

## Grupos

Nº pessoas 4, max: 5.

### *Trabalho Competição de Cavalos:*

#### Requisitos:

No trabalho é aconselhado usar: listas, leitura e escrita de ficheiro, enums, herança, os métodos

toString() das classes devem ser override, deve existir overload (Sugiro a aquisição de um animal ou vários) métodos de string que garantam uma correcta leitura, deve fazer uso correcto de properties, construtores, métodos.

No relatório deve constar todas as decisões que tomou, desde os atributos, passando pelos métodos, etc. A cada passo de resolução deve de forma breve explicar o conceito do que está por trás recorrendo ao que foi lecionado.

O relatório deve conter Capa, índice, introdução, desenvolvimento com “expectativas iniciais” “escolhas de implementação”, “conceitos teóricos”, “dificuldades encontradas”. Na conclusão estabeleça uma comparação entre o que foi planeado o que se implementou justificando, aponte aqui como superou as dificuldades listadas no desenvolvimento.

Se porventura desejarem adicionar algo novo, para poder aplicar algo ao projeto que não parece ser exequível, ou não é a solução ótima com os pressuposto apresentados pode fazê-lo, devendo assinar numa secção do relatório chamada “Extra” e nesta deve descrever o que fez e a motivação.

Não esquecer da bibliografia pode usar o formato que quiser desde de que seja mantida coesão.

---

Pretende-se desenvolver um jogo que simula corridas de Cavalos e Galgos, o jogador deve ser gestor de uma Coudelaria e equipa.

Deve existir equipas opositoras. Deve ser possível listar os animais e stats das equipas.  
Deve ser possível comprar, vender, trocar cavalos entre jogadores ou jogador e AI.

Deve existir uma espécie de calendário, onde cada dia, exceto o dia do evento, o jogador por fazer uma série de ações.

Deve existir diferentes mini-eventos Lesões, Manifestações, quedas, etc. Use a sua imaginação para tornar dinâmico.

O jogo deve conter uma lista de staf cada um com funções diferentes e salários.

Idealmente deve ser possível guardar o estado do jogo.

Para cumprir os requisitos pedidos a cima, adicione o que achar pertinente, sendo que, no que for adicionado de sua inovação deve ser discriminado e encaixado com os conceitos teóricos.  
Como extra: Conta a prevenção de erros, Linq, Interfaces e outras matérias dadas pós colecções e Enums