

Track Sense

Generated by Doxygen 1.9.8



---

<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Carro Class Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.1.2 Constructor & Destructor Documentation . . . . .	5
3.1.2.1 Carro() . . . . .	5
3.1.3 Member Function Documentation . . . . .	6
3.1.3.1 acelerar() . . . . .	6
3.1.3.2 frear() . . . . .	6
3.1.3.3 getVelocidade() . . . . .	6
<b>4 File Documentation</b>	<b>7</b>
4.1 TrackSense.hpp . . . . .	7
<b>Index</b>	<b>9</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Carro</a>	Representa um carro no sistema . . . . .	<a href="#">5</a>
-----------------------	--	-------------------



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

src/ <a href="#">TrackSense.hpp</a> . . . . .	7
---	---





## Chapter 3

# Class Documentation

### 3.1 Carro Class Reference

Representa um carro no sistema.

```
#include <TrackSense.hpp>
```

#### Public Member Functions

- `Carro` (const std::string &marca, const std::string &modelo)  
*Construtor do carro.*
- void `acelerar` (int valor)  
*Acelera o carro.*
- void `frear` (int valor)  
*Reduz a velocidade do carro.*
- int `getVelocidade` () const  
*Obtém a velocidade atual.*

#### 3.1.1 Detailed Description

Representa um carro no sistema.

A classe `Carro` gerencia informações e comportamentos de um veículo, incluindo sua marca, modelo, velocidade atual e métodos para acelerar, frear e obter dados.

Exemplo de uso:

```
Carro c("Toyota", "Corolla");  
c.acelerar(20);  
std::cout << c.getVelocidade() << std::endl;
```

#### 3.1.2 Constructor & Destructor Documentation

##### 3.1.2.1 Carro()

```
Carro::Carro (  
    const std::string & marca,  
    const std::string & modelo )
```

Construtor do carro.

**Parameters**

<i>marca</i>	Marca do carro.
<i>modelo</i>	Modelo do carro.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 acelerar()

```
void Carro::acelerar (
    int valor )
```

Acelera o carro.

Incrementa a velocidade atual pelo valor informado. Se o valor for negativo, ele é ignorado.

**Parameters**

<i>valor</i>	Valor em km/h para aumentar a velocidade.
--------------	---

#### 3.1.3.2 frear()

```
void Carro::frear (
    int valor )
```

Reduz a velocidade do carro.

**Parameters**

<i>valor</i>	Valor em km/h para reduzir a velocidade.
--------------	--

#### 3.1.3.3 getVelocidade()

```
int Carro::getVelocidade ( ) const
```

Obtém a velocidade atual.

**Returns**

Velocidade em km/h.

The documentation for this class was generated from the following file:

- src/TrackSense.hpp

## Chapter 4

# File Documentation

### 4.1 TrackSense.hpp

```
00001 #ifndef TRACKSENSE_HPP
00002 #define TRACKSENSE_HPP
00003
00019 class Carro {
00020 public:
00026     Carro(const std::string& marca, const std::string& modelo);
00027
00035     void acelerar(int valor);
00036
00041     void frear(int valor);
00042
00047     int getVelocidade() const;
00048
00049 private:
00050     std::string marca;
00051     std::string modelo;
00052     int velocidade;
00053
00059     bool validarVelocidade(int v) const;
00060 };
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077 #endif // tRACK
```



# Index

acelerar

Carro, [6](#)

Carro, [5](#)

acelerar, [6](#)

Carro, [5](#)

frear, [6](#)

getVelocidade, [6](#)

frear

Carro, [6](#)

getVelocidade

Carro, [6](#)

src/TrackSense.hpp, [7](#)