

# Módulos



É uma unidade que contem classes, interfaces e variáveis permitindo a organização da aplicação.

**Classes**

**Interfaces**

**Funções**

**Variáveis**

Para utilizar “recursos” de um módulo é necessário fazer a exportação do que se quer utilizar.

# Módulos



A exportação de recursos pode ser feita de duas maneiras:

1

```
export interface Gamer {  
  memoriaVideo: number;  
}
```

2

```
interface Gamer {  
  memoriaVideo: number;  
}
```

```
export {Gamer as G}
```

# Módulos



```
import {Laptop} from "../pasta"
```

```
class Lenovo extends Laptop {  
    // ...  
}
```

- Os tipos podem ser separados por virgula
- O nome dos tipos não devem ter a extensão .ts

# Organizando a casa



**Primeiro vamos criar um arquivo chamado computador-base.ts e transferir os tipos básicos do exemplo (Laptop e Gamer).**

**Em seguida devemos marcar o que será exportado. É possível exportar variáveis, classes, interfaces ou até mesmo funções.**

**Continuando com a organização, vamos criar um arquivo chamado lenovo.ts para a classe Lenovo.**

**Usamos export para disponibilizar recursos e import para ler recursos que foram exportados.**

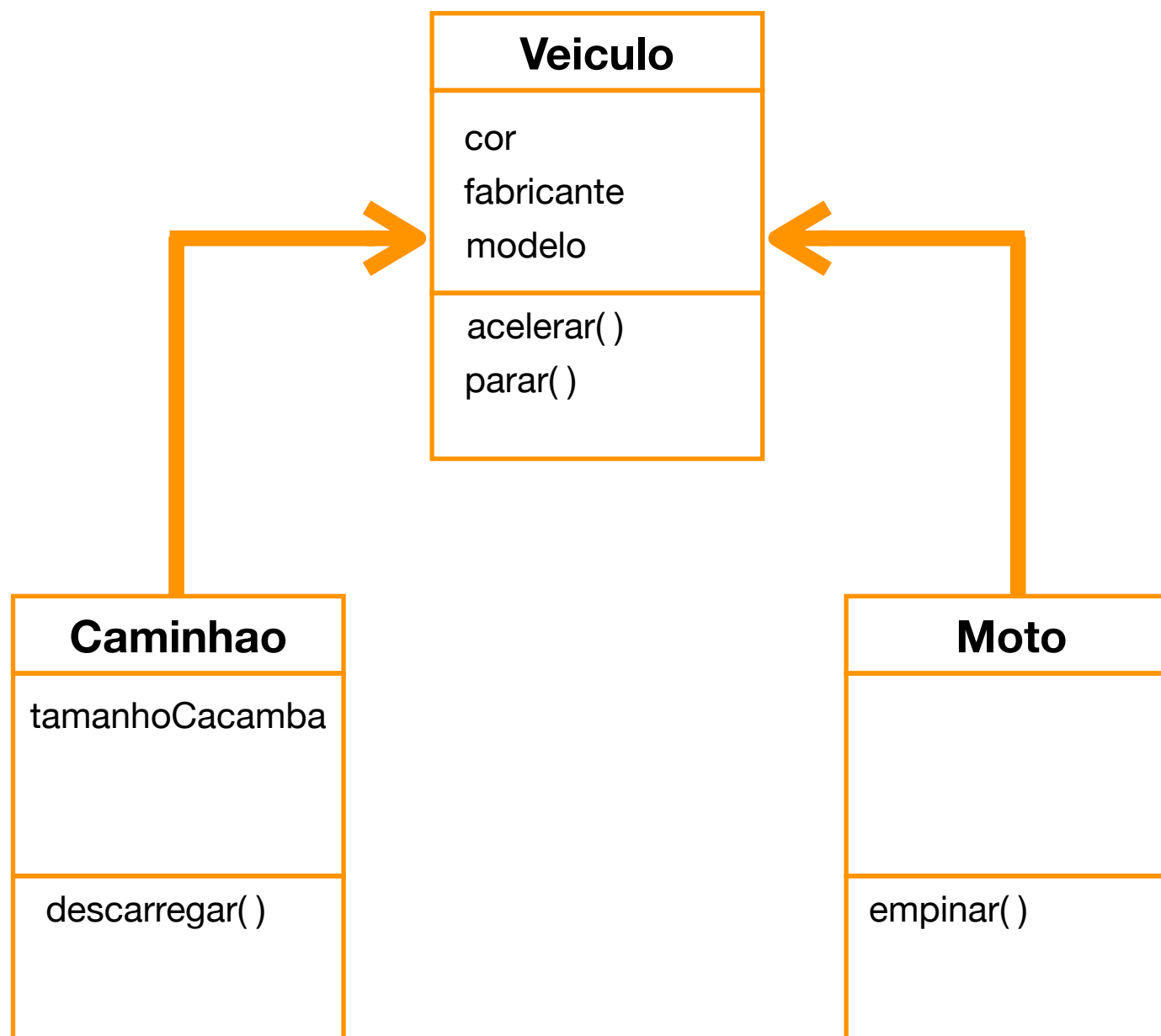
**Também vamos criar um arquivo lenovox.ts para a classe LenovoX.**

**Finalmente acertar os imports do arquivo principal da aplicação.**

# Tarefa



Para praticarmos os conhecimentos adquiridos até o momento desenvolva uma aplicação que tenha como base o seguinte diagrama de classes.



Após definir os arquivos `.ts` necessários para essa aplicação, crie o arquivo principal e faça algumas instâncias usando as classes criadas.

# Primeira aplicação



Usamos o angular-cli para criar a estrutura inicial de uma aplicação em angular.

```
ng new <nome-aplicação> --prefix=<pre>
```

Para testar se a estrutura criada está funcionando podemos iniciar o servidor de desenvolvimento do angular.

```
ng serve
```

**localhost:4200**