

## Organização do código apresentado

### 1. Quais foram as classes consideradas na primeira versão da aplicação foi testada?

A primeira versão tinha as seguintes classes e principais funcionalidades:

- Carro
  -
- Bateria
  - Controla a capacidade da bateria
- Sinal de Emergência
  - Controla o estado do sinal de emergência
- Velocidade
  - Controla a velocidade do carro
- Piloto
  - Tem acesso aos controlos do carro
  - Consegue sair e entrar num carro
- Autodromo
  - Tem uma pista e uma garagem
  - Tem carros e pilotos associados
- Pista
  - Tem nela mesma uma corrida contida
- Garagem
  - Armazena os carros
- Direção Geral de Viação
  - Detém os carros, pilotos e autódromos
  - Gere a criação, procura e eliminação dos mesmos.
- Visualização
  - Controla as vistas de menu do programa
  - Interpreta os comandos em cada um dos módulos.

**2. Quais os conceitos que identificou ao ler o enunciado?**

- Carro
- Bateria
- Sinal de Emergência
- Velocidade
- Piloto (e as suas derivadas)
- Direção Geral de Viação
- Autódromo
- Pista
- Garagem
- Corrida
- Visualização
- Campeonato
- Comando

**3. Relativamente a duas das principais classes da aplicação, identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objectos.**

Carros: Os objetos desta classe são criados, armazenados e destruídos na classe Direção Geral de Viação.

Pilotos: Os objetos desta classe são criados, armazenados e destruídos na classe Direção Geral de Viação.

**4. Indique um exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento.**

A responsabilidade de carregar todos os carros, descritos num ficheiro, é atribuída à classe Direção Geral de Viação, pois detém uma coleção de carros.

- 5. De entre as classes que fez, escolha duas e justifique por que considera que são classes com objectivo focado, coeso e sem dispersão.**

Bateria: tem dados e responsabilidades, relativos apenas à capacidade da bateria, por exemplo, carrega e descarrega a bateria.

Garagem: tem dados e responsabilidades, relativos apenas à armazenagem dos carros na mesma.

- 6. Relativamente à aplicação entregue, quais as classes que considera com responsabilidades de interface com o utilizador e quais as que representam a lógica?**

Responsabilidade de interface: Visualização

Responsabilidade lógica: Piloto, carro, garagem, autódromo, direção geral de viação, sinal de emergência, pista, bateria, velocidade e corrida.

- 7. Identifique o primeiro objecto para além da camada de interacção com o utilizador que recebe e coordena uma funcionalidade de natureza lógica?**

O primeiro objeto que recebe e coordena uma funcionalidade de natureza lógica é a Direção Geral de Viação.

- 8. A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes? Indique um exemplo em que esta classe delega uma funcionalidade noutra classe.**

Neste momento, a classe envolvente de toda a lógica é a classe Visualização. Por exemplo, sempre que é inserido um comando, neste caso o carregaP, para carregar os pilotos, a classe delega a responsabilidade à direção geral de viação.

**9. Dê um exemplo de uma funcionalidade que varia conforme o tipo do objecto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade. (Não é necessário responder a esta pergunta na meta 1).**

Este caso irá acontecer aquando do desenvolvimento do tipo de piloto, pois Piloto será Crazy, rápido ou surpresa.

**10. Apresente as principais classes da aplicação através da seguinte informação:**

**Classe:** Direção Geral Viação

**Responsabilidades:**

- Carregar os pilotos, carros e autódromos
- Procurar os pilotos, carros e autódromos
- Eliminar os pilotos, carros e autódromos
- Armazena os pilotos, carros e autódromos

**Colaborações:** Visualização

**Classe:** Visualização

**Responsabilidades:**

- Interpretar os comandos
- Desenhar a interface

**Colaborações:** Visualização

**Classe:** Piloto

**Responsabilidades:**

- Entrar e sair no carro
- Aceder ao controlo do carro

**Colaborações:** Direção Geral de Viação

## Funcionalidades implementadas

| Componente do Trabalho    | Realizado | Realizado Parcialmente | Não realizado |
|---------------------------|-----------|------------------------|---------------|
| Interpretação de comandos | X         |                        |               |
| Criar autódromo           | X         |                        |               |
| Criar piloto              | X         |                        |               |
| Comando campeonato        |           |                        | X             |
| Comando passatempo        |           | X                      |               |
| Visualização movimento    |           |                        | X             |

## Melhorias

Está prevista uma melhoria na organização da classe Visualização e a criação de uma classe estática Comando, que irá executar todos os comandos inseridos no programa.

Melhorar interface e acabar o desenvolvimento de classes de controlo.

Desenvolver o movimento dos carros e os tipos de pilotos.