**Relatório:**

Para este efólio, utilizei os conteúdos lecionados na UC e vários artigos informativos e audiovisuais encontrados na world wide web. Utilizei principalmente os sites geeksforgeeks.org, stackoverflow.com e Youtube.com.

Para a execução do pedido no briefing, tive algumas dificuldades, pois uma vez já referido no fórum e após consultar informação nos sítios descritos em cima, as resoluções presentes na página da UC estão desatualizadas e evolvem algumas más práticas de escrita de código, atrasando o processo da escrita do código, uma vez que era preciso tentar entender o que estava mal e como fazer bem, num curto espaço de tempo.

Algumas escolhas feitas foram devido ao facto de o tempo ser escasso, sendo essas para mim as mais fáceis e que consumiam menos tempo.

Após ter analisado a resolução do efólioA, tomei por bem também criar uma classe para apresentar a interface do programa, tentando não incluir outputs nas outras classes. Nesta classe, fiz vários métodos para mostrar no ecrã as várias opções a tomar, lendo e validando de seguida o input inserido pelo utilizador passando então a outros métodos dentro da classe interface onde de seguida eram inseridos ou lidos os dados requeridos.

Para guardar os dados, decidi criar vários ficheiros de texto por jogos e modalidades, mas guardado no mesmo ficheiro os jogos entre as mesmas equipas da casa e visitante da mesma modalidade, delimitando cada jogo com uma palavra “Jogox:”, onde x indica o número do jogo em questão começando em x=1. Para tal, os nomes dos ficheiros, ao serem guardados, iriam identificar a modalidade, na primeira letra do nome, e depois iriam ter, por ordem de casa e visitante, respetivamente, o nome das equipas participantes no jogo, onde se pode ver nos ficheiros de texto anexados para poder já ter uma base para teste.

Basicamente o processo verifica se o ficheiro existe, abrindo temporariamente o ficheiro em modo leitura e verificar se abriu, uma vez que em modo leitura, caso o ficheiro não exista, este não abre. Caso exista verifica se tem só um jogo ou se tem mais, e no caso do último, pergunta o jogo a ver, saltando informação até o encontrar, guardando de seguida nas diferentes variáveis e estruturas de dados. O ficheiro é aberto, para escrita, em modo “append”, para adicionar conteúdo ao fim do mesmo e não eliminar o já presente.

Os dados são guardados com delimitadores entre cada valor para facilitar a sua leitura.

Para ler os dados, o processo de verificação de existência de ficheiro é idêntico ao descrito acima. Caso exista mais de um jogo, escolhe-se o que se quer ver. Os delimitadores usados a guardar os dados são explorados aqui. A função getline() pode ser usada em quaisquer dados que estejam em forma de “stream”, podendo esta ser criada com um método de stringstream(). Dentro da função getline(), pode-se adicionar um terceiro parâmetro para delimitar até onde ela função e lê a linha da “stream” selecionada. Segundo isto, decidi, então, utilizar estes dois métodos para criar stream das strings lidas do ficherio, usando depois a função getline() para delimitar as diferentes variáveis a guardar, por jogo, fazendo-o, para a parte de ler os jogadores, dentro de um loop fixo, por modalidade, uma vez que existe um número fixo de jogadores por equipa em cada modalidade e as posições dos mesmo são diferentes.

Para inserir os dados, escolhe-se a modalidade do jogo a inserir e a partir dai, existe um método para cada modalidade, onde algumas funções dentro deles são parecidas, utilizando um identificador para a função distinguir entre modalidades. O programa pregunta ao utilizador os vários dados, sendo estes verificados depois de inseridos, e caso estejam de acordo com os parâmetros, são inseridos nas variáveis em questão. Mais uma vez, como o número de jogadores é fixo, utilizei “loops” fixos para a sua inserção. As substituições, expulsões e golos/pontos, estão inseridos e guardados em forma de string, por ser mais rápido, sendo estas depois separadas em “tokens”, pelos delimitadores, na sua apresentação em ecrã.

Para apresentar as informações no ecrã, decidi fazer isso através de um menu, onde o utilizador escolhe que informação quer ver, por exemplo, resultado, jogadores de uma equipa, etc.

Cada classe tem os seus setters e getters dos diferentes atributos. Os métodos de inserção e verificação de dados também estão dentro de cada classe em questão.

Para os pontos, como é quase humanamente impossível marcar mais de 100 pontos num jogo, meti 99 como o máximo para facilitar a verificação. No tempo de jogo, utilizei como valor máximo 90 minutos para o futebol e 40 minutos para o basquetebol.