

Podium

Linguagem de Simulação de Fórmula 1

Isabela Rodrigues

Maio de 2025

- Capturar a dinâmica real de corridas de F1 em uma DSL intuitiva
- Facilitar o ensino de compiladores e simulações esportivas
- Permitir experimentação de estratégias (pit-stop, pneus, combustível)
- Promover análise imediata de resultados em texto estruturado

Características Principais

Sintaxe Clara

- Delimitadores: `greenLight ... chequeredFlag`
- Declarações: `driver, team`, variáveis tipadas

Comandos F1

- `addPoint(piloto,pontos)`
- `pitStop(), pushLap()`
- `radio(mensagem)`
- `min(a,b), max(a,b)`

Controle de Fluxo

- Loop: `forLap(inicio..fim)`
- Condicional: `ifPit(cond) [elsePit(cond)]`

- Inspirada em regulamentos oficiais da FIA
- Ponto de partida para um compilador em Flex/Bison
- Exibe logs de corrida em estilo retransmissão de rádio
- Extensível para cenários (safety car, chuva, estratégia ERS)

Configuração — GP de Mônaco

```
greenLight
// Pilotos e equipes
driver MaxV;
driver ChecoP;
driver LewisH;
driver CharlesL;
team RedBull[MaxV,ChecoP];
team Ferrari[CharlesL,LewisH];
// Circuito
downforce : int = 9;
lapCount  : int = 10;

// Setores
sector1 : lapTime = 18000;
sector2 : lapTime = 36000;
sector3 : lapTime = 24000;
bestLap : lapTime = 99999;
// Estratégia
tyreWear      : int  = 0;
fuelLoad      : int  = 100;
pitstopDone   : bool = false;
overtakingDifficult : bool = true;
tightCorners  : bool = true;
```

Simulação — GP de Mônaco

```
radio("Luzes apagadas no circuito de Monaco!");
forLap(1..lapCount) {
  tyreWear = tyreWear + 4;
  fuelLoad = fuelLoad - 1;
  currentLap = sector1+sector2+sector3;
  bestLap    = min(bestLap,currentLap);
  ifPit(tyreWear>30 && !pitstopDone) {
    radio("Entrada nos boxes...");
    pitStop(); tyreWear=0; pitstopDone=true;
    ifPit(overtakingDifficult) {
      radio("Difícil recuperar...");
    }
  }
}
```

```
ifPit(tyreWear<20 && fuelLoad>30) {
  pushLap(); radio("Push no túnel!");
  sector2 = sector2 - 500;
}
ifPit(currentLap<80000) {
  addPoint(CharlesL,2);
} elsePit {
  addPoint(MaxV,2);
}
}
radio("Fim da corrida...");
chequeredFlag
```

Saída de Execução — Monaco GP

```
=== Executando ../exemplos/monaco_race.pod ===
```

```
[LLVM] Inicializado (simulação)
```

```
=== Início da Execução ===
```

```
Piloto registrado: MaxV
```

```
Piloto registrado: ChecoP
```

```
Piloto registrado: LewisH
```

```
Piloto registrado: CharlesL
```

```
Equipe registrada: RedBull [MaxV, ChecoP]
```

```
Equipe registrada: Ferrari [CharlesL, LewisH]
```

```
Radio: Luzes apagadas no circuito de Monaco!
```

```
--- Volta 1 ---
```

```
--- Volta 2 ---
```

```
--- Volta 3 ---
```

```
--- Volta 4 ---
```

```
--- Volta 5 ---
```

```
--- Volta 6 ---
```

```
--- Volta 7 ---
```

```
--- Volta 8 ---
```

```
--- Volta 9 ---
```

```
--- Volta 10 ---
```

```
Radio: Entrada nos boxes em Monaco - muito cuidado na pit lane estreita!
```

```
Pit Stop realizado! Pneus novos e combustível reabastecido.
```

```
Radio: Difícil recuperar posições em Monaco!
```

```
Push Lap! Modo de ataque ativado.
```

```
Radio: Push no túnel de Monaco!
```

```
Radio: Fim da corrida em Monaco - Celebrações no pódio real!
```

```
=== Fim da Execução ===
```

```
--- Resultados ---
```

```
Pilotos registrados: 4
```

```
MaxV: 2 pontos
```

```
ChecoP: 0 pontos
```

```
LewisH: 0 pontos
```

```
CharlesL: 2 pontos
```

```
Equipes registradas: 2
```

```
RedBull: 2 pontos (MaxV: 2, ChecoP: 0)
```

```
Ferrari: 2 pontos (CharlesL: 2, LewisH: 0)
```

```
Executando o programa compilado via LLVM...
```

```
[LLVM] Executando módulo (simulação)
```

```
[LLVM] Liberado (simulação)
```

```
Execução concluída com código de saída: 0
```