

# Laboratório de Sistemas Digitais

Ano Letivo 2021/22

## Projeto Final – enunciado 2

### Marcador de ténis

#### 1. Introdução

Pretende-se realizar, com o *kit* de desenvolvimento *Altera DE2-115*, um marcador desportivo para a modalidade de ténis. A especificação aqui apresentada baseia-se nos recursos de entrada/saída mais simples do *kit* (visores hexadecimais, LED, botões de pressão, comutadores), explorados nos guiões das aulas práticas.

#### 2. Pontuação em ténis

O sistema de pontuação de ténis segue um **modelo hierárquico** envolvendo (em sequência ascendente) pontos (**points**), jogos (**games**), e conjuntos de jogos (**sets**).

*[Informação oficial completa sobre as regras do ténis pode ser consultada, por exemplo, em <https://www.itftennis.com/en/about-us/governance/rules-and-regulations/>. No texto seguinte, para evitar ambiguidade, emprega-se a terminologia inglesa].*

Os aspetos mais relevantes para este trabalho são:

- O encontro (**match**) termina quando um jogador (vencedor do encontro) ganha o número pré-definido de sets: por norma, 3 no ténis masculino (o que implica um máximo de 5 sets) e 2 em ténis feminino (máximo de 3 sets).

- O **set** termina quando um jogador (vencedor do set):

- ganha 6 *games* (normais) antes de o adversário chegar a 5 **ou**
- ganha 7 *games* (normais) antes de o adversário chegar a 6 **ou**
- havendo empate a 6 *games*, ganha um sétimo *game*, dito de desempate (*tie-break*).


- Há, portanto, dois tipos de *game* a considerar:

- Um **game normal** termina quando um jogador (vencedor do *game*) atinge um número de *points* igual ou superior a 4 com diferença para o adversário igual ou superior a 2. Até ocorrer igualdade a 3 *points* (ou até o *game* terminar, se não ocorrer essa igualdade), o marcador deve assinalar a pontuação de cada jogador, usando códigos 0, 15, 30 e 40, respetivamente para 0, 1, 2 e 3. No caso de ocorrer igualdade a 3, e em qualquer situação de igualdade a partir daí, o marcador deve assinalar empate (**deuce**: código d para ambos os jogadores). A partir de *deuce*, deve assinalar vantagem para o jogador que conquistar um ponto (**advantage**: código Ad apenas para esse jogador; mostrador de pontuação do adversário apagado). A partir da situação de *advantage*, o *game* termina se o jogador em vantagem conquistar o *point* seguinte e regressa a *deuce* se for o seu adversário a fazê-lo.
- Um **tie-break** termina quando um jogador (vencedor do *game*) atinge um número de *points* igual ou superior a 7 com diferença para o adversário igual ou superior a 2. Neste caso, o marcador deve mostrar simplesmente o número de *points* de cada jogador.

Além da pontuação de cada jogador (*points*, *games* e *sets*), um marcador de ténis deve indicar qual serve no *point* seguinte. O serviço (**service**) é gerido em alternância. Em *games* normais, pertence ao mesmo jogador durante todo o *game* e passa ao outro no início do seguinte. Em *tie-breaks*, alterna sempre que o número de *points* já disputados nesse *tie-break* atinge um valor ímpar.

### 3. Especificações

O sistema deve contemplar as funcionalidades descritas na tabela seguinte:

<b>Reset</b>
- Permitir <i>reset</i> (colocação do sistema no seu estado inicial) através de botão, de preferência com efeito após um tempo de atuação longo (e.g. 3s), para evitar <i>reset</i> acidental.
<b>Pontuação</b>
- Receber informação fornecida pelo utilizador (árbitro) sobre o andamento do jogo. Sugere-se o uso de botões de pressão ligados a <i>debouncers</i> : <ul style="list-style-type: none"><li>• KEY3 – <i>point</i> do jogador A</li><li>• KEY0 – <i>point</i> do jogador B</li></ul>
- Gerir, de acordo com as regras apresentadas, registos internos com a informação de pontuação (dois conjuntos inteiramente análogos, um para cada jogador).
- Mostrar a informação desses registos nos locais indicados e sob a forma adequada: <ul style="list-style-type: none"><li>• <i>Points</i> no <i>game</i> em curso HEX7 e HEX6 (jogador A)                      HEX5 e HEX4 (jogador B) indicações possíveis em <i>games</i> normais:  em <i>tie-breaks</i>:                      quaisquer números inteiros</li><li>• <i>Games</i> no <i>set</i> em curso (cujo número deve ser apresentado em HEX3) HEX1 (jogador A)                      HEX0 (jogador B)</li><li>• <i>Sets</i> no encontro LEDG[6..4] (jogador A)                      LEDG[2..0] (jogador B)</li></ul>
<b>Serviço</b>
Indicar, através de LED, a quem pertence o serviço: <ul style="list-style-type: none"><li>• LEDR[17] – <i>serviço</i> do jogador A</li><li>• LEDR[12] – <i>serviço</i> do jogador B</li></ul> O primeiro serviço deverá ser atribuído preferencialmente de forma aleatória.
<b>Break points</b>
- Detetar as situações em que o próximo <i>point</i> permite a conquista de um <i>game</i> normal em que o serviço pertence ao adversário e assinalá-las no marcador através da indicação $b \leq Pt$ nos visores HEX3 a HEX0, alternando a uma frequência de 1Hz com a informação (especificada antes) sobre <i>games</i> no <i>set</i> em curso.

#### 4. Diretrizes e recomendações

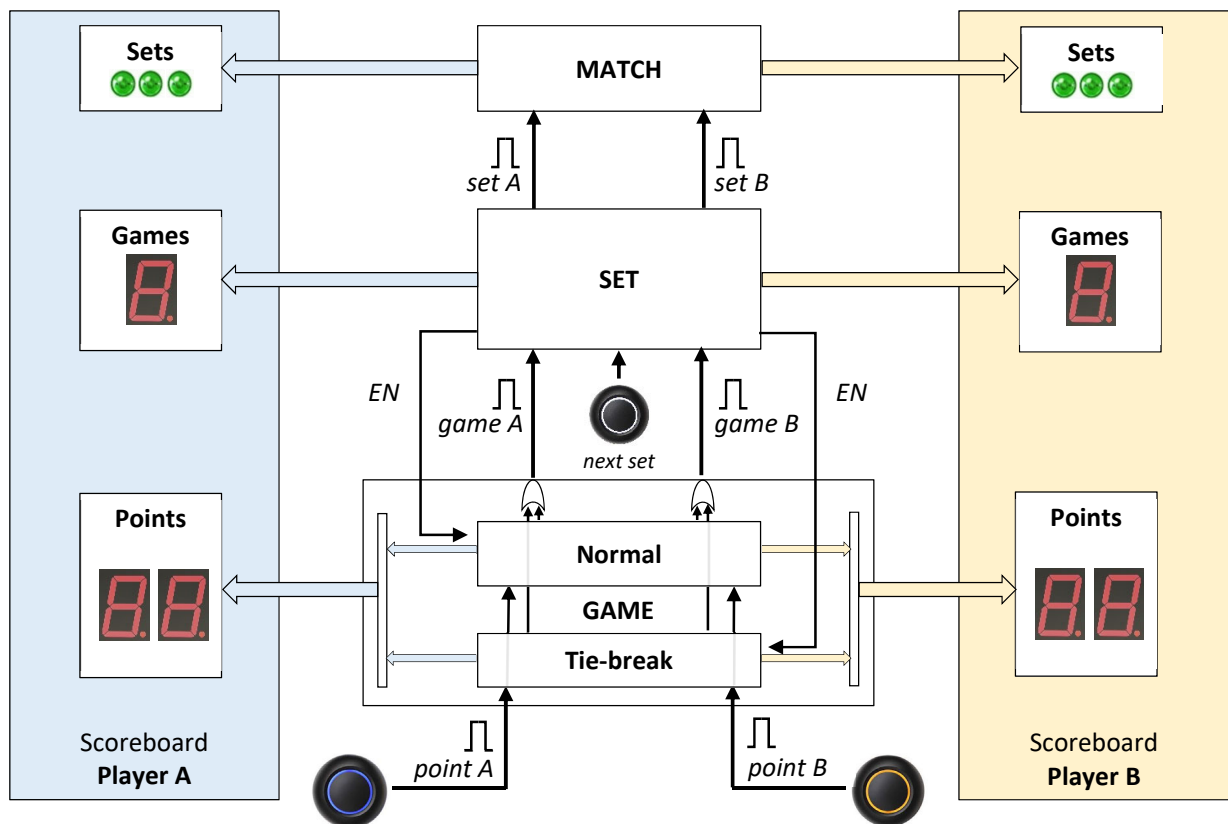
O diagrama de blocos pretende apenas apresentar um esboço da estrutura geral do sistema, evidenciando o seu funcionamento em ‘cascata’, que se presta à aplicação do modelo hierárquico preconizado na UC. Recomenda-se uma implementação faseada, planeando, construindo e testando cada um dos blocos e integrando-os gradualmente numa estratégia *bottom-up*:

**Fase 1 (6 valores):** - Marcador para um único *game* normal:

- O bloco de controlo deve basear-se em MEF – a primeira tarefa é construir cuidadosamente o diagrama de estados.
- Sugere-se minimizar o número de estados com o apoio de contadores na fase inicial de pontuação numérica (até à eventual ocorrência de ‘deuce’).
- O bom funcionamento da MEF em todas as situações deve ser comprovado por simulação.
- A gestão dos visores de 7 segmentos tem que atender à necessidade de apresentar não só informação numérica (0, 15, 30, 40) mas também códigos não numéricos (para ‘deuce’ e ‘advantage’).

**Fase 2 (3 valores):** Bloco de gestão de *tie-break*:

- Recomenda-se uma MEF apoiada por contadores de *points*.
- Os dois blocos já construídos devem ser integrados usando uma forma de seleção provisória (por exemplo através de um comutador).



**Fase 3 (3 valores):** Bloco de gestão de set:

- A solução pode ser, de novo, uma MEF apoiada por contadores (agora de *games*).
- Note-se que cabe a este bloco escolher entre *game* normal e *tie-break*. A seleção provisória da fase anterior pode ser agora dispensada, integrando os três blocos.
- No final do set, o score (*games* e, no caso de ter havido decisão por tie-break, também *points*) deve manter-se visível até o utilizador decidir expressamente avançar (botão de pressão 'next set').

**Fase 4 (2 valores):** Bloco de gestão do *match* e integração. O momento do final do encontro deve ser assinalado com um efeito especial (à escolha) que saliente o vencedor.

**Fase 5 (3 valores):** Sistema de gestão de serviço (vide especificação em 3)

**Fase 6 (3 valores):** Sistema de deteção de *break points* (vide especificação em 3).

Podem ser implementadas, e serão valorizadas, funcionalidades adicionais, tais como um sistema de 'undo' (cancelamento do último ponto e perfeita reposição do estado anterior).