O objetivo desse trabalho é definir uma função que, dado um documento, gera um índice das palavras que ocorrem nesse documento. O programa deve ter como entrada um arquivo texto e o índice deve ser impresso na tela. Considere as seguintes definições:

```
type Doc     = String
type Linha   = String
type Palavra = String
```

O índice de deve ser construído pela função construirIndice respeitando a assinatura de tipos abaixo, a leitura do arquivo texto e impressão na tela deve ser declarada na função main:

```
construirIndice :: Doc → [([Int],Palavra)]
```

O problema de gerar os índices pode ser dividido nos seguintes subproblemas:

a) Separar o documento em linhas: `lines :: Doc → [Linha]`

b) Numerar as linhas do documento: `numLinhas :: [Linha] → [(Int, Linha)]`

c) Associar a cada ocorrência de uma palavra do documento, o número da linha em que essa palavra ocorre: `numeraPalavras :: [(Int, Linha)] → [(Int, Palavra)]`

d) Ordenar alfabeticamente as ocorrências das palavras no texto:

   `ordenar :: [(Int, Palavra)] → [(Int, Palavra)]`

e) Juntar as várias ocorrências de cada palavra, produzindo, para cada palavra, a lista dos números das linhas em que a palavra ocorre:

   `agrupar :: [(Int, Palavra)] → [([Int], Palavra)]`

f) Eliminar, da lista de números de linhas em que cada palavra ocorre, as repetições de um mesmo número de linha:

   `eliminarRep :: [([Int], Palavra)] → [([Int], Palavra)]`

Observações:
As seguintes funções são definidas na biblioteca padrão de *Haskell*:

```
lines :: String → [String] -- Divide um texto em linhas
words :: String → [String] -- Divide uma linha em palavras
```

Antes de separar cada linha do seu texto em palavras, devem ser eliminados da linha os caracteres de pontuação, os quais não devem ser incluídos no índice (pesquisar as funções isAlpha, isSpace, isDigit, zip). Palavras com menos de 3 letras também devem ser eliminadas.