

## TRABALHO DA ÁREA 1: BHAGA-CHALL DISTRIBUÍDO EM JAVA RMI

### Objetivo

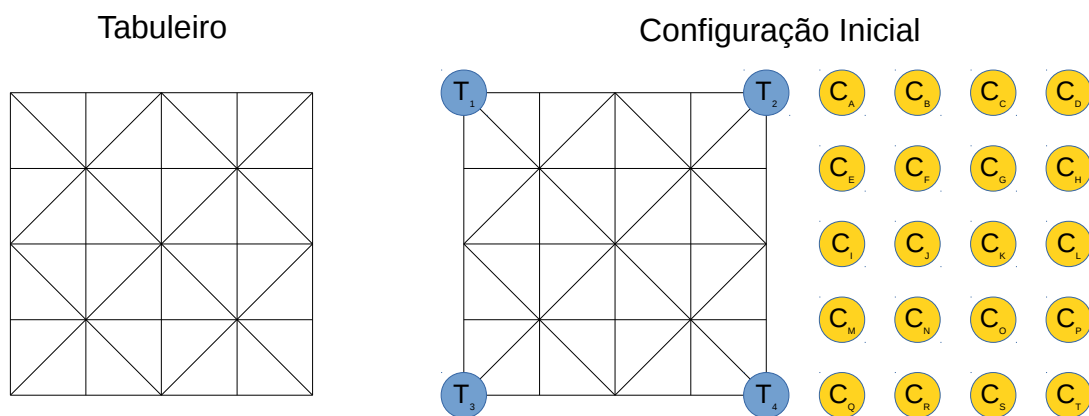
Implementar uma aplicação distribuída em Java RMI (*Remote Method Invocation*) que permita que usuários remotos disputem o jogo Bhaga-Chall (2016).

### Regras do Jogo

De acordo com o livro “Jogos de Todo Mundo” ([s.d.]), o Bhaga-Chall (ou Bhaga Chal) ilustra uma disputa entre tigres e cabras e foi criado no Nepal para simbolizar o equilíbrio entre qualidade (representada no jogo pelos tigres) e quantidade (representada no jogo pelas cabras). “Tigres são ágeis e poderosos, e qualquer descuido de seus adversários pode ser fatal. Por outro lado, as cabras andam em bandos e têm enorme facilidade para alcançar os mais difíceis refúgios.” (JOGOS, [s.d.], p. 21).

As regras a serem usadas neste trabalho são as regras descritas no verbete “Bhaga-Chall” da Wikipedia (BHAGA-CHALL, 2016).

Bhaga-Chall é disputado em um tabuleiro com 25 posições possíveis (5 casas por 5 casas, com determinado padrão de interconexão entre estas casas), tal como mostrado na figura a seguir. A disputa ocorre entre 4 tigres e 20 cabras. Inicialmente os 4 tigres são posicionados nos 4 cantos do tabuleiro e as cabras são posicionadas fora do tabuleiro.



O objetivo do jogador que joga com os tigres é capturar cinco cabras, enquanto o objetivo do jogador que joga com as cabras é cercar e imobilizar quatro tigres.

Depois que os tigres são posicionados nos 4 cantos do tabuleiro (conforme a configuração inicial apresentada na figura), o jogador com as cabras iniciará a partida colocando uma cabra em qualquer uma das interseções de linhas que esteja livre. A partir daí os jogadores alternam jogadas.

Tigres podem se deslocar uma casa (de uma casa até outra sempre percorrendo uma aresta) ou pular sobre uma cabra, desde que a próxima posição esteja livre, capturando-a (de forma semelhante ao que acontece no jogo de Damas, também fazendo o deslocamento da captura de acordo com as arestas). Tigres se movimentam seguindo as seguintes regras: podem capturar cabras em qualquer momento; devem capturar uma cabra sempre que for possível; podem capturar apenas uma cabra de

cada vez; devem pular sobre uma cabra em qualquer direção, desde que o pulo seja para uma interseção adjacente seguindo qualquer direção que parta da posição atual, capturando-a (ou seja, removendo-a do tabuleiro); não podem pular sobre outros tigres.

Até que o jogador que joga com as cabras tenha posicionado todas as 20 cabras, as cabras são apenas colocadas no tabuleiro uma em cada jogada. Depois que as 20 cabras tiverem sido colocadas no tabuleiro, as cabras passam a se mover, deslocando-se de uma casa até outra (sempre percorrendo uma aresta). As cabras movimentam-se seguindo as seguintes regras: são removidas do tabuleiro quando são capturadas; não podem pular sobre tigres ou outras cabras; somente podem movimentar-se depois de todas terem sido colocadas no tabuleiro.

As peças são sempre posicionadas nas interseções das linhas e os movimentos devem obrigatoriamente percorrer uma linha, sendo que a casa de destino deve estar livre.

## **Funcionamento da Aplicação**

O servidor deverá funcionar de modo que:

- sejam suportadas 50 partidas simultâneas de Bhaga-Chall entre 2 jogadores devidamente registrados (ou identificados) no servidor;
- quando um jogador se registra, ele deverá esperar que outro jogador também se registre (quando o próximo jogador se registrar, será formada uma dupla que disputará a próxima partida);
- o primeiro jogador a se registrar joga com as cabras e o segundo jogador com os tigres;
- responda a invocações remotas de métodos realizadas pelos clientes (conforme descrição de operações descrita a seguir);
- haja limites de tempo para determinados eventos: 2 minutos (120 segundos) pelo registro do segundo jogador, 30 segundos pelas jogadas de cada jogador, e 60 segundos depois de definido o vencedor para destruir a partida.

O cliente será responsável:

- pela interface com o usuário (que poderá ser tanto em modo texto quanto em modo gráfico);
- e por executar as invocações remotas de métodos disponíveis no servidor, de modo que os usuários possam jogar partidas consistentes.

## **Operações**

As seguintes operações remotas deverão ser implementadas pelo servidor:

### **1) registraJogador**

Recebe: *string* com o nome do usuário/jogador

Retorna: id (valor inteiro) do usuário (que corresponde a um número de identificação único para este usuário durante uma partida), -1 se este usuário já está cadastrado ou -2 se o número máximo de jogadores tiver sido atingido

### **2) encerraPartida**

Recebe: id do usuário (obtido através da chamada registraJogador)

Retorna: -1 (erro), 0 (ok)

### **3) temPartida**

Recebe: id do usuário (obtido através da chamada registraJogador)

Retorna: -2 (tempo de espera esgotado), -1 (erro), 0 (ainda não há partida), 1 (sim, há partida e o jogador inicia jogando com cabras, identificadas, por exemplo, com letras de “A” até “T”) ou 2 (sim, há partida e o jogador é o segundo a jogar, com os tigres, identificados, por exemplo, com

dígitos de “1” até “4”)

4) ehMinhaVez

Recebe: id do usuário (obtido através da chamada registraJogador)

Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro), 0 (não), 1 (sim), 2 (é o vencedor), 3 (é o perdedor), 4 (houve empate), 5 (vencedor por WO), 6 (perdedor por WO)

5) obtemGrade

Recebe: id do usuário (obtido através da chamada registraJogador)

Retorna: *string* vazio em caso de erro ou *string* com a grade de jogo

A grade de jogo pode, por exemplo, ser uma cadeia de caracteres onde cada caracter corresponde a uma casa do tabuleiro, com: letras de 'A' até 'T' correspondendo a uma casa ocupada por uma cabra, dígitos de '1' a '4' correspondendo a uma casa ocupada por um tigre e '.' correspondendo a uma casa não ocupada. Para a posição inicial do tabuleiro, o *string* seria, por exemplo:

“1...2.....3...4”

Depois da cabra “A” colocada na posição central, do tigre “1” movimentado em direção a esta cabra e da cabra “B” colocada em um posição desta mesma diagonal, o *string* seria, por exemplo:

“...2.1.....A.....B.3...4”

6) moveTigre

Recebe: id do usuário (obtido através da chamada registraJogador), identificador do tigre a ser movimentado (valor de 0 até 3, inclusive, considerando que cada tigre estará em determinada posição do tabuleiro, conforme histórico do jogo) e direção relativa da jogada (0 = para direita; 1 = para direita/abaixo; 2 = para baixo; 3 = para esquerda/abaixo; 4 = para esquerda; 5 = para esquerda/acima; 6 = para cima; 7 = para direita/acima).

Retorna: 2 (partida encerrada, o que ocorrerá caso o jogador demore muito para enviar a sua jogada e ocorra o *time-out* de 30 segundos para envio de jogadas), 1 (tudo certo), 0 (movimento inválido), -1 (erro), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida), -3 (não é a vez do jogador), -4 (não está jogando com o animal correto) ou -5 (direção inválida).

7) posicionaCabra

Recebe: id do usuário (obtido através da chamada registraJogador) e posição do tabuleiro onde a cabra deve ser posicionada (x e y, ambos variando de 0 a 4, inclusive).

Retorna: 2 (partida encerrada, o que ocorrerá caso o jogador demore muito para enviar a sua jogada e ocorra o *time-out* de 30 segundos para envio de jogadas), 1 (tudo certo), 0 (movimento inválido), -1 (erro), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida), -3 (não é a vez do jogador), -4 (não está jogando com o animal correto) ou -5 (todas as cabras já foram posicionadas).

8) moveCabra

Recebe: id do usuário (obtido através da chamada registraJogador), identificador da cabra a ser movimentada (valor de 0 até 19, inclusive, considerando que cada cabra estará em determinada posição do tabuleiro, conforme histórico do jogo) e direção relativa da jogada (0 = para direita; 1 = para direita/abaixo; 2 = para baixo; 3 = para esquerda/abaixo; 4 = para esquerda; 5 = para esquerda/acima; 6 = para cima; 7 = para direita/acima).

Retorna: 2 (partida encerrada, o que ocorrerá caso o jogador demore muito para enviar a sua jogada e ocorra o *time-out* de 30 segundos para envio de jogadas), 1 (tudo certo), 0 (movimento inválido), -1 (erro), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida), -3 (não é a vez do jogador), -4 (não está jogando com o animal correto), -5 (ainda não está na

fase de movimentação de cabras) ou -6 (direção inválida).

#### 9) obtémOponente

Recebe: id do usuário (obtido através da chamada registraJogador)

Retorna: *string* vazio para erro ou *string* com o nome do oponente

#### Sobre a movimentação no tabuleiro

O quadro a seguir mostra, para cada linha e coluna, quais as direções possíveis ou não na movimentação simples (cabras ou tigres fazendo apenas deslocamento sem captura). Neste quadro, direções assinaladas em laranja são impossíveis. Para as direções em verde apresenta-se também o número da linha e coluna da nova posição, caso esta direção seja selecionada, no formato: <direção>:(linha nova,coluna nova).

	0			1			2			3			4		
0	5	6	7	5	6	7	5	6	7	5	6	7	5	6	7
	4	?	0:(0,1)	4:(0,0)	?	0:(0,2)	4:(0,1)	?	0:(0,3)	4:(0,2)	?	0:(0,4)	4:(0,3)	?	0
	3	2:(1,0)	1:(1,1)	3	2:(1,1)	1	3:(1,1)	2:(1,2)	1:(1,3)	3	2:(1,3)	1	3:(1,3)	2:(1,4)	1
1	5	6:(0,0)	7	5:(0,0)	6:(0,1)	7:(0,2)	5	6:(0,2)	7	5:(0,2)	6:(0,3)	7:(0,4)	5	6:(0,4)	7
	4	?	0:(1,1)	4:(1,0)	?	0:(1,2)	4:(1,1)	?	0:(1,3)	4:(1,2)	?	0:(1,4)	4:(1,3)	?	0
	3	2:(2,0)	1	3:(2,0)	2:(2,1)	1:(2,2)	3	2:(2,2)	1	3:(2,2)	2:(2,3)	1:(2,4)	3	2:(2,4)	1
2	5	6:(1,0)	7:(1,1)	5	6:(1,1)	7	5:(1,1)	6:(1,2)	7:(1,3)	5	6:(1,3)	7	5:(1,3)	6:(1,4)	7
	4	?	0:(2,1)	4:(2,0)	?	0:(2,2)	4:(2,1)	?	0:(2,3)	4:(2,2)	?	0:(2,4)	4:(2,3)	?	0
	3	2:(3,0)	1:(3,1)	3	2:(3,1)	1	3:(3,1)	2:(3,2)	1:(3,3)	3	2:(3,3)	1	3:(3,3)	2:(3,4)	1
3	5	6:(2,0)	7	5:(2,0)	6:(2,1)	7:(2,2)	5	6:(2,2)	7	5:(2,2)	6:(2,3)	7:(2,4)	5	6:(2,4)	7
	4	?	0:(3,1)	4:(3,0)	?	0:(3,2)	4:(3,1)	?	0:(3,3)	4:(3,2)	?	0:(3,4)	4:(3,3)	?	0
	3	2:(4,0)	1	3:(4,0)	2:(4,1)	1:(4,2)	3	2:(4,2)	1	3:(4,2)	2:(4,3)	1:(4,4)	3	2:(4,4)	1
4	5	6:(3,0)	7:(3,1)	5	6:(3,1)	7	5:(3,1)	6:(3,2)	7:(3,3)	5	6:(3,3)	7	5:(3,3)	6:(3,4)	7
	4	?	0:(4,1)	4:(4,0)	?	0:(4,2)	4:(4,1)	?	0:(4,3)	4:(4,2)	?	0:(4,4)	4:(4,3)	?	0
	3	2	1	3	2	1	3	2	1	3	2	1	3	2	1

Observação: este quadro serve como exemplo e seus dados/conteúdos devem ser rigorosamente verificados antes de serem usados em qualquer implementação

Por sua vez, o quadro a seguir mostra, para cada linha e coluna, quais as direções possíveis ou não na movimentação realizando um pulo com captura (válido apenas para tigres). Neste quadro, direções em assinaladas em laranja são impossíveis. Para as direções em verde apresenta-se também o número da linha e coluna da nova posição, caso esta direção seja selecionada, no formato: <direção>:(linha nova,coluna nova).

	0			1			2			3			4		
0	5	6	7	5	6	7	5	6	7	5	6	7	5	6	7
	4	?	0:(0,2)	4	?	0:(0,3)	4:(0,0)	?	0:(0,4)	4:(0,1)	?	0	4:(0,2)	?	0
	3	2:(2,0)	1:(2,2)	3	2:(2,1)	1	3:(2,0)	2:(2,2)	1:(2,4)	3	2:(2,3)	1	3:(2,2)	2:(2,4)	1
1	5	6	7	5	6	7	5	6	7	5	6	7	5	6	7
	4	?	0:(1,2)	4	?	0:(1,3)	4:(1,0)	?	0:(1,4)	4:(1,1)	?	0	4:(1,2)	?	0
	3	2:(3,0)	1	3	2:(3,1)	1:(3,3)	3	2:(3,2)	1	3:(3,1)	2:(3,3)	1	3	2:(3,4)	1
2	5	6:(0,0)	7:(0,2)	5	6:(0,1)	7	5:(0,0)	6:(0,2)	7:(0,4)	5	6:(0,3)	7	5:(0,2)	6:(0,4)	7
	4	?	0:(2,2)	4	?	0:(2,3)	4:(2,0)	?	0:(2,4)	4:(2,1)	?	0	4:(2,2)	?	0
	3	2:(4,0)	1:(4,2)	3	2:(4,1)	1	3:(4,0)	2:(4,2)	1:(4,4)	3	2:(4,3)	1	3:(4,2)	2:(4,4)	1
3	5	6:(1,0)	7	5	6:(1,1)	7:(1,3)	5	6:(1,2)	7	5:(1,1)	6:(1,3)	7	5	6:(1,4)	7
	4	?	0:(3,2)	4	?	0:(3,3)	4:(3,0)	?	0:(3,4)	4:(3,1)	?	0	4:(3,2)	?	0
	3	2	1	3	2	1	3	2	1	3	2	1	3	2	1
4	5	6:(2,0)	7:(2,2)	5	6:(2,1)	7	5:(2,0)	6:(2,2)	7:(2,4)	5	6:(2,3)	7	5:(2,2)	6:(2,4)	7
	4	?	0:(4,2)	4	?	0:(4,3)	4:(4,0)	?	0:(4,4)	4:(4,1)	?	0	4:(4,2)	?	0
	3	2	1	3	2	1	3	2	1	3	2	1	3	2	1

Observação: este quadro serve como exemplo e seus dados/conteúdos devem ser rigorosamente verificados antes de serem usados em qualquer implementação

## Avaliação

Trabalhos com trechos copiados integralmente ou parcialmente serão avaliados com a nota mínima (ZERO). Os demais trabalhos serão avaliados numa escala de 0 (ZERO) até 10 (DEZ), levando em consideração as características descritas neste documento. Serão utilizados os seguintes pesos nesta avaliação:

- 40%: a aplicação executou corretamente sem erros, apresentando o comportamento esperado, conforme as regras do jogo Bhaga-Chall;
- 20%: todas as particularidades (tais como número de partidas e usuários, etc.) definidas neste documento foram implementadas.
- 20%: o aluno usou padrões de programação adequados (programação estruturada, nomes de variáveis significativos, comentários, etc.);
- 10%: usou bloqueios para proteger variáveis compartilhadas contra inconsistências referentes a acesso concorrente;
- 10%: foi implementado um mecanismo de temporização que funciona corretamente?

## Entrega

O trabalho deve ser desenvolvido individualmente.

A data de entrega do trabalho é **11 de outubro de 2016**.

Cada aluno deverá entregar todos os arquivos com extensão “.java” necessários para compilar e executar o projeto. E também deverá apresentar a execução de sua aplicação para o professor.

Em caso de cópia de trabalhos serão avaliados com a nota mínima (zero).

## **REFERÊNCIAS**

**BAGHA-CHALL.** In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2016.  
Disponível em: <<https://pt.wikipedia.org/w/index.php?title=Bagha-Chall&oldid=46394300>>.  
Acesso em: 16 ago. 2016.

**JOGOS de Todo Mundo.** [Florianópolis]: SESC Santa Catarina, [s.d.]. 64 p.