





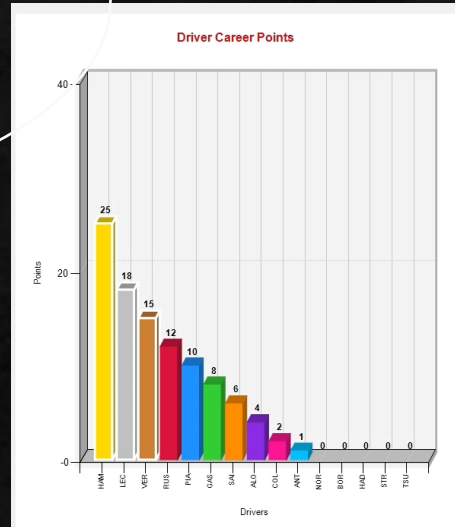
# **FÓRMULA 0 – FBD**

P3G9  
Inês Batista, 124877  
Maria Quinteiro, 124996



# INTRODUÇÃO

- Desenvolvimento de um sistema de base de dados para a gestão de temporadas da Fórmula1;
- Entidades: Circuito, Temporada, Equipa, Membros da Equipa, Staff, Pilotos, Contrato, Grande\_Prémio, Sessões, Resultados, Penalizações e Pitstops;
- Interação por meios de formulários e apresentação de estatísticas;



Add New Staff Member

**Add New Staff Member**

Username:

Password:

Full Name:

Role:

Nationality:

Birth Date:

Team:

**Contract Information (Required)**

Start Year:

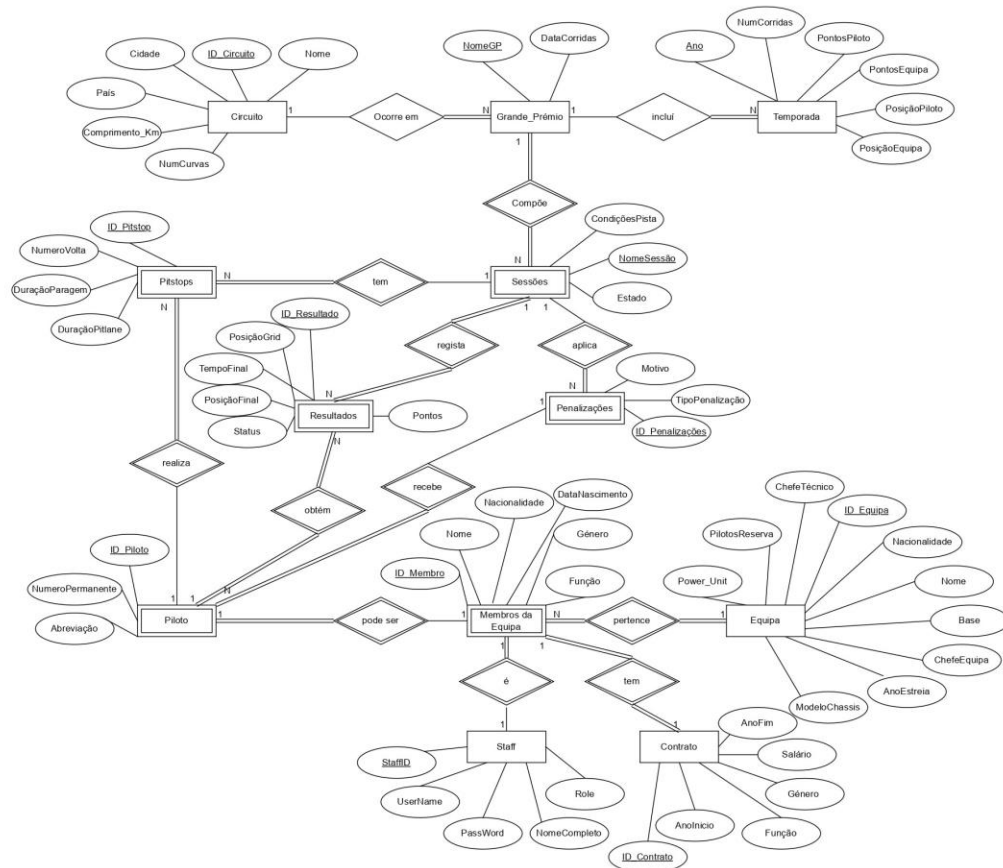
End Year:

Function:

Salary:

Gender:

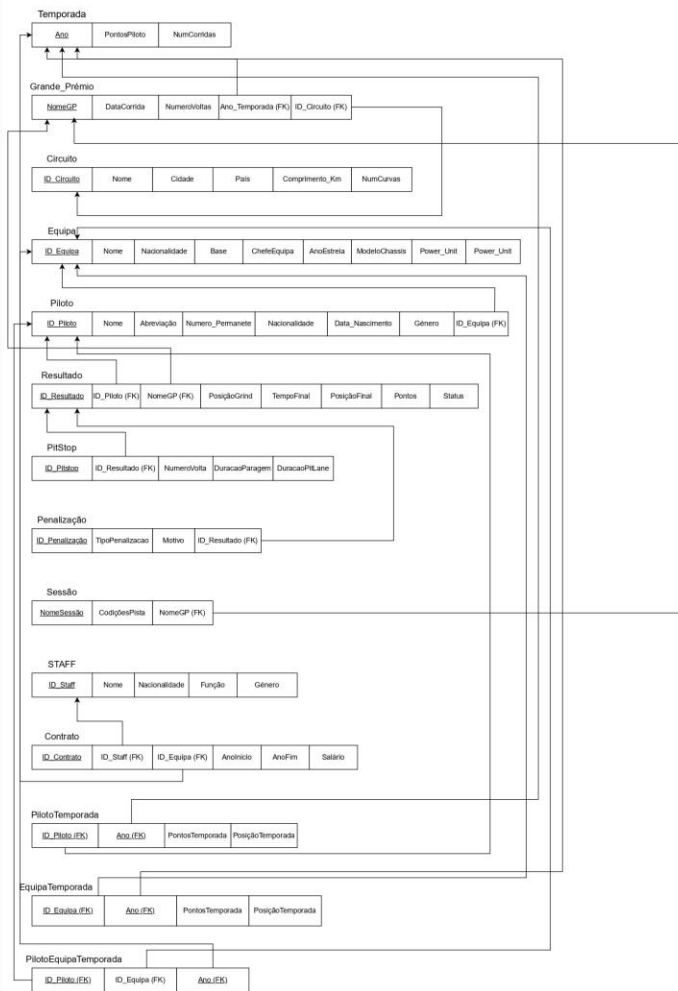
# DIAG. ENTIDADE-RELAÇIONAMETO



# ESQUEMA RELACIONAL

A normalização do modelo foi realizada até à BCNF.

Os atributos que dependiam simultaneamente de uma entidade e da Temporada foram removidos e colocados em tabelas associativas, garantindo que todos os determinantes são super-chaves.



# STORE PROCEDURES

Add Result - Qualification

Driver: #67 VET - Sebastian Vettel (Dusty)

Final Position: 1

Final Time: 00.01 Format: 01:25:45.123  
(Hours:Minutes:Seconds.Milliseconds)

Status: Finished

Points: 0 Auto-calculated based on position

Add Result Cancel

```
5 IF OBJECT_ID('sp_GetDriverStandingsBySeason', 'P') IS NOT NULL
6     DROP PROCEDURE sp_GetDriverStandingsBySeason;
7 GO
8
9 CREATE PROCEDURE sp_GetDriverStandingsBySeason
10     @Season INT
11 AS
12 BEGIN
13     SET NOCOUNT ON;
14
15     SELECT
16         ROW_NUMBER() OVER (ORDER BY ISNULL(SUM(r.Pontos), 0) DESC,
17                             COUNT(CASE WHEN r.PosicaoFinal = 1 THEN 1 END) DESC) AS Position,
18         ISNULL(m.Nome, 'Unknown Driver') AS Driver,
19         ISNULL(e.Nome, 'No Team') AS Team,
20         ISNULL(SUM(r.Pontos), 0) AS TotalPoints,
21         COUNT(CASE WHEN r.PosicaoFinal = 1 THEN 1 END) AS Wins,
22         COUNT(CASE WHEN r.PosicaoFinal <= 3 THEN 1 END) AS Podiums
23     FROM Piloto p
24     INNER JOIN Membros_da_Equipa m ON p.ID_Membro = m.ID_Membro
25     INNER JOIN Equipa e ON p.ID_Equipa = e.ID_Equipa
26     INNER JOIN Resultados r ON p.ID_Piloto = r.ID_Piloto
27     INNER JOIN Grande_Prêmio gp ON r.NomeGP = gp.NomeGP
28     WHERE r.NomeSessão = 'Race' AND gp.Ano_Temporada = @Season
29     GROUP BY p.ID_Piloto, m.Nome, e.Nome
30     HAVING ISNULL(SUM(r.Pontos), 0) > 0
31     ORDER BY TotalPoints DESC, Wins DESC;
32
33 END;
```

```
1 CREATE OR ALTER PROCEDURE sp_InsertResult
2     @PosicaoGrid INT,
3     @TempoFinal TIME = NULL,
4     @PosicaoFinal INT,
5     @Status NVARCHAR(50),
6     @Pontos DECIMAL(10,2),
7     @NomeSessao NVARCHAR(100),
8     @NomeGP NVARCHAR(100),
9     @ID_Piloto INT
10 AS
11 BEGIN
12     INSERT INTO Resultados (PosicaoGrid, TempoFinal, PosicaoFinal, Status, Pontos, NomeSessão, NomeGP, ID_Piloto)
13     VALUES (@PosicaoGrid, @TempoFinal, @PosicaoFinal, @Status, @Pontos, @NomeSessao, @NomeGP, @ID_Piloto)
14 END
15 GO
```

Standings - 2025 Season

2025 Season Standings

Driver Standings: Team Standings

Search Driver: Type driver name or team...

Position	Driver	Team	Points
1	Lewis Hamilton	Scuderia Ferrari	25,00
2	Charles Leclerc	Scuderia Ferrari	18,00
3	Max Verstappen	Red Bull Racing	15,00
4	George Russell	Mercedes	12,00
5	Oscar Piastri	McLaren	10,00
6	Pierre Gasly	Alpine	8,00
7	Carlos Sainz	Williams	6,00
8	Fernando Alonso	Aston Martin	4,00
9	Franco Colapinto	Alpine	2,00
10	Kimi Antonelli	Mercedes	1,00

Close

# TRIGGERS

Garante a consistência na base de dados sempre que ocorre uma modificação.


```
2 IF OBJECT_ID('trg_UpdateSeasonRaceCount', 'TR') IS NOT NULL
3     DROP TRIGGER trg_UpdateSeasonRaceCount;
4 GO
5
6 CREATE TRIGGER trg_UpdateSeasonRaceCount
7 ON Grande_Prêmio
8 AFTER INSERT, DELETE
9 AS
10 BEGIN
11     SET NOCOUNT ON;
12
13     UPDATE t
14     SET t.NumCorridas = (
15         SELECT COUNT(*)
16         FROM Grande_Prêmio gp
17         WHERE gp.Ano_Temporada = t.Ano
18     )
19 FROM Temporada t
20 WHERE t.Ano IN (
21     SELECT DISTINCT Ano_Temporada FROM inserted
22     UNION
23     SELECT DISTINCT Ano_Temporada FROM deleted
24 );
25 END;
26 GO
```



# INDEXES

- Utilizados para melhorar significativamente a performance das consultas frequentes.
- Reduz o tempo de resposta e diminui leituras desnecessárias

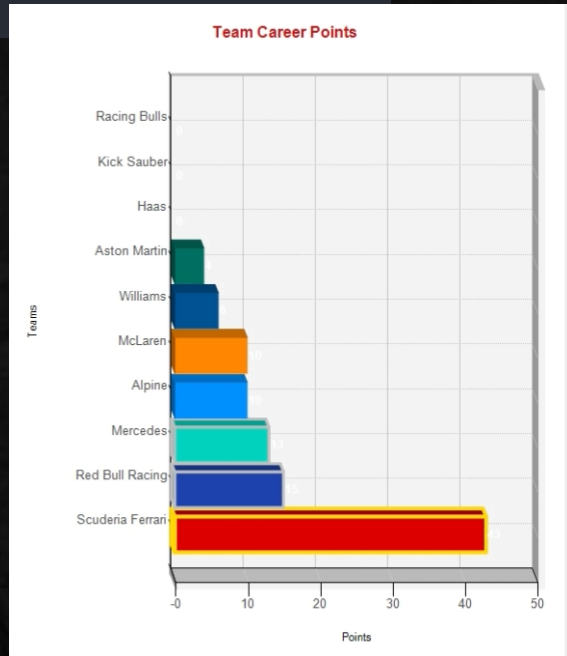
```
12 v IF NOT EXISTS (SELECT * FROM sys.indexes WHERE name = 'IX_Resultados_Session' AND object_id = OBJECT_ID('Resultados'))
13 | CREATE NONCLUSTERED INDEX IX_Resultados_Session ON Resultados(NomeSessão) INCLUDE (Pontos, PosiçãoFinal);
14 GO
```



A tabela Resultados é a base das estatísticas.  
A coluna NomeSessão é usada como filtro frequente.  
O uso de INDICE evita key lookups.

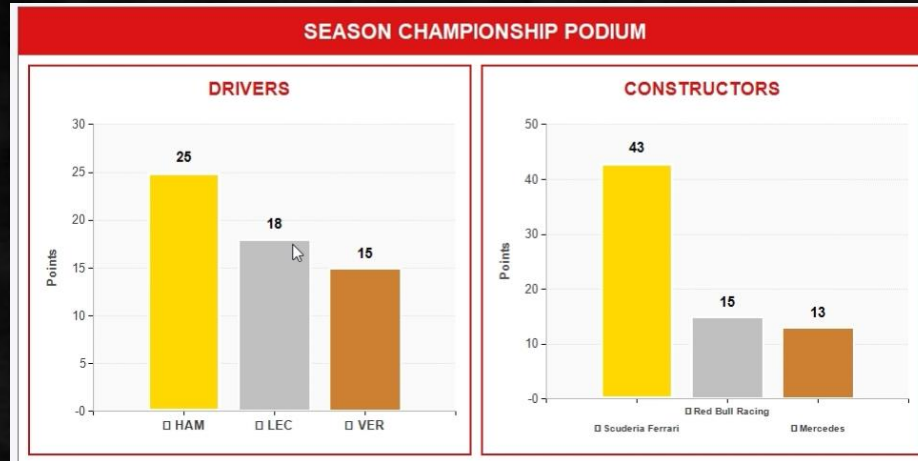
# ESTADÍSTICAS

```
1 CREATE OR ALTER PROCEDURE sp_GetTeamCareerPoints
2 AS
3 BEGIN
4     SELECT * FROM vw_Team_CareerPoints ORDER BY TotalPoints DESC, TeamName ASC;
5 END
```





# ESTATÍSTICAS



```
1 IF OBJECT_ID('vw_Season_DriverPodium', 'V') IS NOT NULL
2     DROP VIEW vw_Season_DriverPodium;
3 GO
4 CREATE VIEW vw_Season_DriverPodium AS
5 SELECT
6     gp.Ano_Temporada AS Year,
7     p.Abreviação,
8     m.Nome AS DriverName,
9     ISNULL(SUM(r.Pontos), 0) AS TotalPoints
10 FROM Grande_Prêmio gp
11 INNER JOIN Resultados r ON r.NomeGP = gp.NomeGP
12 INNER JOIN Piloto p ON r.ID_Piloto = p.ID_Piloto
13 INNER JOIN Membros_da_Equipa m ON p.ID_Membro = m.ID_Membro
14 WHERE r.NomeSessão = 'Race'
15 GROUP BY gp.Ano_Temporada, p.ID_Piloto, p.Abreviação, m.Nome, p.NumeroPermanente;
```

```
1 IF OBJECT_ID('vw_Season_TeamPodium', 'V') IS NOT NULL
2     DROP VIEW vw_Season_TeamPodium;
3 GO
4 CREATE VIEW vw_Season_TeamPodium AS
5 SELECT
6     gp.Ano_Temporada AS Year,
7     e.Nome AS TeamName,
8     ISNULL(SUM(r.Pontos), 0) AS TotalPoints
9 FROM Grande_Prêmio gp
10 INNER JOIN Resultados r ON r.NomeGP = gp.NomeGP
11 INNER JOIN Piloto p ON r.ID_Piloto = p.ID_Piloto
12 INNER JOIN Equipa e ON p.ID_Equipa = e.ID_Equipa
13 WHERE r.NomeSessão = 'Race'
14 GROUP BY gp.Ano_Temporada, e.ID_Equipa, e.Nome;
```



***DEMO***