

# Trabalho final: Caça-palavras

Prof. Raphael Bernardino

Introdução à Programação C/C++ (IC/UFRJ)

27 de junho de 2022

Este trabalho consiste na implementação de um jogo, utilizando a linguagem de programação C ou C++, para o curso de Introdução à Programação C/C++ da Universidade Federal do Rio Janeiro (UFRJ). O peso deste trabalho equivale a 40% da nota final.

## Instruções

- Este trabalho possui o número máximo de integrantes igual a 3. Em outras palavras, pode ser feito individualmente, em dupla ou trio.
- A cada semana as funções descritas na seção abaixo serão sorteadas entre os integrantes do grupo. No caso de trabalho individual, o integrante deverá entregar todas as funções pedidas.
- Não serão aceitos trabalhos idênticos (plágio). Qualquer vestígio de cópia, seja código retirado da internet ou copiado de algum outro colega, implicará em nota zero. No último caso, todos os envolvidos serão penalizados.
- Toda a interação ocorrerá através do *shell*. Portanto, não há necessidade de interfaces gráficas.
- O código deve estar bem organizado, limpo, bem documentado e, de preferência, modularizado.
- Seu trabalho não deve incluir conceitos ou bibliotecas que fogem do escopo deste curso. Caso você precise, ou queira utilizar, peça orientação para o professor ou monitor(es) da turma.
- O código deve ser enviado até a data estipulada no cronograma do curso disponibilizado no Google Sala de Aula na seção Informações Importantes, item Plano de Curso.
- Os trabalhos devem ser entregues no formato `[semanaN]_[DRE].c`. Por exemplo, o aluno com DRE igual a 102030405 deve enviar na primeira semana o trabalho nomeado como `semana1_102030405.c`.

## Caça-palavras

Este trabalho consiste em fazer um jogo de caça-palavras, onde o tabuleiro possui uma região NxN. Cada célula desta região deve conter, no máximo, uma letra (veja Figura 1). O programa deve ser capaz de buscar em arquivo local uma lista de palavras, posicionar no tabuleiro as palavras de forma adequada e permitir que o jogador indique as posições em que as palavras se encontram.

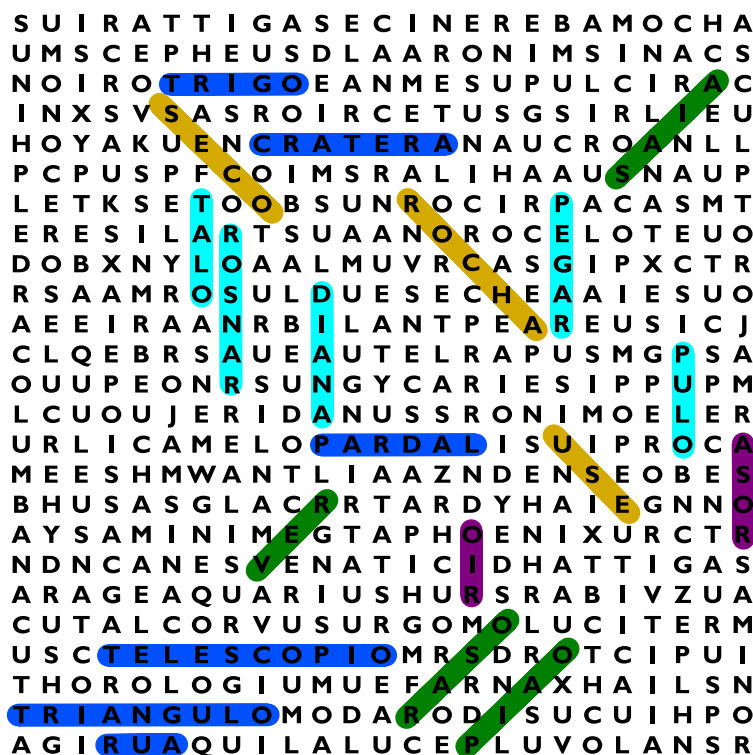


Figura 1: Exemplo de Caça-palavra 25x25 preenchido.

O objetivo do jogo é conseguir completar o tabuleiro seguindo as regras de:

- Encontrar todas as palavras, sendo possível que essas se encontrem na horizontal, vertical, e diagonais.
- Uma ou mais palavras podem possuir interseção de letras. Não existem palavras repetidas e todas são escritas em maiúscula.
- As palavras também não possuem acentuação, caracteres especiais e similares, como: hífen, acento agudo ou circunflexo, e cedilha.

Se o jogador realizar este feito, vence o jogo. Senão uma mensagem é exibida informando que a solução apresentada não satisfaz as condições estabelecidas.

## Entrega da primeira semana (01/07 - 08/07)

As funções a serem desenvolvidas nesta semana envolvem os itens abaixo. Caso seja necessário, crie funções auxiliares para cumprir os objetivos.

- Leitura do arquivo informado pelo usuário. Essa função obrigatoriamente deve ser chamada **le\_caca\_palavras(arquivo)**. Essa função possui como parâmetro o local do arquivo contendo o caça-palavras e a saída é o conteúdo do arquivo lido. Os valores são separados por uma quebra de linha. Em outras palavras, cada linha do arquivo corresponde a uma palavra a ser inserida no caça-palavras. Por exemplo, para gerar o tabuleiro da Figura 1 o arquivo seria escrito da seguinte forma:

```
trigo
saia
cratera
talo
diana
Pardal
RIO
telescopio
rua
Triangulo
rosa
seco
ROCHA
PeGaR
raso
piao
pulo
rosnar
ver
use
```

- Encontrar as letras que possuem interseção com as palavras. Essa função obrigatoriamente deve ser chamada **encontra\_intersecao(palavras)**. A entrada é o conteúdo do arquivo e a saída as interseções das palavras. Pode-se utilizar qualquer estrutura para cumprir este objetivo. Idealmente, converse com o seu grupo e discuta qual será escolhida.
- Determinar qual deve ser o tamanho do tabuleiro e criá-lo a partir da informação de interseção entre palavras. Essa função obrigatoriamente deve ser chamada **cria\_tabuleiro(conteudo)**. Essa função tem como entrada as informações de interseção e sua saída é o tabuleiro NxN totalmente vazio. Lembre-se que é possível posicionar palavras na horizontal, vertical e diagonais — em ambos os sentidos.

## Entrega da segunda semana (08/07 - 15/07)

As funções a serem desenvolvidas nesta semana envolvem os itens abaixo. Caso seja necessário, crie funções auxiliares para cumprir os objetivos.

- Impressão do tabuleiro para que o usuário visualize a situação atual do jogo. Essa função obrigatoriamente deve ser chamada **imprime(T)**, onde recebe como parâmetro o tabuleiro do jogo atual e não possui saída – toda informação será impressa na tela através do comando apropriado.
- Posicionar as palavras no tabuleiro usando a lista de interseções. Essa função obrigatoriamente deve ser chamada **posiciona\_palavras(T, I)**, onde recebe como parâmetro o tabuleiro T, e as informações de interseções I. Essa função possui como saída as posições de cada palavra utilizada, indicando a palavra, em qual linha e coluna se inicia, e sua orientação (horizontal, vertical, ou diagonal). As palavras são posicionadas alterando o tabuleiro diretamente.
- Verificar se o par posição e orientação corresponde à palavra existente no tabuleiro. Essa função obrigatoriamente deve ser chamada **verifica(X, P, K)**, onde recebe como parâmetro as posições de cada palavra utilizada X, as informações de posição P — contendo a linha, coluna e orientação — e o valor K que indica o tamanho da palavra. A saída é um booleano informando se foi possível encontrar ou não a palavra informada. Note que a posição informada pode ser inválida – uma posição maior que N ou menor que zero — e que o valor de K pode exceder o tamanho do tabuleiro ao considerar uma posição próxima às bordas do mesmo.

## Entrega da terceira semana (15/07 - 22/07)

As funções a serem desenvolvidas nesta semana envolvem os itens abaixo. Caso seja necessário, crie funções auxiliares para cumprir os objetivos.

- Marcar a palavra na horizontal. Essa função obrigatoriamente deve ser chamada **marca\_horizontal(T, P, K, C)**, onde recebe como parâmetro o tabuleiro do jogo atual T, a posição da palavra P, seu tamanho K e as informações sobre quais palavras já foram marcadas C. O parâmetro C deve ser alterado agrupando as informações.
- Marcar a palavra na vertical. Essa função obrigatoriamente deve ser chamada **marca\_vertical(T, P, K, C)**, onde recebe como parâmetro o tabuleiro do jogo atual T, a posição da palavra P, seu tamanho K e as informações sobre quais palavras já foram marcadas C. O parâmetro C deve ser alterado agrupando as informações.
- Marcar a palavra na diagonal. Essa função obrigatoriamente deve ser chamada **marca\_diagonal(T, P, K, C)**, onde recebe como parâmetro o tabuleiro do jogo atual T, a posição da palavra P, seu tamanho K e as

informações sobre quais palavras já foram marcadas C. O parâmetro C deve ser alterado agrupando as informações.

### Entrega da quarta semana (22/07 - 29/07)

As funções a serem desenvolvidas nesta semana envolvem os itens abaixo. Caso seja necessário, crie funções auxiliares para cumprir os objetivos.

- Imprima as palavras que ainda faltam ser marcadas no tabuleiro. Essa função obrigatoriamente deve ser chamada **imprime\_palavras(P, C)**, onde recebe como parâmetro as informações de palavras contendo as posições P e as palavras que já foram marcadas no tabuleiro C.
- Verifique a cada jogada se o tabuleiro está completo, ou seja, se todas as palavras foram marcadas no tabuleiro. Caso sim, informe que o jogador ganhou e encerre o jogo. Senão, imprima quais são as palavras a serem marcadas através da função **imprime\_palavras(P, C)**. Essa função deve ser chamada **verifica\_tabuleiro(P, C)**, onde recebe como parâmetro as informações de palavras contendo as posições P e as palavras que já foram marcadas no tabuleiro C.
- Crie uma função principal que irá orquestrar o jogo levando em consideração todas as ações disponíveis. Essa função deve se chamar obrigatoriamente de **inicia\_jogo()**. Essa função também é responsável por realizar a leitura dos dados digitados pelo usuário e gerenciar as chamadas para as funções — lendo o arquivo de palavras, montando o tabuleiro com as palavras, mostrando-o na tela e atualizando-o quando necessário.

Ao final informe ao jogador que ele ganhou. Dê também a opção de jogar novamente, gerando outro tabuleiro com uma configuração distinta da anterior.