

# IGNISRIPT

**IGNISRIPT E UMA LINGUAGEM DE PROGRAMACAO QUE UTILIZA CONCEITOS TRADICIONAIS DE LOGICA, ATRIBUICAO, CONTROLE DE FLUXO E ENTRADA/SAIDA PARA PROGRAMAR O COMPORTAMENTO DO PAINEL DE UM CARRO.**

**O PROGRAMA INTERAGE COM ATRIBUTOS DO CARRO COMO VELOCIDADE, RPM, MARCHA, ENTRE OUTROS, E DEVE GERENCIAR RECURSOS COMO GASOLINA, RESPEITAR LIMITES MECANICOS E RESPONDER A CONDICOES COMO O MOTOR ESTAR LIGADO OU NAO.**

greenLight

```
ligado status tune carOn // ADDED SETUP
marcha horsepower tune 1 pitSop // ADDED SETUP
engineRunning status pitSop
flash("Initial engine status:") pitSop
flash(engenenrunning) pitSop
flash("Directly reversed status:") pitSop
flash(reverse engenerunning g) pitSop // reverse
flash(reverse engenrunning) pitSop / reveuse UnOp, needs engine/gear
ligado tune carOff pitSop
redLight
```

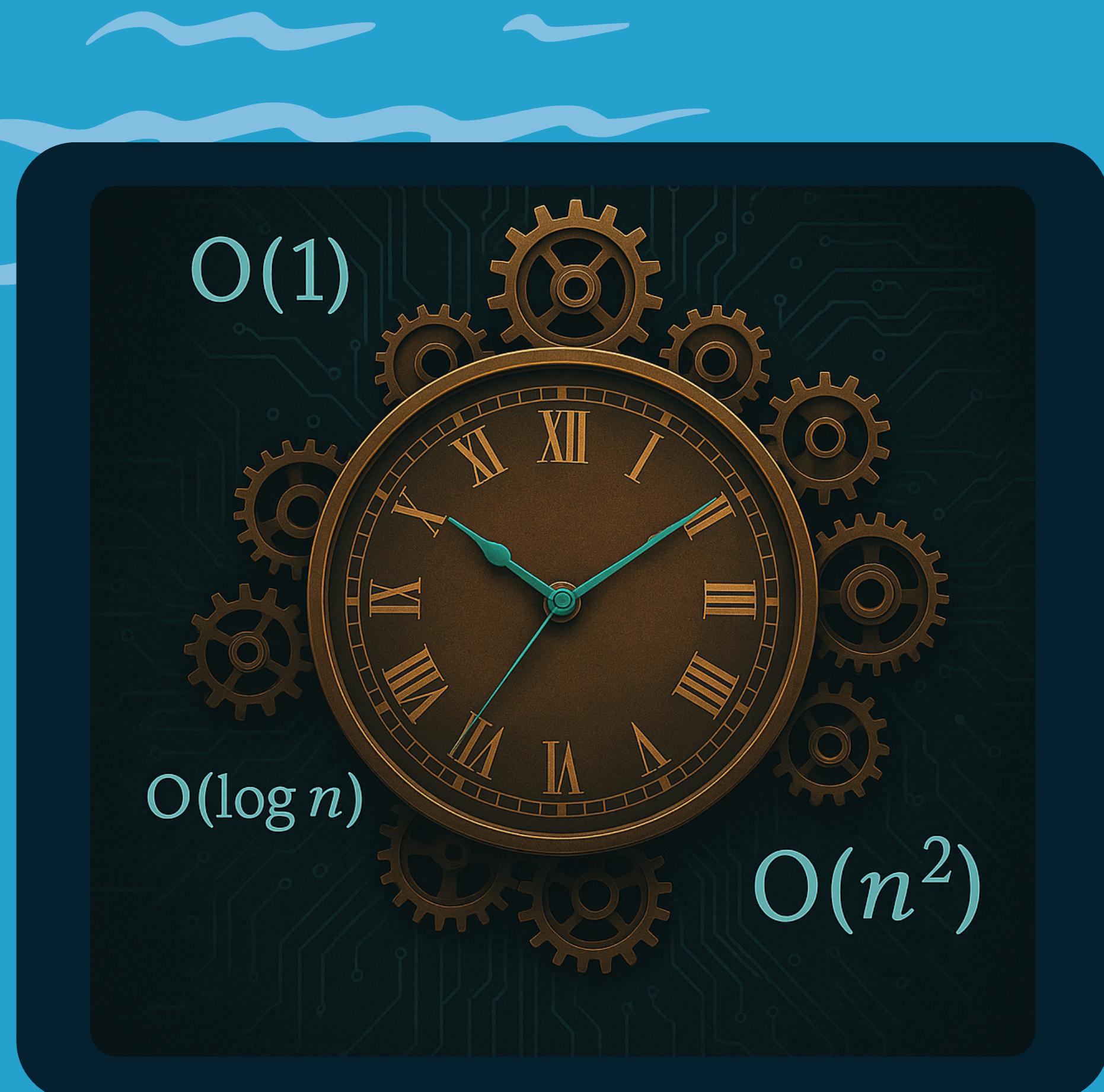


# OBJETIVO

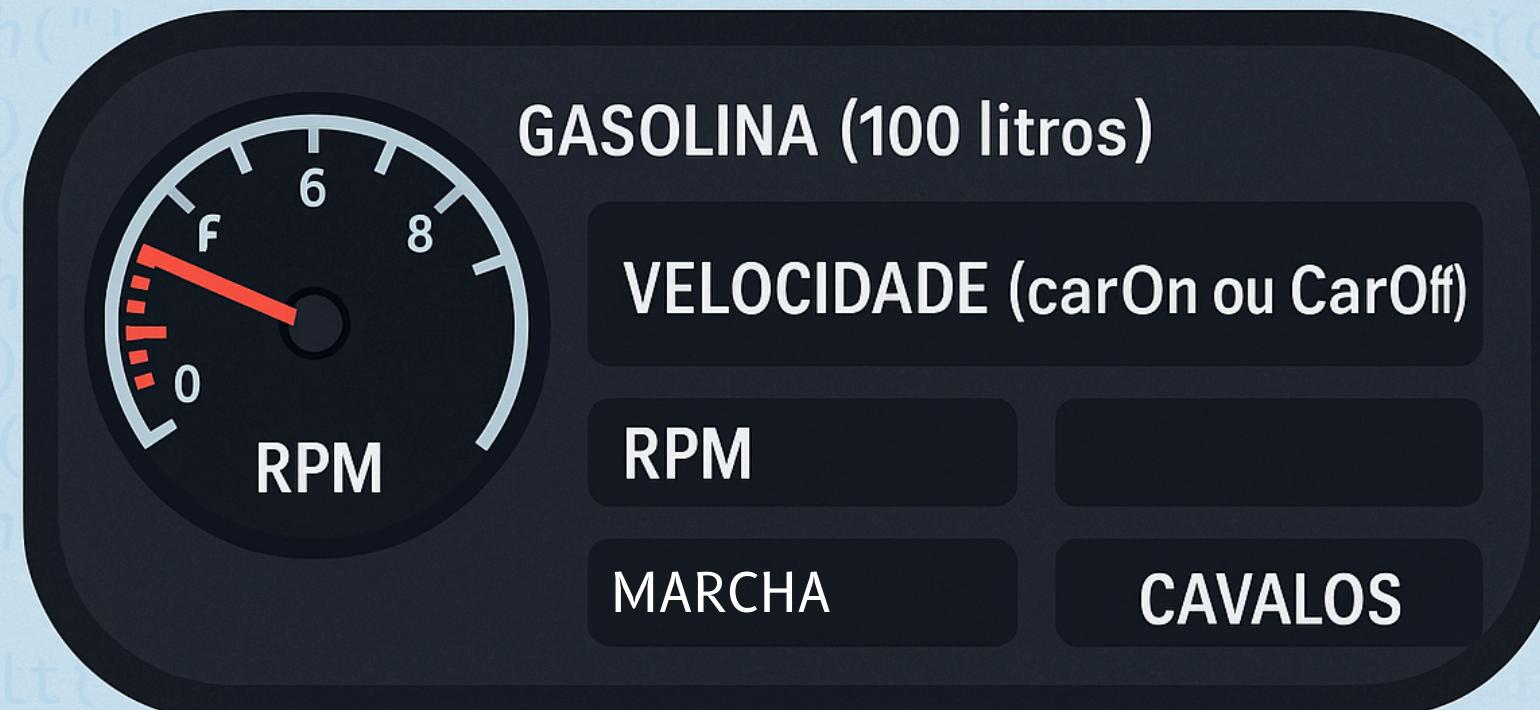
Com o objetivo de ensinar e incentivar pessoas que gostam de carros a aprender programacao, Igniscript adota um linguajar simples e familiar, transformando conceitos tecnicos em comandos intuitivos. Ao usar combustivel como recurso limitado, ela estimula o pensamento logico com foco na eficiencia. No processo, a linguagem continua com potencial como uma ferramenta para resolver problemas computacionais

## CURIOSIDADES

- Palavras-chave são termos automotivos
- Boa para para ensinar fundamentos de computação [temática, foco em reduzir complexidade]



# CARACTERISTICAS



- Gasolina: (100 litros)
- RPM
- Velocidade
- Ligado (carOn ou carOff)
- Marcha
- Cavalos (potencia do motor influencia a velocidade de execução)

- Cada operacao computacional (binaria, unaria, laco, condicional) consome:

- +100 RPM
- -1 litro de gasolina

- O carro comeca com 120 cavalos (quanto mais cavalos menor e o tempo para compilar.)
- E possivel aumentar a cavalaria do carro, mas isso gasta mais gasolina por operacao.
- O rpm nao pode ser alterado diretamente pelo programador.
- O programador e responsavel por trocar de marcha usando marcha tune marcha gearUp 1.
- Se ultrapassar o RPM da marcha atual, o programa encerra com erro.
- Se acabar a gasolina, o carro desliga imediatamente.
- O programador precisa se atentar ao painel do carro [ex: finalizar o programa com carOn gera erro]
- É possivel ter acesso ao painel do carro com o comando info (editado)

# REGRAS DE EXECUÇÃO

Limites de RPM por marcha

Marcha	RPM Maximo	Operacoes permitidas
1	2000	20
2	4000	20
3	6000	20
4	8000	20
5	10000	20

Ex: Comando Info()

--- Car State ---

Engine: carOn

Gasolina: 96.00 L

RPM: 400

Marcha: 1

Cavalos: 120.0

-----

# EXEMPLOS

## PAR ou IMPAR

```
greenLight
    ligado status tune carOn pitStop
    marcha horsepower tune 1 pitStop
    num horsepower pitStop
    temp horsepower pitStop
    flash("Enter a number (horsepower):") pitStop
    num tune sensor() pitStop
    temp tune num clutch 2 pitStop
    temp tune temp accelerate 2 pitStop
    flash("Result:") pitStop
    checkIgnition (num sameAs temp) greenLight
        |   | flash(" Num is Even") pitStop
    redLight
    backup greenLight
        |   | flash(" Num is Odd") pitStop
    redLight
    ligado tune carOff pitStop
redLight
```

Esse código simula a verificação de paridade de um número usando operações de divisão e multiplicação em vez do operador de módulo. Se ao dividir e multiplicar por 2 o número continuar o mesmo, ele é par, senão, ímpar.

Também mostra como se usam os comandos específicos da linguagem Igniscript, como flash, sensor, tune, checkIgnition, backup, e pitStop.

# O DESAFIO DE GAUSS

```
greenLight
    ligado status tune carOn pitStop
    marcha horsepower tune 1 pitStop
    n horsepower tune 100 pitStop
    aux horsepower tune n gearUp 1 pitStop
    soma horsepower tune n accelerate aux pitStop
    soma tune soma clutch 2 pitStop
    temp_plate plate pitStop
    temp_plate tune "Soma de 1 a 100: " turboBoost soma pitStop
    flash(temp_plate) pitStop
    info() pitStop
    ligado tune carOff pitStop
redLight
```

Este código em Igniscript calcula a soma dos números de 1 a 100 usando a fórmula de Gauss. A linguagem simula um carro com consumo de combustível, e laços poderiam estourar o limite de gasolina rapidamente. Esse programa demonstra como a linguagem encoraja soluções diretas e matemáticas (como a de Gauss), evitando laços, o que reduz drasticamente o consumo de combustível simulado e promove um raciocínio algorítmico mais elegante.