

# Trabalho Avaliado: Minesweeper/Campo Minado

André Rauber Du Bois  
*dubois@inf.ufpel.edu.br*

Julho 2020

## 1 Minesweeper/Campo Minado

O primeiro trabalho da disciplina é um conjunto de exercícios, que quando resolvidos, implementam o jogo Campo Minado.

O jogo possui um tabuleiro quadrangular com várias posições, no nosso caso, como o jogo é em modo texto, as posições são acessadas por suas coordenadas  $x$  e  $y$ . Quando o jogador abre uma posição, se essa posição contém uma mina escondida, o jogo acaba. Se a posição aberta é adjacente a uma mina, a posição aberta irá apresentar o número de minas adjacentes a essa posição (uma mina pode estar em até 8 posições adjacentes à posição aberta: N, S, L, O, SE, NE, SO e NO). Se o jogador abre uma posição que não é adjacente à uma mina, então todas as posições adjacentes são abertas automaticamente (podemos dizer recursivamente nesse caso hehe) até que se encontre posições adjacentes à minas. O objetivo do jogador é ir abrindo as posições até que todas as posições ainda fechadas contenham somente minas. Para facilitar a implementação, não existe a opção de marcar minas como em algumas versões do jogo (fica a dica para quem quiser se aventurar :)). Aconselho a quem não conhece o jogo, que procure jogar um pouco para se familiarizar. Existem várias versões de apps gratuitos e também versões online. **Vários dos exercícios propostos para este trabalho, não precisam nenhum conhecimento a respeito do jogo para serem resolvidos.**

## 2 Representações

A ideia da solução proposta é representar o estado do jogo como duas matrizes (que são representadas por listas de listas). Uma é uma matriz cujas posições são caracteres e que representa o tabuleiro do jogo (tipo **GBoard**) e a outra matriz contém booleanos indicando se aquela posição do jogo possuí uma bomba ou não (tipo **MBoard**):

```
type GBoard = [[Char]]  
type MBoard = [[Bool]]
```

Exemplo de tabuleiro de jogo todo fechado (9x9):

```
gBoard :: GBoard
gBoard = [['-', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '],
          ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ', '- ']]
```

No mapa do jogo, as posições fechadas são representadas pelo caractér '–', e posições abertas sem minas adjacentes são representadas pelo zero ('0'). Exemplo do tabuleiro com a disposição das minas:

```
mBoard :: MBoard
mBoard =
  [[False, False, False, True, True, False, False, False, False],
   [False, False, False, False, True, False, True, False, False],
   [False, False, False, False, False, False, False, False, False],
   [False, True, False, False, False, False, False, False, False],
   [False, True, False, False, True, False, False, False, False],
   [False, False, False, False, False, True, False, False, False],
   [True, False, False, False, False, True, False, False, False],
   [False, False, False, False, False, False, True, False, False],
   [False, True, False, True, False, False, False, False, False]]
```

Neste exemplo, podemos ver que as posições (3,1) e (0,3) possuem minas. As posições começam em zero e vão até **tamanho -1**.

Os exercícios são divididos em três grupos: funções que facilitam o acesso às matrizes, funções que contém a lógica do jogo e funções da interface do jogo para o jogador.

### 3 Formato das Soluções e Avaliação

Como o trabalho é um conjunto de exercícios, o aluno não precisa resolver todos para ganhar nota. Os exercícios devem ser resolvidos no arquivo **Minesweeper.hs** disponível no Moodle. O arquivo fornece o protótipo das funções que devem ser implementadas (nome e tipo) além de comentários com explicações. **Qualquer dúvida sobre as funções a serem implementadas, favor usar o respectivo forum.** Tentei quebrar o problema do Campo Minado em uma série de exercícios independentes. É claro que muitas vezes tentei seguir uma abordagem *bottom up*, ou seja, começamos com problemas mais simples que

ajudam a resolver os problemas mais complexos. Porém existem também vários problemas que são completamente independentes, por exemplo, imprimir o tabuleiro inicial, é completamente independente das regras do jogo. Da mesma forma, o aluno também pode quebrar uma solução em várias funções. O aluno deve entregar o arquivo **Minesweeper.hs** com todas as funções que conseguiu implementar. O arquivo deve carregar no ghci, ou seja, só entregue exercícios que realmente compilam no interpretador sem mensagens de erro. Se existe algum exercício que você tentou resolver mas não conseguiu terminar, e este gera mensagens de erro, deixe esse exercício comentado no código. Por favor, seguir também as outras recomendações de sempre: não usar tabulação para identar, não modificar os nomes das funções, não enviar arquivos compactados, etc. O trabalho vale **5.0** na primeira avaliação. Os outros **5.0** pontos serão dados pelas listas de exercícios.

## 4 Exemplo de interface para o jogo

Esses são exemplos da jogabilidade usando a interface simples que eu fiz. Você pode usar a imaginação para fazer algo bem melhor. O motor do jogo está comentado, só deve ser descomentado quando as respectivas funções do jogo estiverem implementadas.

Exemplo de jogo:

```
*Main> main
Digite o tamanho do tabuleiro: 9
0 1 2 3 4 5 6 7 8
0 - - - - - - - -
1 - - - - - - - -
2 - - - - - - - -
3 - - - - - - - -
4 - - - - - - - -
5 - - - - - - - -
6 - - - - - - - -
7 - - - - - - - -
8 - - - - - - - -
Digite uma linha: 3
Digite uma coluna: 3
0 1 2 3 4 5 6 7 8
0 - - - - - - - -
1 - - - - - - - -
2 - - - - - - - -
3 - - 1 - - - - -
4 - - - - - - - -
5 - - - - - - - -
6 - - - - - - - -
7 - - - - - - - -
```

```
8 - - - - -  
Digite uma linha: 4  
Digite uma coluna: 4  
0 1 2 3 4 5 6 7 8  
0 - - - - - - -  
1 - - - - - - -  
2 - - - - - - -  
3 - - - 1 - - - -  
4 - - - 3 - - -  
5 - - - - - - -  
6 - - - - - - -  
7 - - - - - - -  
8 - - - - - - -  
Digite uma linha: 2  
Digite uma coluna: 2  
0 1 2 3 4 5 6 7 8  
0 - 1 0 0 1 - - -  
1 1 1 0 0 1 2 - - -  
2 0 0 0 0 0 1 - - -  
3 0 1 1 1 1 1 - - -  
4 0 1 - - 3 - - -  
5 0 1 - - - - - -  
6 1 1 - - - - - -  
7 - - - - - - -  
8 - - - - - - -  
Digite uma linha: 0  
Digite uma coluna: 0  
VOCE PERDEU!  
0 1 2 3 4 5 6 7 8  
0 * 1 0 0 1 * 2 1 0  
1 1 1 0 0 1 2 * 1 0  
2 0 0 0 0 0 1 1 1 0  
3 0 1 1 1 1 1 1 1 1  
4 0 1 * 3 3 * 1 1 *  
5 0 1 2 * * 3 2 2 1  
6 1 1 2 2 2 3 * 2 0  
7 2 * 2 0 0 2 * 2 0  
8 2 * 2 0 0 1 1 1 0  
TENTE NOVAMENTE!
```