



# **Campo DEISlado**

## Enunciado do projecto prático - Segunda Parte

### Nota prévia

Na parte 2 do projeto devem manter os grupos criados na parte 1. Em casos excepcionais, mediante autorização prévia do professor das práticas e com uma justificação adequada, é possível dividir grupos na parte 2 (ou seja, cada elemento do grupo passa a entregar sozinho). Não é possível formar novos grupos.

### Objectivo

Mantém-se o objetivo da parte 1, desenvolver um programa em Kotlin que implemente o jogo “Campo DEISlado”.

Deverão, portanto, continuar o projeto entregue na parte 1.

**Este é o enunciado da Segunda Parte.**

### Descrição do Projeto

A segunda parte do projecto encontra-se descrito em vídeo:

<https://www.youtube.com/watch?v=tM-99QyLFvc>

Poderão ser identificados erros/omissões no vídeo. Devem estar atentos a comentários que sejam adicionados ao vídeo.

Caso tenham dúvidas sobre o conteúdo do vídeo, devem colocá-las sob a forma de comentários no youtube, indicando o timestamp do vídeo ao qual diz respeito a pergunta.

### Restrições Técnicas - Segunda Parte

Na parte 2 deixam de existir algumas das restrições técnicas da parte 1. É proibida a utilização de estruturas dinâmicas como listas, maps e sets. Também não é permitido criarem classes. Podem usar constantes globais mas **não podem usar variáveis globais, por ser uma má prática**. Também por ser uma má prática de programação (aliás, nem sequer se fala disso nas aulas), é proibido utilizar o `break`, o `continue` e o `exitProcess()`. Quase tudo o resto é permitido (o Drop Project avisará se usarem instruções proibidas) - no entanto, salientamos que **o projeto pode ser implementado apenas com as instruções ensinadas nas aulas teóricas**. Não serão esclarecidas dúvidas sobre código que use instruções não ensinadas nas aulas.

### Utilização de LLMs

A utilização de LLMs (ChatGPT, Gemini, Copilot, Claude, etc.) é permitida e até incentivada.

No entanto, tal como tem sido referido nas fichas práticas semanais, devem evitar recorrer ao LLM constantemente. Comecem por tentar implementar sozinhos as funções do projeto antes de recorrerem ao LLM. Se recorrerem ao LLM, não lhe peçam código! Peçam-lhe para analisar os erros do vosso código ou para vos explicar como devem implementar. **Lembrem-se que ele pode ser usado para vos apoiar na construção do programa e nunca para vos substituir completamente.**

Relembramos que haverá defesa individual do projeto, durante a qual terão que alterar o código do projeto. Se o código foi produzido pelo LLM e não o entendem (embora funcione), provavelmente não conseguirão alterá-lo com sucesso.

### Entrega

O projecto será entregue via Drop Project (DP), através do link:

<https://deisi.ulusofona.pt/drop-project/upload/fp-2526-projeto-p2>

Não serão aceites entregas por outra via.

Os alunos são incentivados a testar o projeto no DP à medida que o vão implementando. Podem e devem fazer quantas submissões acharem necessárias. Para efeitos de avaliação será considerada a melhor submissão feita dentro do prazo.

**De notar que os alunos, antes de enviarem para o DP, devem testar no seu próprio computador.**

### Vídeo demonstrativo

Nesta segunda parte, os alunos são convidados a criar um vídeo demonstrativo do jogo.

O vídeo deve apresentar um jogo completo mas não pode exceder 2 minutos de duração.

O vídeo deve ser carregado para o youtube e configurado como “unlisted” de forma a não aparecer nos resultados de pesquisa. O título do vídeo deve incluir os números dos alunos (ex: “campo-deisiado-21701234-21704567”).

O URL do vídeo deve estar num ficheiro README.txt que deve ser colocado na raiz do projeto (“ao lado” do AUTHORS.txt)

O jogo deve ser narrado, preferencialmente por ambos os elementos do grupo. Sejam criativos na forma como narram o jogo!

Este vídeo não é obrigatório, no entanto pode valer até 1 valor na nota final da parte 2.

<b>Prazo de entrega intermédio</b>
------------------------------------

Os grupos têm que fazer pelo menos uma submissão no Drop Project que **não tenha erros de compilação** até dia 14 de Dezembro às 23h59. Não é preciso passarem nenhum teste, basta terem todas as variáveis e cabeçalhos das funções implementados.

Grupos que falhem este requisito têm nota zero na parte 2 do projeto e **reprovam na componente prática da disciplina (1ª época)**

<b>Prazo de entrega</b>
-------------------------

A data limite de entrega é o dia **30 de Dezembro de 2025**, pelas **10h00 da manhã** (hora de Lisboa, Portugal).

Os alunos que entreguem depois do prazo, ainda durante o dia 30 de Dezembro de 2025, até às 23h59, **terão uma penalização de 5 valores** na nota final desta parte do projecto.

Não serão aceites entregas após dia 30 de Dezembro de 2025 às 23h59. Nesse caso os alunos terão nota zero no projeto, **reprovando na componente prática da disciplina (1ª época)**.

<b>Avaliação</b>
------------------

Mantêm-se as regras da primeira parte, às quais se acrescenta:

- Na **nota final** do projeto existe a nota mínima de 9.5 (nove e meio) valores.
- Há um conjunto mínimo de funcionalidades implementadas para que o projeto possa ser defendido com sucesso. **Ou seja, alguns dos testes vão ser obrigatórios.**
  - Estes testes estarão identificados com o sufixo “**\_OBG**” (ex: test\_XXX\_OBG)
  - Os alunos que entreguem projectos que não passem **todos os testes obrigatórios** serão reprovados em primeira época.

**As defesas realizar-se-ão no dia 5 de Janeiro**, de forma presencial. A presença da defesa é obrigatória para todos os alunos - quem não comparecer terá zero na nota final do projeto.

<b>Cotações da Segunda Parte</b>
----------------------------------

A avaliação do projecto será feita através dos seguintes tópicos:

Tópico	Descrição	Pontuação (0..20)
Qualidade de código	O código cumpre as boas práticas de programação ensinadas nas aulas (nomes apropriados para as variáveis e funções em camelCase, utilização do val e do var, código bem indentado, funções que não sejam demasiado grandes, etc.).	2
Testes automáticos	Será aplicada uma bateria de testes automáticos cujo relatório poderá ser consultado após cada submissão. Quanto mais testes passarem, melhor nota terão nesta componente.	14
Avaliação manual da aplicação	Os professores farão testes adicionais assim como inspeção de código para avaliar esta componente	3
Vídeo	Ver secção “Vídeo de demonstração”	1

<b>Cópias</b>
---------------

Mantêm-se as regras indicadas no enunciado da parte 1

<b>Outras informações relevantes</b>
--------------------------------------

Todas as descritas no enunciado da parte 1.

**FIM**