

# Programação Orientada por Objetos

## BlockuDoku

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

Ano Letivo: 2019/2020

Época Normal e de Recurso

## Índice

1. Introdução.....	1
2. Funcionalidades .....	2
2.1. O Jogo.....	2
2.1.1. Blocos do jogo .....	2
2.1.2. Rodada .....	3
2.1.3. Sistema de pontuação.....	6
2.2. Jogadores .....	6
2.3. Desenrolar do jogo em modo consola .....	7
2.3.1. Menu da aplicação .....	11
3. Fases de desenvolvimento e entrega .....	12
4. Implementação e codificação .....	12
5. Constituição de grupos .....	12
6. Entrega do projeto .....	12
7. Regras e Critérios de Avaliação do Projeto .....	14
7.1. Regras de Avaliação .....	14
7.2. Critérios de Avaliação .....	14
8. Resumo das Datas Importantes .....	15
8.1. Entrega da 1ª fase .....	15
8.2. Avaliações da 1ª fase .....	15

## 1. Introdução

O objetivo deste projeto passa pelo desenvolvimento, utilizando a linguagem Java e a Programação Orientada por Objetos (POO), de uma versão de um jogo de blocos chamado **BlockuDoku**.

O jogo **BlockuDoku** consiste numa mistura de Sudoku com jogo de blocos em que o principal objetivo do jogo passa por efetuar o preenchimento de linhas, colunas ou quadrados, mantendo o tabuleiro com o mínimo de blocos possíveis.

O projeto será desenvolvido em duas fases. A primeira fase consiste na modelação e implementação de um conjunto de classes que permitam representar a lógica da aplicação e que permitam a sua utilização em modo **consola**. A segunda fase será dedicada à criação da **interface gráfica** com o utilizador.

Tenha particular atenção ao uso do paradigma de POO na modelação das classes, i.e., ao correto uso dos conceitos de encapsulamento, herança, classes abstratas, polimorfismo, interfaces, maximização da coesão (responsabilidade única), minimização do acoplamento, desenho orientado por responsabilidades, etc. Uma modelação bem pensada facilitará todo o processo de desenvolvimento e manutenção.

O presente documento versa apenas a primeira fase do projeto. Em data oportuna, após entrega da primeira fase, será apresentado o descritivo da segunda fase.

## 2. Funcionalidades

### 2.1. O Jogo



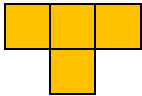
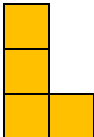
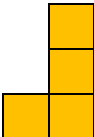
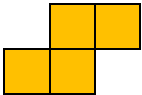
O jogo desenrola-se num tabuleiro de 9x9, um tabuleiro semelhante ao utilizado no jogo Sudoku. Ao jogador são disponibilizados blocos que deve encaixar nas zonas vazias do tabuleiro. Sempre que um conjunto de blocos formam uma linha completa, uma coluna completa e/ou um grande quadrado completo (entende-se por grande quadrado por exemplo o conjunto de quadrados A1:C3 ou D1:F3, etc...) estes quadrados são limpos e a pontuação do jogador sobe.

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

#### 2.1.1. Blocos do jogo





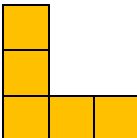
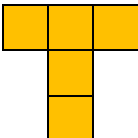
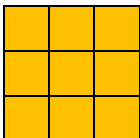
O jogo deverá ter um **modo básico** e um **modo avançado**, escolhido pelo utilizador no início de cada jogo. Consoante o modo escolhido pelo utilizador serão os blocos disponibilizados.

##### Blocos do modo básico:

<p>Bloco I</p> 	<p>Bloco Q</p> 	<p>Bloco T</p> 
<p>Bloco L</p> 	<p>Bloco J</p> 	<p>Bloco S</p> 

<p>Bloco Z</p> 		
--------------------------------------------------------------------------------------------------	--	--

**Blocos do modo avançado:**

<p>Bloco I1</p> 	<p>Bloco I2</p> 	<p>Bloco I3</p> 
<p>Bloco L - mínimo</p> 	<p>Bloco L - máximo</p> 	<p>Bloco T Estendido</p> 
<p>Bloco Q estendido</p> 		

No modo avançado do jogo devem ser disponibilizados os blocos dos dois modos do jogo.

**2.1.2. Rodada**

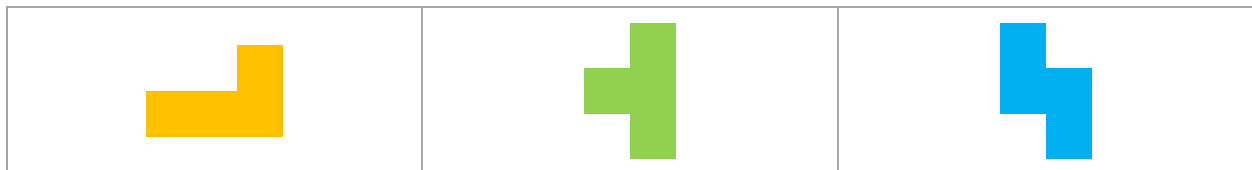
Cada rodada é composta por 3 jogadas, em que cada jogada consiste em colocar um bloco no tabuleiro.

São disponibilizados, de forma aleatória, 3 blocos por rodada, sendo que cada bloco deverá, também de forma aleatória, apresentar uma rotação de 0º, 90º, 180º ou 270º quando for fornecido ao jogador.

**Exemplo:****Rodada 1:**

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

Blocos disponibilizados:



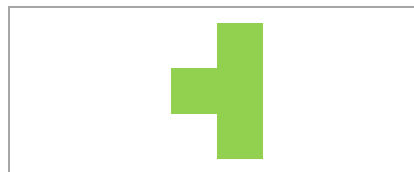
**Jogada 1**

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									



**Jogada 2**

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

**Jogada 3**

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

Após a colocação de cada bloco, o jogo, deverá verificar se foram preenchidas:

- Na totalidade uma ou mais linhas
- Na totalidade uma ou mais colunas
- Na totalidade um ou mais dos 9 grandes quadrados do tabuleiro

- A1:C3; D1:F3; G1:I3; A4:C6; D4:F6; G4:I6; A7:C9; D7:F9; G7:I9

Se uma ou mais premissas indicadas atrás foram satisfeitas, então cada quadrado do elemento preenchido deve ser limpo no tabuleiro e consequentemente o espaço ficará disponibilizado para receber novo bloco.

**Contudo deverá sempre mostrar no ecrã todas as iterações do tabuleiro, mostrando os quadrados preenchidos e depois limpos.**

Após a colocação dos três blocos o jogo segue para a rodada seguinte, e assim sucessivamente.

Em cada jogada deverá existir no tabuleiro espaço para pelo menos um dos blocos disponíveis, caso contrário o **jogo termina**.

### 2.1.3. Sistema de pontuação

A pontuação do jogador aumenta sempre que é colocado um bloco no tabuleiro ou algum quadrado do tabuleiro é limpo.

Sempre que um bloco é colocado no tabuleiro a pontuação aumenta de acordo com o número de quadrados que o bloco ocupa, tendo em consideração o seguinte:

- Bloco do modo básico: 1 ponto por quadrado
- Bloco do modo avançado: 2 pontos por quadrado

Por sua vez, sempre que se verificar a libertação de algum quadrado:

- Por cada quadrado limpo: 4 pontos
- Bónus por cada grande quadrado limpo do tabuleiro: 10 pontos

## 2.2. Jogadores

O jogo é jogado apenas por um jogador, contudo deve ser possível:

1. Registrar o nome do jogador que se encontra a jogar
2. Registrar o histórico de pontuações, com a respetiva data e hora do jogo, de todos os jogos efetuados pelo jogador
3. Mostrar um ranking do TOP 10, de todos os jogadores, de forma decrescente, com a melhor pontuação de cada um.



4. Deverá ser possível gravar um jogo a meio e continuar um jogo previamente gravado.

Salienta-se que tanto o jogo como os jogadores são “pontuáveis”. Devem ser definidos comportamentos comuns para estas duas entidades, como: adicionar pontos e obter total de pontos.

### 2.3. Desenrolar do jogo em modo consola

No desenrolar do jogo em modo consola o tabuleiro deve ser apresentado do seguinte modo:

```

      |A|B|C|D|E|F|G|H|I
1|.|.|.|.|.|.|.|.|.
2|.|.|.|.|.|.|.|.|.
3|.|.|.|.|.|.|.|.|.
4|.|.|.|.|.|.|.|.|.
5|.|.|.|.|.|.|.|.|.
6|.|.|.|.|.|.|.|.|.
7|.|.|.|.|.|.|.|.|.
8|.|.|.|.|.|.|.|.|.
9|.|.|.|.|.|.|.|.|.

Blocos a jogar:
Bloco A
  #
###

Bloco B
#
##
#

Bloco C
#
##
#

Indique a próxima jogada (Bloco-ColunaLinha): _

```

Para jogar o jogador deve escolher o bloco, a coluna e a linha onde vai colocar o bloco.

Exemplo:

	A	B	C	D	E	F	G	H	I
1	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	.

Blocos a jogar:

Bloco A

```

#
###

```

Bloco B

```

#
##
#

```

Bloco C

```

#
##
#

```

Indique a próxima jogada (Bloco-ColunaLinha): **A-G9**

A posição do bloco é definida pelo seu quadrado mais à esquerda e topo. Neste exemplo:

<b>A</b> – Selecionado o bloco A	# ###
<b>G9</b> – O quadrado mais esquerda e acima do bloco vai ser colocado nesta posição	# ###

	A	B	C	D	E	F	G	H	I
1	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.	#
9	.	.	.	.	.	.	.	#	#

Blocos a jogar:

Bloco B

```
#
##
#
```

Bloco C

```
#
##
#
```

Indique a próxima jogada (Bloco-ColunaLinha): **B-H5**

**B** – Selecionado o bloco B

```
#
##
#
```

**H5** – O quadrado mais esquerda e acima do bloco vai ser colocado nesta posição

```
#
##
#
```

	A	B	C	D	E	F	G	H	I
1	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	#
6	.	.	.	.	.	.	.	.	#
7	.	.	.	.	.	.	.	.	#
8	.	.	.	.	.	.	.	.	#
9	.	.	.	.	.	.	.	#	#

Blocos a jogar:

Bloco C

```

#
##
#

```

Indique a próxima jogada (Bloco-ColunaLinha): **C-H4**

**C** – Selecionado o bloco C

```

#
##
#

```

**H4** – O quadrado mais esquerda e acima do bloco vai ser colocado nesta posição

```

#
##
#

```

	A	B	C	D	E	F	G	H	I
1	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	#
4	.	.	.	.	.	.	.	#	#
5	.	.	.	.	.	.	.	#	#
6	.	.	.	.	.	.	.	#	#
7	.	.	.	.	.	.	.	#	#
8	.	.	.	.	.	.	.	#	#
9	.	.	.	.	.	.	.	#	#

Blocos a jogar:

(nova ronda de blocos)

Indique a próxima jogada (Bloco-ColunaLinha):

- Para cancelar o jogo o jogador deve indicar a instrução: CANCEL
- Para guardar o jogo em execução o jogador deve indicar a instrução: SAVE

Em qualquer dos casos anteriores o jogo deverá regressar ao menu da aplicação.

### 2.3.1. Menu da aplicação

A aplicação deverá apresentar o seguinte menu:

```
Olá <Nome do Jogador>

1 - Iniciar novo jogo
2 - Abrir jogo
3 - Mostrar pontuações pessoais
4 - Ranking (TOP 10)
0 - Sair
```

Ao iniciar novo jogo deverão aparecer as seguintes opções:

```
1 - Iniciar novo jogo - modo básico
2 - Iniciar novo jogo - modo avançado
0 - Voltar
```

### 3. Fases de desenvolvimento e entrega

O projeto está dividido em 2 fases, com a cotação distribuída da seguinte forma:

- Fase I – 60% da avaliação final
- Fase II – 40% da avaliação final

Conforme referido na ficha da disciplina, poderá alternativamente entregar o projeto numa só data, na época de recurso (sem possibilidade de incorporar componente de avaliação contínua), contemplando as duas fases do projeto.

### 4. Implementação e codificação

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação *camelCase* para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação *PascalCase* para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize o símbolo ‘\_’ nos identificadores (exceto nas constantes), nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

### 5. Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo eventualmente ser elaborado individualmente. Não serão permitidos, em nenhum caso, grupos com mais do que dois alunos.

Os grupos dos alunos já se encontram determinados através da metodologia de *pair programming* que está a ser utilizada nos laboratórios. Caso existam alunos que não têm o grupo escolhido, deverão contactar o respetivo docente de laboratório para regularizar a situação.

### 6. Entrega do projeto

O projeto será entregue em duas fases:

- Uma primeira fase (**até às 23:55:00 do dia 1 de junho de 2020**) com a implementação da lógica da aplicação (descrita no presente documento);
- A segunda fase (**até às 23:55:00 do dia 28 de junho de 2020**) com a parte gráfica (descrita em documento entregue futuramente).

O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica utilizando a área dos trabalhos no Moodle**. Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em **formato ZIP**. Em caso de dificuldades no acesso à plataforma Moodle, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.

**Não serão aceites quaisquer projetos entregues fora do prazo!**

Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- Um Manual Técnico onde conste uma breve descrição do programa, incluindo a explicação das classes/interfaces implementadas, principais atributos e métodos e suas relações.
- A documentação do programa em JavaDoc (não converta o documento gerado automaticamente em HTML para DOC!).
- O código fonte do programa na forma de projeto em *NetBeans* ou *IntelliJ*, com um *main* de testes a funcionar e com todas as funcionalidades implementadas.
- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura: `<curso>_<numAluno1>_<numAluno2>.zip`.

## 7. Regras e Critérios de Avaliação do Projeto

### 7.1. Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- Caso o aluno falte ao momento de supervisão, terá essa componente avaliada com zero valores.
- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da Programação Orientada por Objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento da linguagem Java.
- Serão premiadas a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respetiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão anulados.
- As avaliações da primeira fase do projeto serão realizadas na segunda semana de junho de 2020.

### 7.2. Critérios de Avaliação

A primeira fase do projeto será avaliada segundo os seguintes critérios:

<b>Funcionalidades</b>	<b>45%</b>
------------------------	------------



<b>Implementação</b>	<b>35%</b>
Estrutura de classes	20%
Conhecimento e boa utilização da linguagem	5%
Bom estilo (nomes, comentários, indentação)	5%
Definição de testes unitários	5%

<b>Documentação</b>	<b>10%</b>
JavaDOC	5%
Manual técnico	5%

<b>Avaliação qualitativa</b>	<b>10%</b>
------------------------------	------------

## 8. Resumo das Datas Importantes

### 8.1. Entrega da 1ª fase

A entrega da 1ª fase do projeto será até às **23:55:00 de segunda-feira, dia 1 de junho de 2020**.

### 8.2. Avaliações da 1ª fase

As avaliações da primeira fase do projeto serão realizadas na semana de **08 a 12 de junho de 2020**.