

1 Instruções Importantes

Nesta seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem funcional Haskell. Qualquer biblioteca que simule estados, como a *Data.IORef* ou semelhantes não podem usadas. Trabalhos que usem tais bibliotecas serão desconsiderados.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- relatório do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código executa no *ghci*. Programas com erros de sintaxe receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo de relatório tenha a identificação do autor do trabalho.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, relatório e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

O relatório deve conter informações relevantes para auxiliar no entendimento da estratégia algorítmica adotada para solucionar o problema, como, por exemplo, os tipos de dados usados para representar / modelar a solução.

O trabalho deverá ser apresentado ao professor da disciplina e, só será avaliado após a realização da entrevista, i.e., trabalhos que não forem apresentados não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, documentação e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina.**

Atrasos serão penalizados por uma função exponencial de dias de atraso, i.e., será reduzido da nota um percentual referente a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

Dias de Atraso	Nota
1	$n \cdot 0.98$
2	$n \cdot 0.96$
3	$n \cdot 0.92$
4	$n \cdot 0.84$
5	$n \cdot 0.68$
6	$n \cdot 0.36$
7	0

2 O Jogo Mastermind

O jogo *mastermind* é um jogo de dois jogadores no qual um jogador (Jogador 1) escolhe um segredo de quatro dígitos, dentre dez possíveis, e o segundo jogador (Jogador 2) tenta adivinhar qual o segredo. A cada rodada, o Jogador 2 “chuta” um valor para o segredo e, então, o Jogador 1 informa quantos dígitos estão certos, na posição correta, quantos estão certos na posição errada e quantos estão errados. Com base nas informações fornecidas pelo Jogador 1, o Jogador 2 tenta um novo “chute”. Esse processo continua até que o Jogador 2 acerte o segredo ou o número de tentativas se esgote.

Por exemplo, suponha que o Jogador 1 escolhe o segredo *1234* e que o número máximo de tentativas para adivinhar o segredo pelo Jogador 2 será 4. Agora, suponhamos que, na primeira rodada, o Jogador 2 “chutar” o valor *3254*. Assim, o Jogador 1 deverá informar que o Jogador 2 acertou dois números na posição correta, um número na posição errada e os demais dígitos estão errados. Usaremos o símbolo “O” para denotar que um dígito está na posição correta, o símbolo “-” para informar um dígito na posição errada e o símbolo “X” para dizer que um dígito está errado. Usando essa notação, o Jogador 1 poderia dar a resposta *OXO-*. Note que a ordem da resposta não, necessariamente, deve seguir a mesma ordem dos dígitos dos chutes do Jogador 2. A Tabela 2 mostra uma sequência de chutes para o segredo *1234* e as respostas do Jogador 1.

Chute do Jogador 2	Resposta do Jogador 1
3254	OXO-
6674	XOXX
3256	OXX-
1234	OOOO

3 Especificação Técnica do Trabalho

O objetivo deste trabalho é implementar, em Haskell, o jogo *mastermind*. Ao iniciar o programa, o usuário irá digitar os chutes para o segredo. Após cada chute, o programa irá exibir na tela os símbolos indicando os acertos e erros, conforme as regras do jogo.

O sua implementação deverá atender aos seguintes critérios:

- A apresentação dos símbolos indicando os acertos deverá ser numa ordem aleatória e não deve seguir a disposição dos dígitos informados pelo usuário;
- O programa deverá aceitar somente o quatro dígitos como “chutes”, nem mais e nem menos;
- O programa deverá permitir jogar com a possibilidade de dígitos repetidos ou não;
- O usuário terá 8 tentativas para acertar o segredo. Caso não acerte dentro do número de tentativas, o programa informará que ele perdeu, exibirá o segredo na tela e encerrará.

4 Entrega do Trabalho e Apresentação

A data da entrega do trabalho será até o dia **29 de setembro de 2023**, via plataforma do Google-Classroom. A entrevista será agendada após a entrega do trabalho, podendo ser realizada presencialmente ou virtualmente pelo Meet, e deverá ser realizada até o dia **20 de novembro de 2023**.