**TP 2 – Busca por palavras**

**Grupo:** Diego Polanski, Guilherme de Almeida e Raphael Arnault

Durante a execução do projeto, começamos apenas adaptando, para que a pesquisa pudesse ocorrer pelo título e autor, e não apenas pelo ID. Após sucesso nesse processo, pensamos em adicionar o split, para que a busca fosse realmente a partir dos termos.

A parte mais difícil do projeto foi entender como tratar os IDs duplicados que surgiam dependendo da busca. Para isso, transformamos a lista em uma tabela hash padrão, eliminando assim os itens duplicados.

Todos os resultados foram sim alcançados, de forma organizada no código.

Para executar o trabalho, foi necessário alterar todas as etapas do CRUD.

1. Criamos uma função para fazer o “split” do título e dos autores dos livros. Essa função retorna um array de Strings, com a divisão da frase.
2. **Create:**
3. No create, após o usuário preencher todos os dados do registro, chamamos a função do split, para o título e o autor.
4. Depois com um laço de repetição adicionamos cada palavra do array de strings na lista invertida.
5. Foram criadas 2 listas invertidas, uma para os títulos e outra para os autores.
6. **Read:**
   1. No momento da leitura, disponibilizamos 2 opções para o usuário. Pesquisar pelo título ou pelo autor.
   2. Após a escolha, a string é coletada e levada para uma função de leitura específica de cada modelo de procura.
   3. Então é levado para a função split.
   4. O resultado será levado para uma função que criará um array com todos os IDs que correspondem a busca das strings nas listas invertidas. (Tratamos para que não seja permitido IDs repetidos).
   5. É enviado, cada ID desse array, para o método read original, que procurará o registro.
   6. Então escrevemos na tela os dados de todos os registros que estão de acordo com a busca feita.
7. **Uptade:**
   1. Ao selecionar o uptade, o usuário vai selecionar qual id será atualizado.
   2. Depois irá selecionar qual será o item a ser atualizado, sendo ele, título, autor ou preço.
   3. O usuário irá colocar o novo valor.
   4. Será armazenado o antigo valor, para então deletar o dado na lista.
   5. O novo dado do livro é registrado, então esse novo valor é criado na lista.
8. **Delete:**
   1. É perguntado ao usuário qual id deseja deletar.
   2. Após informar, será armazenado o valor do título e do autor deste registro.
   3. Então será apagado das listas tais dados com o ID inserido pelo usuário.
9. OBS: As palavras sempre eram transformadas com letras minúsculas, nas listas invertidas, no momento da pesquisa e de atualizações para que todas operação ocorram corretamente, independente da forma como foi escrita.

**Questionário**

1. Sim, acrescenta cada termo do título e do nome do autor.
2. Sim, toda atualização também é alterada na lista. Apagando a relação do antigo nome com o ID atualizado, e criando a relação com o novo título.
3. Sim, todo registro apagado, também ocorre a remoção na lista.
4. Sim as stop words foram removidas de todo o processo, as palavras com menos de 4 letras, são removidas.
5. Além do mínimo, acrescentamos os autores nos registros.
6. Todas as funções estão ocorrendo corretamente.
7. Está completo com todos os requisitos necessários.
8. O trabalho é original.