámetros se utilizan en el código para realizar cálculos y evaluaciones con el fin de tomar decisiones estratégicas durante la carrera, como planificar paradas en boxes, evaluar el estado de los neumáticos y comparar habilidades y desempeño entre los pilotos.

En el código proporcionado, se definen varias variables auxiliares y arrays que se utilizan para realizar cálculos y evaluar restricciones en el contexto de una estrategia de carreras en la Fórmula 1. A continuación, se explica cada uno de ellos:

```
int: delante = if current_position == 1 then 20 else current_position - 1 endif;  
int: piloto = current_position;  
int: detras = if current_position == 20 then 1 else current_position + 1 endif;
```

Variables auxiliares:

- **delante**: Almacena la posición justo delante del piloto actual. Si el piloto está en la posición 1, se asigna la posición 20, de lo contrario, se resta 1 a la posición actual.
- **piloto**: Representa la posición actual del piloto en la carrera.
- **detras**: Almacena la posición justo detrás del piloto actual. Si el piloto está en la posición 20, se asigna la posición 1, de lo contrario, se suma 1 a la posición actual.

```
array[1..20] of int: calidad_coche = [velocidad_maxima[i] + velocidad_medio[i] | i in 1..20];  
array[1..20] of int: prediction_stop = [if estado_neumaticos[i] <= MUY_GASTADOS then PARA  
    else if estado_neumaticos[i] <= USADOS /\ conduccion_actual[i] ==  
    AGRESIVO then PARA  
    else NO_PARA  
    endif endif | i in 1..20];  
array[1..20] of int: prob_pinchazo = [degradacion[i] * conduccion_actual[i] | i in 1..20];  
array[1..20] of int: estado_neumaticos = [if degradacion[i] > 90 then REVENTADOS  
    else if degradacion[i] >= 70 then MUY_GASTADOS  
    else if degradacion[i] >= 50 then GASTADOS  
    else if degradacion[i] >= 30 then USADOS  
    else if degradacion[i] >= 20 then POCO_USADOS  
    else NUEVOS  
    endif endif endif endif | i in 1..20];
```

Arrays:

- **calidad\_coche**: Almacena la calidad del coche de cada piloto. Se calcula sumando la velocidad máxima y la velocidad media de cada piloto.
- **prediction\_stop**: Indica la predicción de parada en la siguiente vuelta para cada piloto. Dependiendo del estado de los neumáticos y el tipo de conducción actual, se asigna el valor "PARA" si se predice que el piloto debe parar, de lo contrario, se asigna "NO\_PARA".
- **prob\_pinchazo**: Representa la probabilidad de pinchazo en la siguiente vuelta para cada piloto. Se calcula multiplicando la degradación de los neumáticos por el tipo de conducción actual.

- **estado\_neumaticos**: Almacena el estado de los neumáticos para cada piloto. Se asigna una categoría de estado (como "REVENTADOS", "MUY\_GASTADOS", "GASTADOS", etc.) según el nivel de degradación de los neumáticos.

Variables auxiliares para restricciones:

```
bool: parada_pronto = vueltas_sigiente_parada <= 2 /\ vueltas_sigiente_parada > 0;
bool: mi_equipo_delante = equipo[delante] == equipo[piloto];
bool: mi_equipo_detras = equipo[detras] == equipo[piloto];
bool: mas_habilidad_que_delante = habilidad_piloto[piloto] >= habilidad_piloto[delante];
bool: mas_habilidad_que_detras = habilidad_piloto[piloto] >= habilidad_piloto[detras];
bool: mejor_coche_que_delante = calidad_coche[piloto] >= calidad_coche[delante];
bool: mejor_coche_calidad_que_detras = calidad_coche[piloto] >= calidad_coche[detras];
bool: muy_cerca_delante = seconds_between[delante] <= 1000;
bool: muy_cerca_detras = seconds_between[detras] <= 1000;
bool: cerca_delante = seconds_between[delante] <= 3500;
bool: cerca_detras = seconds_between[detras] <= 3500;
bool: mas_bueno_que_delante = not mi_equipo_delante /\ mas_habilidad_que_delante;
bool: mas_bueno_que_detras = not mi_equipo_detras /\ mas_habilidad_que_detras;
bool: final_carrera = remaining_laps <= 15;
bool: mejor_parada_que_delante = prediction_stop[delante] == PARA /\ tiempo_pitstop[piloto] <
tiempo_pitstop[delante];
bool: mejor_parada_que_detras = prediction_stop[detras] == PARA /\ tiempo_pitstop[piloto] <
tiempo_pitstop[detras];
bool: posible_adelantar = calidad_coche[piloto] - calidad_coche[delante] +
habilidad_piloto[piloto] - habilidad_piloto[delante] > 0;
bool: posible_adelantado = calidad_coche[detras] - calidad_coche[piloto] +
habilidad_piloto[detras] - habilidad_piloto[piloto] > 0;
bool: parada_obligatoria_hecha = n_paradas > 0;
bool: parada_obligatoria = not parada_obligatoria_hecha /\ remaining_laps == 1;
```

Estas variables auxiliares se utilizan para evaluar diferentes condiciones y restricciones en la estrategia de carrera. Por ejemplo, si el piloto debe parar pronto en boxes, si el piloto está en el mismo equipo que el piloto que está delante o detrás, si el piloto tiene más habilidad que el piloto delante o detrás, si el coche del piloto es mejor en calidad que el coche delante o detrás, entre otras restricciones y condiciones relacionadas con las posiciones y rendimiento de los pilotos.

Se definen las siguientes funciones auxiliares relacionadas con la gestión de neumáticos en una estrategia de carreras. A continuación, se explica el propósito y el funcionamiento de cada una de ellas:

```
function int: neumatico_next_lap(int: intencion, array[1..5] of int: neumaticos_usados) =
  if intencion == 3 then
    if neumaticos_usados[1] == 0 then SOFT
    else if neumaticos_usados[2] != 0 then SOFT
    else if neumaticos_usados[3] != 0 then SOFT
```

```

else if neumaticos_usados[4] != 0 then SOFT
else if neumaticos_usados[5] != 0 then SOFT
else if neumaticos_usados[2] == 0 then MEDIUM
else if neumaticos_usados[3] == 0 then HARD
else SOFT
endif endif endif endif endif endif endif
else if intencion == 2 then
if neumaticos_usados[2] == 0 then MEDIUM
else if neumaticos_usados[1] != 0 then MEDIUM
else if neumaticos_usados[3] != 0 then MEDIUM
else if neumaticos_usados[4] != 0 then MEDIUM
else if neumaticos_usados[5] != 0 then MEDIUM
else if neumaticos_usados[3] == 0 then HARD
else if neumaticos_usados[1] == 0 then SOFT
else MEDIUM
endif endif endif endif endif endif endif
else if intencion == 1 then
if neumaticos_usados[3] == 0 then HARD
else if neumaticos_usados[1] != 0 then HARD
else if neumaticos_usados[2] != 0 then HARD
else if neumaticos_usados[4] != 0 then HARD
else if neumaticos_usados[5] != 0 then HARD
else if neumaticos_usados[2] == 0 then MEDIUM
else if neumaticos_usados[1] == 0 then SOFT
else HARD
endif endif endif endif endif endif endif
else if intencion == -1 then
if neumaticos_usados[1] == 0 then SOFT
else if neumaticos_usados[2] == 0 then MEDIUM
else if neumaticos_usados[3] == 0 then HARD
else if neumaticos_usados[4] == 0 then MEDIUM
else if neumaticos_usados[5] == 0 then MEDIUM
else SOFT
endif endif endif endif endif
else DEFAULT
endif endif endif endif;

```

Función **neumatico\_next\_lap(intencion, neumaticos\_usados)**:

Esta función se utiliza para determinar el tipo de neumático que se debe utilizar en la siguiente vuelta, según la intención y los neumáticos usados hasta el momento. Toma dos argumentos:

- **intencion**: Representa la intención del equipo en cuanto al tipo de neumático a utilizar en la siguiente vuelta. Los posibles valores son:
  - 3: Indica una intención de usar neumáticos "SOFT" (blandos).
  - 2: Indica una intención de usar neumáticos "MEDIUM" (medios).
  - 1: Indica una intención de usar neumáticos "HARD" (duros).
  - -1: Indica una intención de usar el mismo tipo de neumático que en la vuelta anterior.



- **neumaticos\_usados**: Es un array que almacena la cantidad de neumáticos usados para cada tipo en las vueltas anteriores.

La función utiliza una serie de condicionales anidados para determinar el tipo de neumático que se debe utilizar en la siguiente vuelta, teniendo en cuenta la intención del equipo y los neumáticos usados hasta ese momento. Por ejemplo, si la intención es usar neumáticos "SOFT", la función verifica si hay neumáticos "SOFT" no utilizados en las vueltas anteriores y, en ese caso, devuelve "SOFT". Si no hay neumáticos "SOFT" disponibles, comprueba si hay neumáticos "MEDIUM" no utilizados y, de ser así, devuelve "MEDIUM". Si no se cumplen ninguna de las condiciones anteriores, devuelve "HARD". Si la intención es utilizar el mismo tipo de neumático que en la vuelta anterior, la función devuelve el mismo tipo de neumático que se utilizó en la última vuelta.

```
function array[1..6] of int: neumatico_usable(array[1..5] of int: neumatico) =  
  [if neumatico[1] == 0 then 1 else if (neumatico[2] != 0 \\/ neumatico[3] != 0 \/  
neumatico[4] != 0 \\/ neumatico[5] != 0) then 1 else -1 endif endif  
  ,if neumatico[2] == 0 then 2 else if (neumatico[1] != 0 \\/ neumatico[3] != 0 \/  
neumatico[4] != 0 \\/ neumatico[5] != 0) then 2 else -1 endif endif  
  ,if neumatico[3] == 0 then 3 else if (neumatico[1] != 0 \\/ neumatico[2] != 0 \/  
neumatico[4] != 0 \\/ neumatico[5] != 0) then 3 else -1 endif endif  
  ,if neumatico[4] == 0 then 4 else if (neumatico[1] != 0 \\/ neumatico[2] != 0 \/  
neumatico[3] != 0 \\/ neumatico[4] != 0) then 4 else -1 endif endif  
  ,if neumatico[5] == 0 then 5 else if (neumatico[1] != 0 \\/ neumatico[2] != 0 \/  
neumatico[3] != 0 \\/ neumatico[4] != 0) then 5 else -1 endif endif,0];
```

### **Función `neumatico_usable(neumatico)`:**

Esta función se utiliza para determinar qué neumáticos son utilizables en función de los neumáticos disponibles en cada posición del array `neumatico`. Toma un argumento:

- **neumatico**: Es un array que representa la disponibilidad de neumáticos en cada posición. Un valor de 0 indica que el neumático correspondiente está disponible.

La función devuelve un array que indica la utilizabilidad de cada posición del neumático. Los valores de retorno son los siguientes:

- distinto de -1 : Indica que el neumático en esa posición es utilizable.
- -1: Indica que el neumático en esa posición no es utilizable porque hay otros neumáticos disponibles en otras posiciones.

La función utiliza una serie de condicionales para determinar si un neumático en una posición específica es utilizable o no de acuerdo con la normativa FIA que obliga a usar en cada carrera al menos 2 compuestos diferentes. Comprueba si el neumático en la posición correspondiente es igual a 0. Si es así, se verifica si hay neumáticos disponibles en las demás posiciones.

Los parámetros adicionales que se muestran son variables utilizadas en la toma de decisiones de la estrategia de carrera. A continuación, se explica el propósito de cada uno de ellos:

```
int: factor_adelantar = if (current_position == 1) then 0 else 1 endif;  
int: factor_perder = if (current_position == 20) then 0 else 1 endif;
```

- **factor\_adelantar**: Es una variable que indica el factor de peso para adelantar a otros pilotos. Si la posición actual del piloto es la primera (es decir, el piloto está en la posición de liderazgo), se establece el valor en 0, lo que implica que no se considerará el factor de adelantar en las decisiones. De lo contrario, si el piloto se encuentra en una posición diferente de la primera, se establece el valor en 1, lo que indica que el factor de adelantar sí se tendrá en cuenta.

- **factor\_perder**: Es una variable similar a factor\_adelantar, pero indica el factor de peso para evitar perder posiciones. Si la posición actual del piloto es la última (es decir, el piloto está en la posición más baja), se establece el valor en 0, lo que implica que no se considerará el factor de perder en las decisiones. De lo contrario, si el piloto se encuentra en una posición diferente de la última, se establece el valor en 1, lo que indica que el factor de perder sí se tendrá en cuenta.

```
var 0..5: neumatico_next_lap_index;  
var 1..3: conduccion_next_lap_index;
```

- **neumatico\_next\_lap\_index**: Es una variable de decisión que representa el índice del tipo de neumático que se utilizará en la siguiente vuelta. Puede tomar valores en el rango de 0 a 5, donde 0 representa que no se desea cambiar el neumático.

- **conduccion\_next\_lap\_index**: Es una variable de decisión que representa el índice del tipo de conducción que se utilizará en la siguiente vuelta. Puede tomar valores en el rango de 1 a 3, donde 1, 2 y 3 representan diferentes tipos de conducción.

Estas variables se utilizan para tomar decisiones óptimas en función de las condiciones y estrategias de la carrera. Dependiendo de los valores asignados a estas variables, se determinará el tipo de neumático y el estilo de conducción a utilizar en la siguiente vuelta.

La función **calcular\_adelantamiento** es una función que toma dos parámetros: el índice del piloto que adelanta (**adelantador**) y el índice del piloto que está siendo adelantado (**adelantado**). Calcula un valor numérico que representa la capacidad de adelantamiento del piloto adelantador sobre el piloto adelantado. Representa las funciones  $R_i$  y  $P_{ij}$  definidas anteriormente.

```
function var int: calcular_adelantamiento(int: adelantador, int: adelantado) =
    (habilidad_piloto[adelantador] - habilidad_piloto[adelantado] +
    calidad_coche[adelantador] - calidad_coche[adelantado] + velocidad_maxima[adelantador] -
    velocidad_maxima[adelantado] + conduccion_next_lap_index - conduccion_actual[adelantado] +
    nuevo_estado_neumaticos - degradacion[adelantado]);
var int: diferencia_pit_stop_delante = parada*tiempo_pitstop[piloto] -
prediction_stop[delante]*tiempo_pitstop[delante];
var int: diferencia_pit_stop_detras = prediction_stop[detras]*tiempo_pitstop[detras] -
parada*tiempo_pitstop[piloto];
var int: adelantar = calcular_adelantamiento(piloto,delante) - seconds_between[delante] -
diferencia_pit_stop_delante + prob_pinchazo[delante];
var int: perder = -(calcular_adelantamiento(detras,piloto)) - seconds_between[detras] -
diferencia_pit_stop_detras - prob_pinchazo[detras];
var int: objetivo = factor_adelantar*adelantar - factor_perder*perder;
var int: nueva_degradacion = if (neumatico_next_lap_index == DEFAULT) then
degradacion[piloto] else 0 endif;
var int: nuevo_estado_neumaticos = nueva_degradacion;
var int: parada = if (neumatico_next_lap_index == DEFAULT) then 0 else 1 endif;
```

La función calcula este valor sumando y restando varias variables y parámetros, incluyendo:

- La diferencia de habilidad entre el piloto adelantador y el piloto adelantado ( $\text{habilidad\_piloto}[\text{adelantador}] - \text{habilidad\_piloto}[\text{adelantado}]$ ).
- La diferencia de calidad del coche entre el piloto adelantador y el piloto adelantado ( $\text{calidad\_coche}[\text{adelantador}] - \text{calidad\_coche}[\text{adelantado}]$ ).
- La diferencia de velocidad máxima entre el piloto adelantador y el piloto adelantado ( $\text{velocidad\_maxima}[\text{adelantador}] - \text{velocidad\_maxima}[\text{adelantado}]$ ).
- La diferencia de tipo de conducción entre la conducción prevista para la siguiente vuelta ( $\text{conduccion\_next\_lap\_index}$ ) y la conducción actual del piloto adelantado ( $\text{conduccion\_actual}[\text{adelantado}]$ ).
- La diferencia en el estado de los neumáticos entre el estado previsto para la siguiente vuelta ( $\text{nuevo\_estado\_neumaticos}$ ) y el estado actual de los neumáticos del piloto adelantado ( $\text{degradacion}[\text{adelantado}]$ ).

La función también tiene otras variables y parámetros que se utilizan para calcular la diferencia en las paradas en boxes y las probabilidades de pinchazo. Al final, la función retorna el valor calculado para representar la capacidad de adelantamiento del piloto adelantador sobre el piloto adelantado.

Además de la función `calcular_adelantamiento`, se definen otras variables que se utilizan para el cálculo de la estrategia de carrera:

- **diferencia\_pit\_stop\_delante:** Representa la diferencia en tiempo de pit stop entre el piloto actual y el piloto que está delante. Se calcula multiplicando el valor de parada por el tiempo de pit stop del piloto actual (`tiempo_pitstop[piloto]`) y restando el producto de `prediction_stop[delante]` (predicción de parada del piloto delante) por el tiempo de pit stop del piloto delante (`tiempo_pitstop[delante]`).

- **diferencia\_pit\_stop\_detrás:** Representa la diferencia en tiempo de pit stop entre el piloto actual y el piloto que está detrás. Se calcula multiplicando el producto de `prediction_stop[detrás]` (predicción de parada del piloto detrás) por el tiempo de pit stop del piloto detrás (`tiempo_pitstop[detrás]`) y restando el producto de parada por el tiempo de pit stop del piloto actual (`tiempo_pitstop[piloto]`).

Estructuración del objetivo:

```
var int: adelantar = calcular_adelantamiento(piloto,delante) - seconds_between[delante] -
diferencia_pit_stop_delante + prob_pinchazo[delante];
var int: perder = -(calcular_adelantamiento(detrás,piloto)) - seconds_between[detrás] -
diferencia_pit_stop_detrás - prob_pinchazo[detrás];
var int: objetivo = factor_adelantar*adelantar - factor_perder*perder;
var int: nueva_degradacion = if (neumatico_next_lap_index == DEFAULT) then degradacion[piloto]
else 0 endif;
var int: nuevo_estado_neumaticos = nueva_degradacion;
var int: parada = if (neumatico_next_lap_index == DEFAULT) then 0 else 1 endif;
```

- **adelantar:** Representa la capacidad de adelantamiento del piloto actual sobre el piloto que está delante. Se calcula restando la función `calcular_adelantamiento` entre el piloto actual y el piloto delante, el tiempo transcurrido entre ellos (`seconds_between[delante]`), la diferencia en el tiempo de pit stop delante (`diferencia_pit_stop_delante`), y la probabilidad de pinchazo del piloto delante (`prob_pinchazo[delante]`).

- **perder:** Representa la capacidad de adelantamiento del piloto detrás sobre el piloto actual. Se calcula restando la negación de la función `calcular_adelantamiento` entre el piloto detrás y el piloto actual, el tiempo transcurrido entre ellos (`seconds_between[detrás]`), la diferencia en el tiempo de pit stop detrás (`diferencia_pit_stop_detrás`), y la probabilidad de pinchazo del piloto detrás (`prob_pinchazo[detrás]`).

- **objetivo:** Representa el objetivo de la estrategia de carrera, que es tratar de no perder posiciones y ganarlas si es posible. Se calcula multiplicando el factor de adelantamiento (factor\_adelantar) por adelantar y restando el producto del factor de perder (factor\_perder) por perder.

- **nueva\_degradacion:** Representa la degradación prevista para los neumáticos en la siguiente vuelta. Si el índice del tipo de neumático para la siguiente vuelta (neumatico\_next\_lap\_index) es el valor por defecto (DEFAULT), se asigna el valor actual de degradación del piloto (degradacion[piloto]), de lo contrario, se asigna 0.

- **nuevo\_estado\_neumaticos:** Representa el nuevo estado de los neumáticos para la siguiente vuelta. Se asigna el valor de nueva\_degradacion.

- **parada:** Representa si se realizará una parada en boxes en la siguiente vuelta. Si el índice del tipo de neumático para la siguiente vuelta (neumatico\_next\_lap\_index) es el valor por defecto (DEFAULT), se asigna 0, indicando que no se realizará una parada. De lo contrario, se asigna 1, indicando que se realizará una parada.

Estas variables y funciones se utilizan para tomar decisiones estratégicas en la carrera, considerando factores como la habilidad del piloto, la calidad del coche, la velocidad máxima, el tipo de conducción, el estado de los neumáticos, las paradas en boxes y las probabilidades de pinchazo.

En la realidad hay diferentes reglas y condiciones relacionadas con el clima, el cambio de neumáticos y la estrategia de conducción que limitan la elección de las variables de decisión. Vamos a revisar cada una de ellas, en este módulo se pueden dividir en dos tipos, relacionadas con las condiciones físicas e intrínsecas a la carrera (decisiones no obligadas por las condiciones físicas pero que intentan buscar beneficio para el piloto):

### Físicas:

- Si no está lloviendo y el suelo está seco, no se pueden usar neumáticos de lluvia. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta no puede ser INTERMEDIUM\_WET ni EXTREME\_WET.

```
constraint (weather_index == CLEAR /\ estado_circuito == SECO) ->
  (neumatico_next_lap_index != INTERMEDIUM_WET /\
   neumatico_next_lap_index != EXTREME_WET);
```

- Si está lloviendo o el suelo está muy mojado, no se pueden usar neumáticos de seco. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser EXTREME\_WET.

```
constraint ((estado_circuito == MUY_MOJADO \/ weather_index == HEAVY_RAIN) /\
neumaticos_actual[piloto] < EXTREME_WET) ->
    (neumatico_next_lap_index = EXTREME_WET);
```

- Si está lloviendo poco o el suelo está mojado, no se pueden usar neumáticos de seco. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser INTERMEDIUM\_WET.

```
constraint ((weather_index == LIGHT_RAIN \/ estado_circuito == MOJADO) /\
neumaticos_actual[piloto] != INTERMEDIUM_WET) ->
    (neumatico_next_lap_index = INTERMEDIUM_WET);
```

- Si los neumáticos están muy gastados, se deben cambiar. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser diferente al valor por defecto (DEFAULT).

```
constraint (estado_neumaticos[piloto] <= MUY_GASTADOS) ->
    (neumatico_next_lap_index > DEFAULT);
```

- Si los neumáticos están gastados y se va a realizar una parada en la siguiente vuelta, no se deben cambiar. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser el valor por defecto (DEFAULT).

```
constraint (not parada_obligatoria /\ (parada_pronto /\ estado_neumaticos[piloto] >
MUY_GASTADOS)) ->
    (neumatico_next_lap_index == DEFAULT);
```

### Intrínsecas a la carrera:

- Si estamos cerca del piloto que está detrás y somos agresivos, cambiamos la actitud a más agresiva para defendernos. Esto se traduce en la restricción de que el índice de la conducción para la siguiente vuelta no puede ser CONSERVADOR.

```
constraint (muy_cerca_detras /\ ((not mi_equipo_detras) \/ (posible_adelantado))) ->
    (conduccion_next_lap_index != CONSERVADOR);
```

- Si no estamos cerca de ningún piloto delante ni detrás, somos conservadores. Esto se traduce en la restricción de que el índice de la conducción para la siguiente vuelta no puede ser AGRESIVO.

```
constraint (not cerca_delante /\ not cerca_detras) ->
    (conduccion_next_lap_index != AGRESIVO);
```

- Si estamos cerca de algún piloto, somos menos conservadores a menos que sea un piloto de nuestro equipo. Esto se traduce en la restricción de que el índice de la conducción para la siguiente vuelta debe ser menor que AGRESIVO.

```
constraint ((cerca_detras /\ not (mas_bueno_que_detras)) /\ (cerca_delante /\ not
(mas_bueno_que_delante))) ->
    (conduccion_next_lap_index < AGRESIVO);
```

- Si no somos mejores que el piloto delante y somos del mismo equipo, somos conservadores. Esto se traduce en la restricción de que el índice de la conducción para la siguiente vuelta no puede ser AGRESIVO.

```
constraint (not (mas_bueno_que_delante)) ->
    (conduccion_next_lap_index != AGRESIVO);
```

- No podemos ser agresivos si los neumáticos están muy gastados o reventados. Esto se traduce en la restricción de que el índice de la conducción para la siguiente vuelta debe ser menor que NORMAL.

```
constraint (estado_neumaticos[piloto]<= MUY_GASTADOS) ->
    (conduccion_next_lap_index < NORMAL);
```

- Si estamos a menos de 5 segundos del piloto delante, somos agresivos, creemos que el piloto delante va a parar, tenemos los neumáticos en buen estado y es posible adelantar, paramos y cambiamos a neumáticos blandos. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser menor o igual a `neumatico_next_lap(3,neumaticos_usados)` y el índice de la conducción para la siguiente vuelta no puede ser CONSERVADOR.

```
constraint (not parada_obligatoria /\ (conduccion_actual[piloto] == AGRESIVO /\ final_carrera
/\ cerca_delante /\ posible_adelantar /\ estado_neumaticos[piloto] > MUY_GASTADOS /\
    mejor_parada_que_delante /\ estado_neumaticos[piloto]<= USADOS)) ->
    (neumatico_next_lap_index <= neumatico_next_lap(3,neumaticos_usados) /\
conduccion_next_lap_index != CONSERVADOR);
```

- Si estamos a menos de 2 segundos del piloto delante, somos agresivos, creemos que el piloto delante va a parar, tenemos los neumáticos en buen estado y es posible adelantar, paramos y cambiamos a neumáticos blandos. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser menor o igual a `neumatico_next_lap(3,neumaticos_usados)` y el índice de la conducción para la siguiente vuelta debe ser AGRESIVO.

```
constraint (not parada_obligatoria /\ (conduccion_actual[piloto] <= AGRESIVO /\ final_carrera
/\ muy_cerca_delante /\ posible_adelantar /\ estado_neumaticos[piloto] > MUY_GASTADOS /\
mejor_parada_que_delante /\ estado_neumaticos[piloto]<= USADOS)) ->
(neumatico_next_lap_index <= neumatico_next_lap(3,neumaticos_usados) /\
conduccion_next_lap_index = AGRESIVO);
```

- Si estamos a tiro del piloto detrás, somos agresivos, creemos que el piloto detrás va a parar, tenemos los neumáticos en buen estado y es posible adelantar, paramos y cambiamos a neumáticos blandos. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser menor o igual a `neumatico_next_lap(3,neumaticos_usados)` y el índice de la conducción para la siguiente vuelta no puede ser CONSERVADOR.

```
constraint (not parada_obligatoria /\ (conduccion_actual[piloto] == AGRESIVO /\ final_carrera
/\ cerca_detras /\ mas_bueno_que_detras /\ posible_adelantado /\ estado_neumaticos[piloto] >
MUY_GASTADOS /\
mejor_parada_que_detras /\ estado_neumaticos[piloto]<= USADOS)) ->
(neumatico_next_lap_index <= neumatico_next_lap(3,neumaticos_usados) /\
conduccion_next_lap_index != CONSERVADOR);
```

- Si estamos a menos de 2 segundos del piloto detrás, somos agresivos, creemos que el piloto detrás va a parar, tenemos los neumáticos en buen estado y es posible adelantar, paramos y cambiamos a neumáticos blandos. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe ser menor o igual a `neumatico_next_lap(3,neumaticos_usados)` y el índice de la conducción para la siguiente vuelta debe ser AGRESIVO.

```
constraint (not parada_obligatoria /\ (conduccion_actual[piloto] <= AGRESIVO /\ final_carrera
/\ muy_cerca_detras /\ mas_bueno_que_detras /\ posible_adelantado /\
estado_neumaticos[piloto] > MUY_GASTADOS /\
mejor_parada_que_detras /\ estado_neumaticos[piloto]<= USADOS)) ->
(neumatico_next_lap_index <= neumatico_next_lap(3,neumaticos_usados) /\
conduccion_next_lap_index = AGRESIVO);
```

## Restricciones FIA:

- Si no hemos realizado las paradas obligatorias, debemos realizarlas obligatoriamente. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta no puede ser el valor por defecto (DEFAULT).

```
constraint parada_obligatoria -> neumatico_next_lap_index != DEFAULT;
```



- Debemos usar al menos dos tipos de compuestos de neumáticos. Esto se traduce en la restricción de que el índice del tipo de neumático para la siguiente vuelta debe estar dentro de los neumáticos utilizables según los neumáticos usados.

```
constraint neumatico_next_lap_index in neumatico_usable(neumaticos_usados);
```

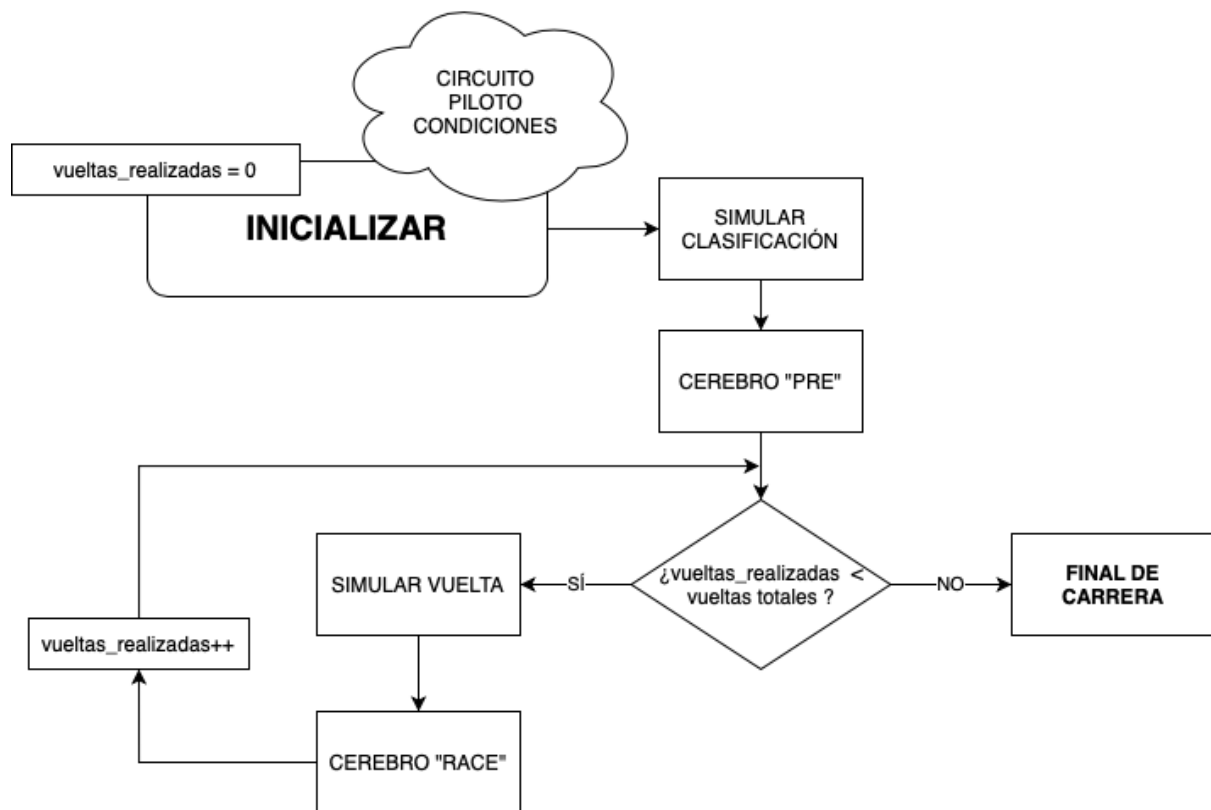
### **Objetivo:**

- El objetivo de la optimización es maximizar la variable "objetivo", que representa la estrategia de carrera para tratar de no perder posiciones y ganarlas si es posible.

```
solve maximize objetivo;
```

## 8. Simulador

Para conocer más en profundidad y mejorar el funcionamiento de los modelos, hemos creado un simulador de carrera. Este instrumento busca simular las circunstancias y desafíos que pueden surgir durante una carrera real de Fórmula 1, aunque es importante reconocer que dada la imprevisibilidad inherente a este deporte, nuestro simulador no puede garantizar una precisión absoluta.



Flujograma del simulador de carrera.

Iniciamos nuestra simulación estableciendo los parámetros iniciales, es decir, inicializando las variables que van a guiar nuestra simulación. Este proceso es fundamental ya que establece las condiciones iniciales que determinarán el desarrollo de nuestra carrera simulada. Hemos decidido valernos de bases de datos reales de carreras de la Fórmula 1 para establecer estas variables, ya que esto nos permitirá cruzarlo con datos reales de estrategias de la fórmula 1 de forma que podamos conseguir la máxima precisión a nuestro alcance.

Hemos escogido centrarnos en el circuito de Azerbaiyán, Bakú. Nuestra elección se basa en la disponibilidad de una gran cantidad de datos históricos de este circuito que hemos conseguido para nuestro análisis. Las variables que se han inicializado para este escenario incluyen aspectos como los detalles del circuito, los pilotos, las escuderías e incluso variables ambientales como el clima.

Una vez que las variables están listas, el simulador se adentra en la etapa de clasificación. Durante esta fase, el simulador considera varios aspectos, como la destreza y habilidades de cada piloto, las condiciones climáticas en ese momento y el historial de rendimiento de las escuderías en el circuito de Azerbaiyán. Esta información nos permite obtener una grilla de salida para la carrera simulada, que si bien no es perfecta, es una representación razonable de lo que podría suceder en una carrera real.

Después de la fase de clasificación, el "cerebro pre" entra en acción. Este sistema genera las estrategias de pre-carrera. Esta etapa nos proporciona información vital como los neumáticos que deberían usar los pilotos en la primera vuelta y la agresividad que deberían aplicar durante la conducción.

La carrera comienza con una simulación de cada vuelta. Aquí, el simulador intenta recrear varios factores como la degradación de los neumáticos, posibles accidentes y cambios en las condiciones climáticas. Estos elementos nos permiten predecir los tiempos de vuelta de cada piloto, que luego se suman a sus tiempos totales para obtener un ranking actualizado.

Este ranking, junto con otra información relevante, se envía al "cerebro race", que genera nuevas estrategias para la próxima vuelta.

Es importante mencionar que el simulador fue creado en Python, un lenguaje de programación accesible y ampliamente utilizado en la comunidad científica, mientras que los "cerebros" se han construido en Minizinc. La simulación se ejecuta de manera cíclica hasta que se concluye la carrera simulada.

Además, cada vez que se activa uno de los "cerebros", se genera un gráfico. Este nos brinda una representación visual de las decisiones tomadas durante la simulación, ayudándonos a entender y analizar los patrones que emergen.

## 9. Conclusiones

Tras numerosas iteraciones y pruebas con nuestro simulador, equipado con los "cerebros pre" y "race" que hemos desarrollado e implementado, hemos llegado a una serie de conclusiones importantes y notables. La más destacada es la observación de que las gráficas que surgen de nuestras simulaciones, es decir, las estrategias de carrera generadas por nuestros algoritmos, muestran similitudes notables con las estrategias que actualmente se utilizan en el mundo real de la Fórmula 1. Este descubrimiento es significativo, ya que sugiere que nuestros modelos de programación por restricciones están capturando aspectos fundamentales de la estrategia de carrera de la Fórmula 1.

Sin embargo, aunque estos resultados son alentadores, debemos reconocer con humildad que nuestro sistema todavía tiene un gran margen de mejora. Los algoritmos y los modelos utilizados tienen limitaciones y hay aspectos de las carreras de Fórmula 1 que nuestro simulador aún no puede capturar con precisión. Algunos de estos aspectos incluyen la

adaptabilidad y la toma de decisiones en tiempo real de los equipos y pilotos humanos, la influencia de los factores psicológicos en el rendimiento de los pilotos y las sutilezas de la interacción entre los equipos, los pilotos y los coches durante la carrera.

A pesar de estos retos y limitaciones, creemos firmemente en el valor de nuestro proyecto. Consideramos que nuestro sistema no está destinado a reemplazar a los seres humanos en la tarea de la estrategia de carrera de la Fórmula 1. En cambio, vemos nuestro simulador y los "cerebros" que hemos desarrollado como herramientas valiosas que pueden apoyar y mejorar la toma de decisiones de los equipos y los pilotos.

La toma de decisiones durante una carrera de Fórmula 1 puede ser increíblemente compleja y abrumadora. Los equipos y los pilotos deben procesar y reaccionar a una gran cantidad de información en tiempo real, tomar decisiones bajo una presión intensa y adaptarse constantemente a las circunstancias cambiantes de la carrera. En este entorno, la existencia de una herramienta que pueda proporcionar análisis de datos en tiempo real y sugerencias de estrategia basadas en modelos de programación por restricciones podría ser muy valiosa.

Nuestro sistema puede proporcionar un punto de vista más calculado y objetivo, que complementa la intuición y la experiencia de los equipos y pilotos humanos. Este punto de vista cuantitativo puede ser especialmente útil en situaciones en las que la presión del tiempo y el estrés pueden dificultar la toma de decisiones objetiva. Por ejemplo, nuestro sistema podría ayudar a los equipos a tomar decisiones más fundamentadas sobre cuándo hacer pit stops, qué tipo de neumáticos usar en función de las condiciones de la pista y el clima, y cómo ajustar la estrategia de carrera en función de los incidentes durante la carrera.

Además, nuestro sistema tiene el potencial de ser útil para la formación y el entrenamiento de los equipos y los pilotos. Al proporcionar un entorno simulado en el que se pueden probar y explorar diferentes estrategias de carrera, nuestro sistema podría ayudar a los equipos y los pilotos a adquirir una comprensión más profunda de las dinámicas de la carrera y a desarrollar mejores intuiciones estratégicas.

En conclusión, aunque reconocemos que nuestro proyecto todavía tiene margen para mejorar, estamos orgullosos de lo que hemos logrado hasta ahora y estamos emocionados por el potencial que nuestro simulador y los "cerebros" asociados tienen para contribuir al mundo de la Fórmula 1 y más allá.

## 10. Líneas futuras

Como bien se señaló en la sección de conclusiones, no dejamos de lado el reconocimiento de que nuestro sistema, aunque ya ha demostrado tener un considerable potencial como facilitador en la toma de decisiones, aún no está a la altura de enfrentar de manera directa y plena los desafíos que una carrera real de Fórmula 1 puede presentar. Esto no nos desalienta, al contrario, consideramos que la programación por restricciones y su aplicación en el contexto de la Fórmula 1 es un campo de estudio en expansión, por lo que estamos entusiasmados y comprometidos con las oportunidades para optimizar y evolucionar aún más nuestro sistema.

Entre las posibles líneas de investigación y desarrollo que hemos identificado, y que podrían contribuir significativamente a que nuestro simulador y los "cerebros" asociados lleguen a ser una herramienta altamente beneficiosa y relevante para las escuderías reales de la Fórmula 1, destacamos las siguientes:

1. **Ampliación de restricciones incorporadas:** La programación por restricciones se distingue por su habilidad para modelar y resolver problemas que comprenden una gran diversidad de restricciones y objetivos. En el caso de nuestra simulación, esto podría traducirse en la incorporación de restricciones relativas a los recursos a disposición de los equipos (como la cantidad limitada de neumáticos), las capacidades físicas y técnicas de los coches y los pilotos, así como las reglas y regulaciones de la Fórmula 1. Mediante la adición y ajuste de estas restricciones en nuestros modelos, se espera poder incrementar la precisión y el realismo de las simulaciones.
2. **Integración de banderas y Safety-Car:** No cabe duda de que las carreras de Fórmula 1 están repletas de eventos inesperados, como accidentes y condiciones climáticas adversas, que pueden modificar drásticamente las dinámicas de una carrera. Estos incidentes suelen llevar a la implementación de banderas de distintos colores, y en casos específicos, al despliegue del Safety-Car. Como parte del plan para incrementar el realismo y la precisión de nuestras simulaciones, se busca incluir estos eventos y su impacto en la estrategia de carrera en los modelos.
3. **Refinamiento de los perfiles de los circuitos:** Sabemos que cada circuito de la Fórmula 1 posee características propias y únicas, que van desde la longitud y el diseño de la pista, hasta las condiciones climáticas típicas. Con esto se mejoraría la precisión del simulador al desarrollar perfiles más detallados y precisos de cada circuito, utilizando para ello datos históricos y actuales.
4. **Optimización de la función de degradación:** La degradación de los neumáticos es un factor crítico en las carreras de Fórmula 1, ya que tiene impacto tanto en el rendimiento del coche como en la estrategia de carrera. Es por ello que se propone mejorar nuestra función de degradación, con el fin de reflejar de manera más exacta las complejidades de este fenómeno, considerando factores como el tipo y la edad de los neumáticos, las condiciones de la pista y el estilo de conducción del piloto.
5. **Incorporación de rivalidades:** Las rivalidades entre pilotos y equipos pueden tener un impacto considerable en las dinámicas de la carrera. Esto podría añadir una

dimensión adicional a las simulaciones y a las estrategias de carrera generadas por el sistema.

6. **Consideración del campeonato mundial y los constructores:** Las carreras de Fórmula 1 no suceden en un vacío, sino que forman parte de una serie de campeonatos que se llevan a cabo a lo largo de la temporada. Los resultados de cada carrera tienen implicaciones tanto para los campeonatos de pilotos como para los constructores, y estos contextos más amplios pueden tener influencia en las estrategias de carrera. El objetivo sería que las simulaciones y estrategias generadas se ajusten aún más a la realidad de las competiciones.

En conclusión, a pesar de los avances que hemos logrado hasta ahora, estamos expectantes y emocionados por las oportunidades que estas líneas de investigación y desarrollo futuras nos ofrecen. Estamos convencidos de que al continuar por estos caminos el sistema se podría convertir en algo real y a considerar por las escuderías presentes en la Fórmula 1.