



EXAMEN MIAR

Autor: Guillermo Fora

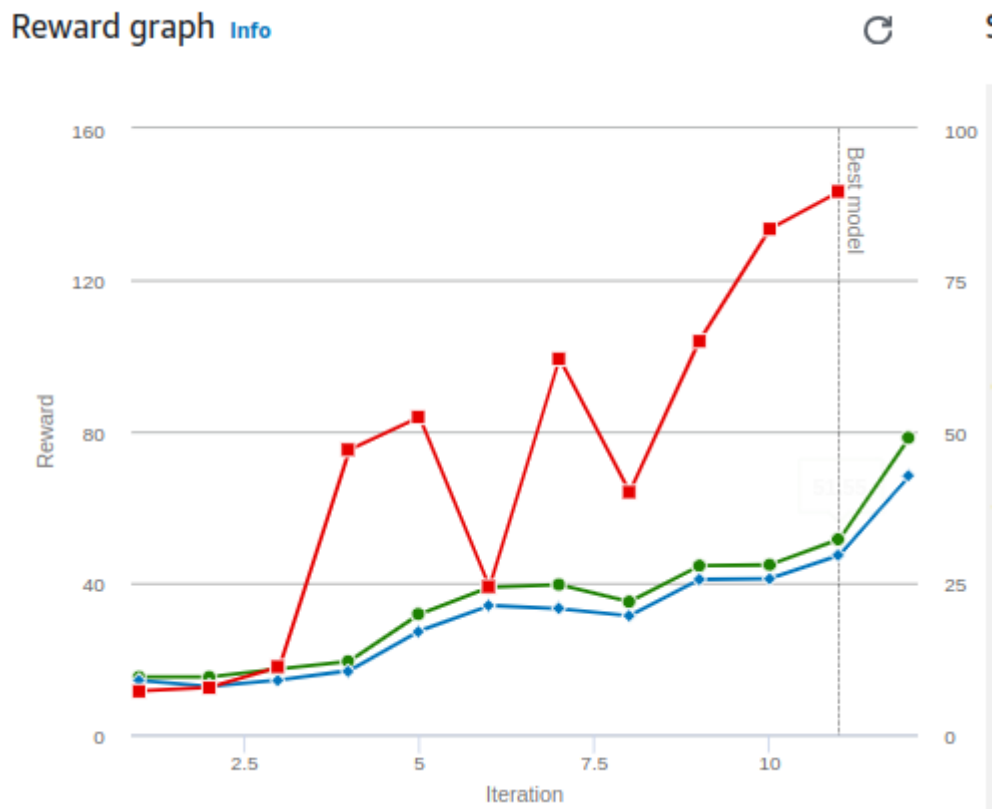
<b>Modelo: Guillermo1</b>	<b>3</b>
Función de recompensa	3
Gráficos de recompensa	3
Evaluación	4
Primera evaluación	4
Segunda evaluación	4
Conclusión	4
<b>Modelo: Guillermo2</b>	<b>5</b>
Función de recompensa	5
Gráficos de recompensa	6
Evaluación	7
Primera evaluación	7
Segunda evaluación	7
Tercera evaluación	8
Conclusión	8
<b>Modelo: Guillermo1-clon</b>	<b>9</b>
Función de recompensa	9
Gráficos de recompensa	10
Evaluación	11
Primera evaluación	11
Segunda evaluación	11
Conclusión	11
<b>Modelo: Guillermo2-clon</b>	<b>12</b>
Función de recompensa	12
Gráficos de recompensa	14
Evaluación	14
Primera evaluación	14
Segunda evaluación	15
Conclusión	15
<b>Tiempos de la carrera MIAR</b>	<b>16</b>

# Modelo: Guillermo1

## Función de recompensa

En este modelo utilizaremos la función de recompensa por defecto.

## Gráficos de recompensa



# Evaluación

## Primera evaluación

Circuito de entrenamiento

Trial	Time (MM:SS.mm m)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:13.792	100%	Lap complete	0	--	0	--
2	00:14.341	100%	Lap complete	0	--	0	--
3	00:19.755	100%	Lap complete	2	4 seconds	0	--

## Segunda evaluación

Circuito: Ace Speedway

Trial	Time (MM:SS.mm m)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:37.925	100%	Lap complete	3	6 seconds	0	--
2	00:35.003	100%	Lap complete	2	4 seconds	0	--
3	00:33.136	100%	Lap complete	1	2 seconds	0	--

## Conclusión

En la primera evaluación podemos ver que el coche solo se sale 2 veces en la última vuelta pero la primera vuelta la hace muy rápido (en 13,7 seg). Es un modelo que está bien entrenado en ese circuito pero si lo llevamos a otro circuito vemos que ya se sale varias veces, en la primera vuelta se sale 3 veces, en la segunda 2 y en la última 1 vez.

# Modelo: Guillermo2

## Función de recompensa

```
import math
```

```
def reward_function(params):
```

```
    """
```

```
    Example of rewarding the agent to follow center line
```

```
    """
```

```
    # Read input parameters
```

```
    track_width = params['track_width']
```

```
    distance_from_center = params['distance_from_center']
```

```
    progress = params['progress'] # Progreso en la pista
```

```
    all_wheels_on_track = params['all_wheels_on_track'] # Si todas las ruedas están la pista
```

```
    is_offtrack = params['is_offtrack'] # Si se sale de la pista
```

```
    speed = params['speed'] # Parametros de velocidad
```

```
    steering_angle = abs(params['steering_angle']) # Usamos el valor absoluto para evitar
```

```
    direcciones negativas
```

```
    # Calculate 3 markers that are at varying distances away from the center line
```

```
    marker_1 = 0.1 * track_width
```

```
    marker_2 = 0.25 * track_width
```

```
    marker_3 = 0.5 * track_width
```

```
    # Give higher reward if the car is closer to center line and vice versa
```

```
    if distance_from_center <= marker_1:
```

```
        reward = 1.0
```

```
    elif distance_from_center <= marker_2:
```

```
        reward = 0.5
```

```
    elif distance_from_center <= marker_3:
```

```
        reward = 0.1
```

```
    else:
```

```
        reward = 1e-3 # likely crashed/ close to off track
```

```
    # Recompensa por mantener todas las ruedas dentro de la pista
```

```
    if all_wheels_on_track:
```

```
        reward += 5 # Si todas las ruedas están en la pista, se suman 5 puntos
```

```
    else:
```

```
        reward -= 5 # Si alguna rueda está fuera de la pista, se penalizará con 5 puntos
```

```
    if is_offtrack==False: # Si se sale de la pista le penalizamos
```

```
        reward *= 1e-3
```

```
    else:
```

```

reward += 4 # Si el no se sale de la pista recibirá 4 puntos

# Recompensar por velocidad alta sin perder el control
max_speed=1.8 #Le asignamos como velocidad máxima 1.8
if speed < max_speed:
    reward += 2 # Le recompensamos con 2 punto si la velocidad es menor a la velocidad
máxima

# Queremos penalizar el zigzaguo, es decir, si el ángulo de dirección es muy alto
if steering_angle > 15:
    reward -= 2 # Le penalizamos con 2 puntos si el ángulo de direccion es más alto de 15

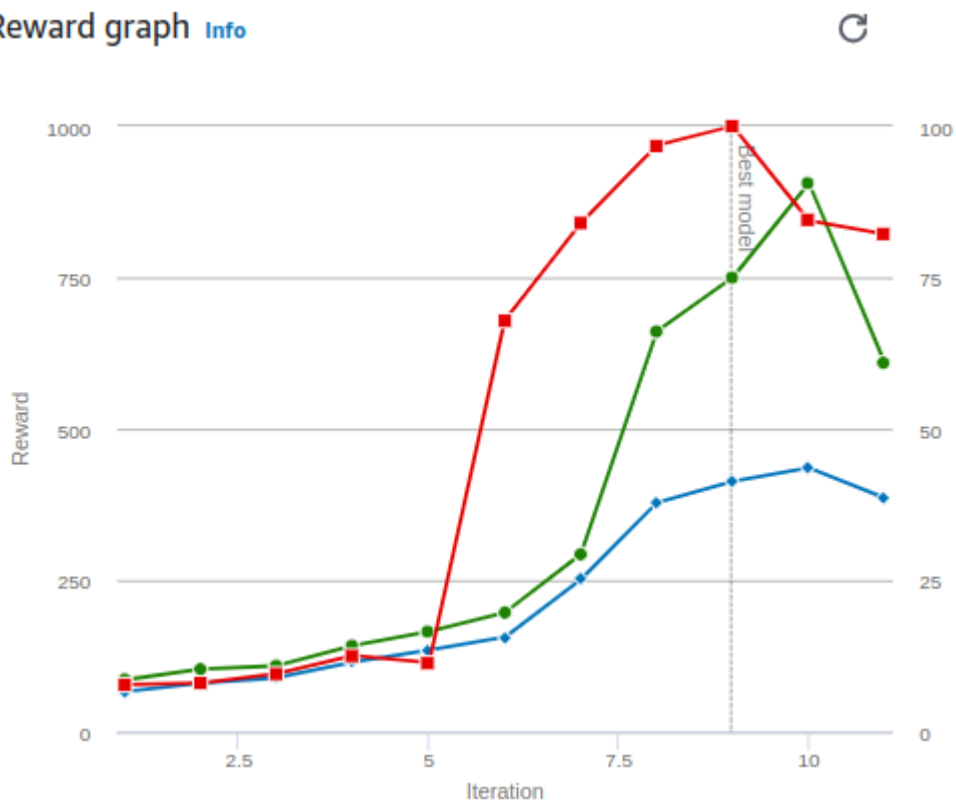
# Si ha superado el 85% de la pista recibirá 50 puntos
if progress >= 85:
    reward+=50

return float(reward)

```

## Gráficos de recompensa

Reward graph [Info](#)



# Evaluación

## Primera evaluación

Circuito: mismo entrenamiento

<b>Trial</b>	<b>Time (MM:SS.mm m)</b>	<b>Trial results (% track completed)</b>	<b>Status</b>	<b>Off-track ck</b>	<b>Off-track penalty</b>	<b>Crash es</b>	<b>Crash penalty</b>
1	00:14.772	100%	Lap complete	0	--	0	--
2	00:14.116	100%	Lap complete	0	--	0	--
3	00:14.120	100%	Lap complete	0	--	0	--

## Segunda evaluación

Circuito : Forever Raceway

<b>Trial</b>	<b>Time (MM:SS.mm m)</b>	<b>Trial results (% track completed)</b>	<b>Status</b>	<b>Off-track ck</b>	<b>Off-track penalty</b>	<b>Crash es</b>	<b>Crash penalty</b>
1	00:17.935	100%	Lap complete	1	2 seconds	0	--
2	00:18.741	100%	Lap complete	1	2 seconds	0	--
3	00:21.532	100%	Lap complete	2	4 seconds	0	--

## Tercera evaluación

circuito: Forever raceway

Trial	Time (MM:SS.mm m)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:17.088	100%	Lap complete	1	2 seconds	0	--
2	00:23.861	100%	Lap complete	3	6 seconds	0	--
3	00:17.655	100%	Lap complete	1	2 seconds	0	--

## Conclusión

En la primera evaluación (en el mismo circuito de entrenamiento) podemos observar que lo ha hecho muy bien y no se ha salido ninguna vez, que es muy bueno.

He cometido el error de poner mal la función de recompensa, ya que quería recompensarle, si la velocidad era mayor a la velocidad máxima pero he puesto el signo al revés. Esto sería algo a cambiar en otro modelo para ver si mejoraría.

Debido a ese error no ha hecho mejor tiempo que el primer modelo pero se ha quedado muy cerquita de superarle en la mejor vuelta.

En la segunda evaluación la he realizado en otro circuito y podemos ver que se ha salido 1 vez en la primera vuelta, 1 en la segunda y 2 en la tercera por lo que podría adaptarse a otras pistas.

Por curiosidad, he realizado una tercera evaluación en el mismo circuito que la segunda evaluación y vemos que esta vez se ha salido 1 veces en la primera vuelta, 3 en la segunda y 1 en la última. Lo bueno de esta prueba es que ha superado su mejor vuelta respecto a la segunda evaluación y la tercera vuelta la ha hecho más rápido que cualquiera de la segunda evaluación, puede que si hago una cuarta evaluación saque peores tiempos pero parece ser un buen modelo para utilizar en otros circuitos.



# Modelo: Guillermo1-clon

En esta función de recompensa hemos intentado aplicar los waypoints para que el coche vaya lo más cerca del centro de la pista , para ello necesitamos calcular el punto previo y el próximo punto, luego calculamos la dirección del siguiente segmento de la pista, convertimos esa dirección de grados a radianes y calculamos la diferencia entre la dirección de la pista y la dirección actual del coche.

## Función de recompensa

```
import math
def reward_function(params):
    """
    Example of rewarding the agent to follow center line

    """

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    waypoints = params['waypoints']
    closest_waypoints=params['closest_waypoints']
    heading = params['heading']

    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.0
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-3 # likely crashed/ close to off track

    # Vamos a intentar que siga la línea central
    next_waypoint = params['waypoints'][closest_waypoints[1]]
    prev_waypoint = params['waypoints'][closest_waypoints[0]]

    # calculamos la dirección del siguiente segmento de la pista
    track_direction = math.atan2(next_waypoint[1] - prev_waypoint[1], next_waypoint[0] -
prev_waypoint[0])

    # Convierte la dirección de grados a radianes
```

```
track_direction = math.degrees(track_direction)
```

```
# Calculamos la diferencia entre la dirección de la pista y la dirección actual del coche
```

```
direction_diff = abs(track_direction - heading)
```

```
if direction_diff > 180:
```

```
    direction_diff = 360 - direction_diff
```

```
# Penalizamos si el ángulo de dirección es muy grande
```

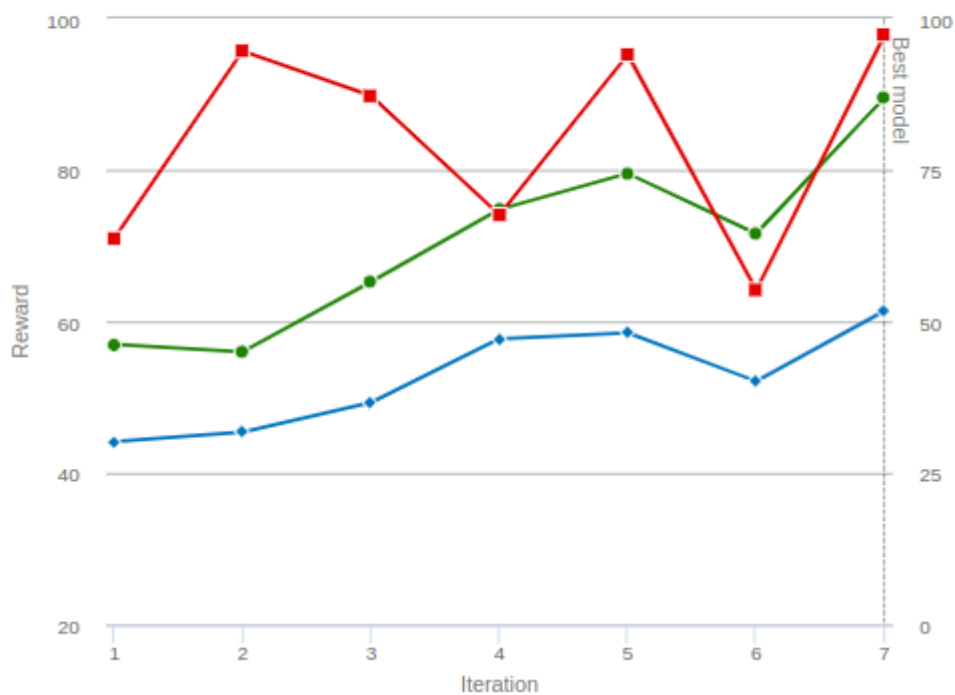
```
if direction_diff > 20:
```

```
    reward *= 0.9
```

```
return float(reward)
```

## Gráficos de recompensa

Reward graph [Info](#)



# Evaluación

## Primera evaluación

Trial	Time (MM:SS.mm)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:16.344	100%	Lap complete	1	2 seconds	0	--
2	00:17.609	100%	Lap complete	1	2 seconds	0	--
3	00:14.669	100%	Lap complete	0	--	0	--

## Segunda evaluación

Circuito: Forever Raceway

Trial	Time (MM:SS.mm)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:20.867	100%	Lap complete	2	4 seconds	0	--
2	00:25.261	100%	Lap complete	4	8 seconds	0	--
3	00:24.937	100%	Lap complete	4	8 seconds	0	--

## Conclusión

En la primera evaluación podemos ver que se ha salido 1 vez en la primera vuelta, otra en la segunda y ninguna en la tercera vuelta. Su tiempo no ha sido el mejor de todos los modelos pero podría ser aceptable. Viendo el video de la evaluación puedo decir que sí que ha cumplido bastante bien el objetivo que era que fuera por el centro de la pista.

En cambio, en la segunda evaluación vemos que se ha salido 2 veces en la segunda vuelta, 4 en la segunda y 4 en la última vuelta, pero viendo el video no sigue mucho el centro de la pista por lo que no sería un buen modelo para utilizar en otros circuitos.

Para mejorar este modelo pondría más recompensas en varios aspectos, ya que solo nos hemos centrado en que vaya por el centro de la pista y no le hemos recompensado ni penalizado por otras cosas. Sería un buen punto a mejorar en el siguiente modelo.

## Modelo: Guillermo2-clon

En este modelo vamos a mejorar la recompensa de la velocidad, para que cuando sea mayor que la velocidad máxima le de puntuación y he querido agregarle que a medida que vaya aumentando en steps, si la velocidad es mayor a la velocidad máxima recibirá más recompensa. Según mi deducción, al salirse de la pista o al reiniciar la vuelta, se reinician los steps, por lo que si no se sale de la pista puede seguir aumentando la recompensa que gana por superar la velocidad máxima.

### Función de recompensa

```
import math
```

```
def reward_function(params):
```

```
    """
```

```
    Example of rewarding the agent to follow center line
```

```
    """
```

```
    # Read input parameters
```

```
    track_width = params['track_width']
```

```
    distance_from_center = params['distance_from_center']
```

```
    progress = params['progress'] # Progreso en la pista
```

```
    all_wheels_on_track = params['all_wheels_on_track'] # Si todas las ruedas están la pista
```

```
    is_offtrack = params['is_offtrack'] # Si se sale de la pista
```

```
    speed = params['speed'] # Parametros de velocidad
```

```
    steering_angle = abs(params['steering_angle']) # Usamos el valor absoluto para evitar
```

```
    direcciones negativas
```

```
    steps=params['steps']
```

```
    # Calculate 3 markers that are at varying distances away from the center line
```

```
    marker_1 = 0.1 * track_width
```

```
    marker_2 = 0.25 * track_width
```

```
    marker_3 = 0.5 * track_width
```

```
    # Give higher reward if the car is closer to center line and vice versa
```

```
    if distance_from_center <= marker_1:
```

```
        reward = 1.0
```

```
    elif distance_from_center <= marker_2:
```

```
        reward = 0.5
```

```
    elif distance_from_center <= marker_3:
```

```
        reward = 0.1
```

```

else:
    reward = 1e-3 # likely crashed/ close to off track

# Recompensa por mantener las ruedas dentro de la pista
if all_wheels_on_track:
    reward += 5 # Si todas las ruedas están en la pista, se suman 5 puntos
else:
    reward -= 5 # Penalización si alguna rueda está fuera de la pista

if is_offtrack==False: #Si se sale de la pista le penalizamos
    reward *= 1e-3
else:
    reward += 4 # Si el no se sale de la pista recibirá 4 punto

# Penalizar zigzagado, es decir, si el ángulo de dirección es muy alto
if steering_angle > 15:
    reward -= 2 # Penalización por zigzagado

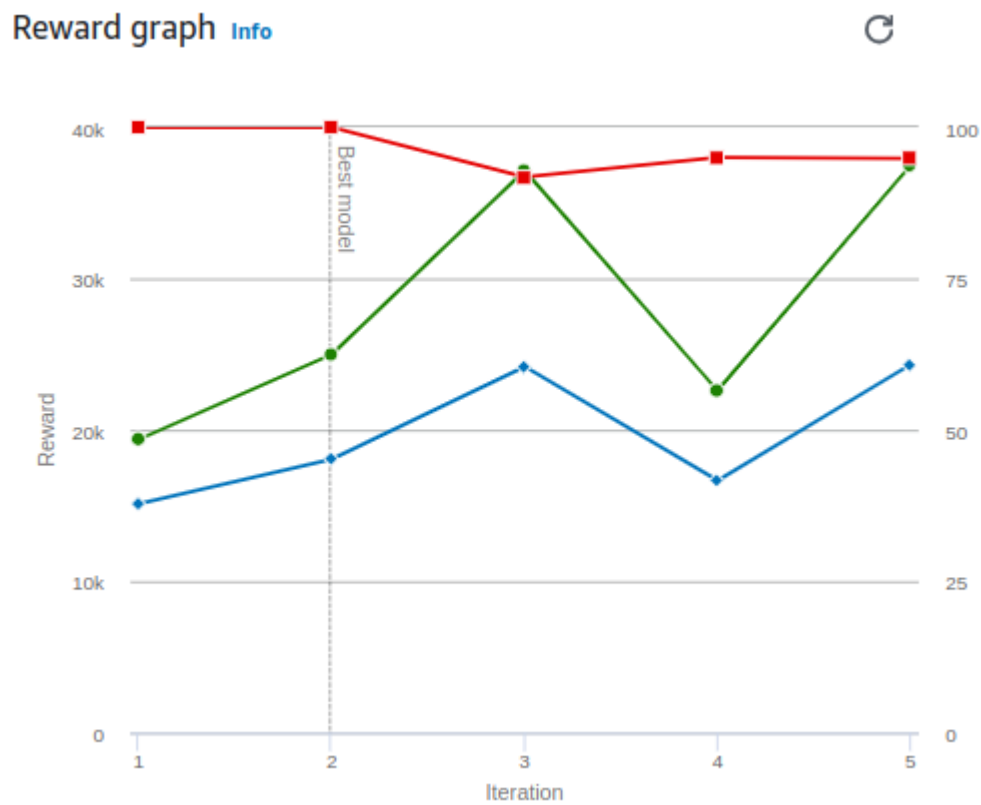
# Recompensar por velocidad alta sin perder el control
max_speed=1.8
#Bucle para que cuantos más steps consiga con la velocidad máxima recibe recompensa
for step in range (5,100):
    if steps > step:
        if speed > max_speed:
            reward += 10 # Recompensa por encima de la velocidad máxima

# Si ha superado el 85% de la pista recibirá 50 puntos
if progress >= 85:
    reward+=50

return float(reward)

```

# Gráficos de recompensa



## Evaluación

### Primera evaluación

Circuito de entrenamiento

Trial	Time (MM:SS.mm m)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:14.461	100%	Lap complete	0	--	0	--
2	00:14.154	100%	Lap complete	0	--	0	--
3	00:15.156	100%	Lap complete	0	--	0	--

## Segunda evaluación

Circuito: Forever raceway

Trial	Time (MM:SS.mm m)	Trial results (% track completed)	Status	Off-track	Off-track penalty	Crashes	Crash penalty
1	00:21.567	100%	Lap complete	2	4 seconds	0	--
2	00:17.946	100%	Lap complete	1	2 seconds	0	--
3	00:17.465	100%	Lap complete	1	2 seconds	0	--

## Conclusión

En la primera evaluación podemos ver que no se ha salido ninguna vez del circuito y si parece que ha mejorado respecto al primer modelo pero el mejor tiempo lo sigue teniendo el primer modelo, puede que realizando más evaluaciones saque mejores tiempos.

En la segunda evaluación, se ha salido 2 veces en la primera vuelta, 1 en la segunda y otra en la última. Si comparamos los tiempos con su primer modelo, si que parece que haya mejorado algo pero es por centésimas de segundo. Vemos que en la segunda evaluación si que ha hecho mejor tiempo que "Padre".

Respecto a todos los modelos, este parece el que mejor se adapta a otros circuitos.

Para mejorarlo podríamos subir la velocidad máxima a ver si hace mejores tiempos que los otros modelos y podemos penalizar más el zigzaguo y darle alguna recompensa más

# Tiempos de la carrera MIAR

Your submissions (9)						
<div><div>Q</div><div>Find by model name</div></div>				<div>&lt; 1 &gt; ⚙</div>		
Model name	Qualifying time	Best lap time	Average lap time	Status	Date submitted to race	Video
<a href="#">Guillermo-2</a>	00:40.328	00:12.797	00:13.442	✔ Submitted	12/12/2024, 8:44 PM	<a href="#">View</a>
<a href="#">Guillermo2-clone</a>	00:41.526	00:13.599	00:13.842	✔ Submitted	12/12/2024, 8:32 PM	<a href="#">View</a>
<a href="#">Guillermo-2</a>	00:39.728	00:12.734	00:13.242	✔ Submitted	12/12/2024, 8:21 PM	<a href="#">View</a>
<a href="#">Guillermo2</a>	00:40.921	00:13.330	00:13.640	✔ Submitted	12/12/2024, 8:06 PM	<a href="#">View</a>
<a href="#">Guillermo1-clon</a>	00:48.989	00:14.798	00:16.329	✔ Submitted	12/12/2024, 7:49 PM	<a href="#">View</a>
<a href="#">Guillermo1</a>	00:54.591	00:17.069	00:18.197	✔ Submitted	12/12/2024, 7:35 PM	<a href="#">View</a>
<a href="#">Guillermo1</a>	00:57.992	00:18.261	00:19.330	✔ Submitted	12/12/2024, 7:24 PM	<a href="#">View</a>
<a href="#">Guillermo2</a>	00:40.799	00:13.532	00:13.599	✔ Submitted	12/12/2024, 7:07 PM	<a href="#">View</a>
<a href="#">Guillermo1</a>	00:48.186	00:14.662	00:16.062	✔ Submitted	12/12/2024, 6:54 PM	<a href="#">View</a>