

# Trabalho Final – Laboratório de Programação 1

O objetivo do trabalho é criar um corretor ortográfico. O corretor deve ser um programa de linha de comando que recebe dois parâmetros:

- um arquivo de texto a ser corrigido.
- um dicionário contendo palavras aceitas.

A partir do arquivo recebido, o corretor deve apresentar ao usuário as sugestões de correção, para as quais o usuário deve responder se aceita ou não. Ao final da verificação, a versão corrigida deve ser salva em um arquivo de saída, cujo nome é o arquivo original sufixado pela data/hora da correção.

Para cada sugestão apresentada, o corretor deve exibir as seguintes informações:

- o trecho a ser corrigido
- o trecho corrigido
- informações referentes à correção sugerida

O trecho é composto pelas palavras onde o possível problema foi encontrado e algumas palavras que aparecem antes e depois, para contextualização. O exemplo abaixo apresenta como a sugestão deve ser apresentada ao usuário.

**trecho original:** "Esta massa de ar seco e frio que está chegando, porém, não será de forte intensidade. muito pelo contrário, é fraca"

**trecho alterado:** "Esta massa de ar seco e frio que está chegando, porém, não será de forte intensidade. Muito pelo contrário, é fraca"

**Correção:** Todas as frases devem iniciar com letra maiúscula.

**Aplicar?** (s/n)

A seguir, as regras que deverão ser validadas pelo corretor:

## Tamanho das letras:

**Regra 1:** Todas as frases devem iniciar com letra maiúscula.

**Regra 2:** Em qualquer palavra, o tamanho da letra deve ser o mesmo para todas as letras. Em caso de inconformidade, sugerir como correção a opção usada na maioria das letras da palavra. A exceção são as palavras onde apenas a primeira letra está em maiúscula, pois elas provavelmente se referem a nomes pessoais. Exemplos de inconformidades: OVNi => OVNI, planeTa => planeta.

## **Uso de Parênteses/Chaves/Colchetes**

**Regra 3.** Sinais abertos e não fechados antes de um final de frase devem ser descartados. Da mesma forma, sinais fechados sem terem sido abertos devem ser descartados.

**Regra 4.** Caso haja um sinal de fechamento diferente de um sinal de abertura anterior, em vez de descartar ambos, pode-se sugerir que o sinal de fechamento seja modificado.

## **Uso de números:**

**Regra 5:** números naturais inferiores a 11 devem ser escritos por extenso.

## **Remoção de caracteres:**

**Regra 6:** Caracteres especiais ('@', '#', '\$', '%', '&', '\*') devem ser descartados

**Regra 7:** Excessos de espaços em branco consecutivos (mais do que um) devem ser descartados.

## **Correção de palavras:**

**Regra 8.** Apenas palavras presentes no dicionário devem ser aceitas. As demais devem ser descartadas. Exceção: acrônimos e nomes pessoais.

**Regra 9.** Se uma palavra não existir no dicionário, mas existir alguma que seja similar, deve-se sugerir a palavra similar para substituição. O sistema deve dar três sugestões, considerando as alternativas mais similares. Cabe ao aluno definir como as sugestões serão apresentadas.

**Regra 10.** Caso nenhuma palavra similar exista, o sistema deve perguntar se o usuário deseja acrescentar a palavra ao dicionário em vez de descartá-la. O arquivo contendo o dicionário deve ser atualizado para contemplar as palavras adicionadas. Cabe ao aluno definir como o sistema apresentará essa pergunta ao usuário.

**Obs.** Uma palavra A é considerada similar à uma palavra do dicionário B se a remoção de caracteres de A a deixa equivalente a B. Quanto menos caracteres precisaram ser removidos, mais similar é a palavra. Ex. 'teclXar' é similar a 'teclar'(uma remoção) e 'tecla'(duas remoções)

## **Cuidados:**

- As regras devem ser verificadas na ordem em que os problemas aparecem no texto
- Evite reprocessar todo o arquivo a cada situação verificada.
- Não use bibliotecas prontas para implementar a solução