



Campo DEISlado

Enunciado do projeto prático — Primeira Parte

Objetivo

Este projeto tem como objectivo o desenvolvimento de um programa usando a **linguagem Kotlin**.

Para alcançar estes objetivos, os alunos devem ter em conta tudo o que foram aprendendo em Fundamentos de Programação (FP).

O projecto está dividido em 2 partes:

- **Primeira Parte** - apresentada no presente enunciado, incide nos menus e na introdução e validação dos dados iniciais do jogo;
- Segunda Parte - onde se implementará o jogo propriamente dito

As 2 partes são de entrega obrigatória e terão prazos de entrega distintos. O não cumprimento dos prazos de entrega de qualquer uma das partes do projeto levará automaticamente à reprovação dos alunos na avaliação de primeira época da componente prática.

Descrição do Projeto

O projeto encontra-se descrito em vídeo:

<https://www.youtube.com/watch?v=y2XhzFNo-f4>

Poderão ser identificados erros/omissões no vídeo. Devem estar atentos à errata que está publicada no primeiro comentário do 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 - Primeira Parte

O projeto deverá utilizar apenas as instruções ensinadas nas aulas até à aula 9. **Nomeadamente não deverão ser utilizados ciclos for, arrays, indexOf(), contains(), split(), listas, maps, Pairs, break, continue, exitProcess, etc.** Não deverão criar classes novas. Projetos que usem estas instruções vão ter fortes penalizações. Na dúvida, deverão contactar o vosso professor das aulas práticas.

A versão do Kotlin ensinada nas aulas e que é oficialmente suportada pelo Drop Project é a versão 2.2.

Formação dos grupos

Os projetos devem ser realizados em grupos de 2 alunos. Em situações excecionais poderão ser aceites grupos de um aluno — quem o pretender fazer deve efetuar um pedido (justificado) aos docentes da cadeira através de e-mail. **Alunos que entreguem individualmente sem terem um pedido aceite por um professor terão nota zero.**

Os elementos do grupo devem pertencer à mesma turma prática. Isto facilita o acompanhamento do projeto por parte do professor respetivo.

Os grupos de alunos definidos para a primeira parte do projeto devem ser mantidos para a segunda parte do projeto. Não serão permitidas entradas de novos alunos nos grupos entre as duas partes do projeto.

É obrigatória a inscrição prévia dos grupos, até dia 16 de Novembro, às 23h59. A inscrição é feita através do Moodle, no link “Grupos para o projeto” [[link direto](#)]. Só terão acesso ao Drop Project após inscrição no Moodle (poderá demorar até 24 horas a ser atribuído o acesso). Quando submeterem no Drop Project, **devem indicar no AUTHORS.txt os dois elementos do grupo.** Exemplo:

```
a22200000;Eddie Vedder  
a22200001;Chris Cornell
```

Metodologia Recomendada

Os testes do Drop Project fazem 2 tipos de validação: testes de função e testes interativos.

Os testes de função validam que as funções obrigatórias estão bem implementadas, executando diretamente essas funções com parâmetros diferentes e verificando que o retorno é o esperado.

Os testes interativos validam um fluxo completo do jogo, desde que mostra o menu, escolhe uma opção, vai respondendo às questões colocadas até que termina mostrando o tabuleiro. Ou seja, testam a execução do `main()`

A nossa sugestão é que se concentrem primeiro em passar os testes de função (identificados no relatório do Drop Project com `TestTeacherFunctions`) e só depois se preocupem com os testes interativos (identificados no relatório do Drop Project com `TestTeacherInteractive`).

A metodologia a seguir deve ser então:

1 - Comecem por implementar todas as funções obrigatórias de forma “*dummy*”, isto é, com uma implementação mínima que compile (ex: a função retornar apenas uma String vazia). Submetam no Drop Project e verifiquem se não há erros de compilação. Deixem a função `main()` vazia.

2 - Vão implementando as funções obrigatórias (agora de forma correta), uma a uma. Para cada função que implementarem, comecem por testá-la no vosso computador. Para isso, devem chamar a vossa função na `main()` e imprimir o resultado, para ver se está correta. Por exemplo, para testarem a função `validaNumeroDeMinas()`, podem fazer algo deste género:

```
fun main() {  
    println(validaNumeroDeMinas(3, 3, 2))    // deve escrever true  
    println(validaNumeroDeMinas(3, 3, 20))   // deve escrever false  
}
```

3 - Após testarem no vosso computador, submetam ao Drop Project, para validar que realmente a função está a funcionar bem (os testes relacionados com essa função já deverão passar). Vão fazendo isso função a função, até passarem todos os testes de função.

4 - Neste ponto, deverá estar a faltar apenas a implementação do jogo propriamente dito, em que o jogador vai introduzindo informações e o programa vai reagindo. Essa implementação deve ser feita na `main()`, usando as funções que desenvolveram previamente e funções auxiliares que decidam implementar. As funções auxiliares podem fazer `readln()`'s e `println`'s. Pensem que já têm algumas peças aparentemente desconexas de um puzzle, que vão agora montar.

5 - Com o `main()` implementado, devem agora concentrar-se em passar os testes interativos. Mas devem sempre testar primeiro no vosso computador. Por exemplo, repitam no vosso programa os exemplos apresentados neste enunciado e verifiquem se o output produzido coincide exatamente com os exemplos. Basta um espaço a mais para falharem no teste.

Entrega

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

<https://deisi.ulusofona.pt/drop-project/upload/fp-2526-projeto-p1>

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.

Prazo de entrega

A data limite de entrega é o dia **24 de Novembro de 2025**, pelas **10h00 da manhã** (hora de Lisboa, Portugal).

Os alunos que entreguem depois do prazo, ainda durante o dia 24 de Novembro de 2025, até às 23h59, **terão uma penalização de 5 valores** na nota final desta parte do projeto.

Não serão aceites entregas após dia 24 de Novembro de 2025 às 23h59. Nesse caso os alunos terão nota zero no projeto, **reprovando na componente prática da disciplina (1ª época)**.

Avaliação

A avaliação do projeto será feita considerando as entregas feitas pelos alunos em ambas as partes. A nota será divulgada no final de cada entrega.

Após a entrega final, será atribuída ao projeto uma nota final quantitativa, que será calculada considerando a seguinte fórmula:

$$0.3 * \text{NotaParte1} + 0.7 * \text{NotaParte2}$$

Existe uma nota mínima de 5 valores na parte 1 do projeto. Grupos que não consigam atingir esta nota estão automaticamente reprovados em época normal, tendo que entregar o projeto em época de recurso.

Cotações da Primeira Parte

A avaliação do projeto 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, funções que não sejam demasiado grandes, etc.).	3
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

Cópias

Trabalhos que sejam identificados como cópias serão anulados e os alunos que os submetam terão nota zero (0).

Uma cópia numa das entregas intermédias afastará os alunos (pelo menos) da restante avaliação de primeira época, podendo inclusivamente ter efeitos nas restantes épocas.

Notem que, em caso de cópia, serão penalizados quer os alunos que copiarem, quer os alunos que deixarem copiar.

Nesse sentido, recomendamos que não partilhem o código do vosso projecto (total ou parcialmente). Se querem ajudar colegas em dificuldade, expliquem-lhes o que têm que fazer, não lhes forneçam o vosso código, pois assim eles não estão a aprender!

A decisão sobre se um trabalho é uma cópia cabe exclusivamente aos docentes da unidade curricular.

Outras informações relevantes

— Recomenda-se que eventuais dúvidas sobre o enunciado sejam esclarecidas (o mais depressa possível) no **zulip** de Fundamentos de Programação 2025/2026

— Grupos que não entreguem a primeira parte ou tenham nota zero na mesma (seja por cópia, não compilação ou outra das situações indicadas no enunciado), não podem entregar a segunda parte do projeto, e reprovam automaticamente na componente prática da disciplina (de primeira época)

— Existirá uma defesa presencial e individual do projeto. A defesa será feita após a segunda entrega. Durante esta defesa individual, será solicitado ao aluno que faça alterações ao código do projeto, de forma a dar resposta a alterações aos requisitos.

— É possível que sejam feitas pequenas alterações a este enunciado, durante o tempo de desenvolvimento do projeto. Por esta razão, os alunos devem estar atentos ao **Moodle**.

— Eventuais situações omissas e/ou imprevistas serão analisadas caso a caso pelos docentes da cadeira.

FIM