**Relatório: TAD Biblioteca**

**Gabriel Doimo Filippi 9292625**

**Vitor Massuella Sobral 10368969**

**Considerações Iniciais**

O trabalho consistiu em implementar um sistema de controle de uma biblioteca, utilizando duas versões, uma totalmente estática e outra totalmente dinâmica. As dificuldades foram muitas, mas em especial, a harmonia entre as duas versões, sem conflitos na única main. A seguir nesse relatório, iremos detalhar como são as seis principais funções do sistema da biblioteca, que são: Cadastro de aluno, Cadastro de livro, Retirar livro, Retornar livro, remover livro, Remover aluno.

Obviamente foram criadas outras funções para auxiliar na execução do trabalho, entretanto essas não serão citadas aqui, pois possuem comentários e estão detalhadas dentro dos respectivos .c e .h.

**TAD**

**Dinâmica:** A implementação na forma dinâmica foi feita utilizando uma Lista Dinâmica Encadeada para representar a Biblioteca, sendo ela um conjunto de livros, apenas. Dentro de cada estrutura de livro, há informações referentes ao livro (strings dinâmicas em sua grande maioria) e também uma fila de espera, implementada na forma de fila utilizando uma Lista Dinâmica Ecadeada.

**Estática:** O sistema da biblioteca foi feito com o formato de banco de memória, tanto para os livros quanto para os alunos e fila de espera. Dentro da struct do aluno também tem o inteiro “topo” que controla a pilha de mensagens. A Fila de espera aponta para o aluno com um vetor de ponteiros. O TAD conta com um struct apenas para armazenar as mensagens do aluno, em forma de pilha.

**Cadastrar Aluno**

**Dinâmica:** A implementação dinâmica consistiu em receber o nome do aluno e um conjunto de informações deste. Tais informações foram armazenadas em uma estrutura do tipo Aluno e guardadas em uma lista. A inserção nessa lista foi sempre no fim, de forma a não se preocupar com a ordem dos alunos.

Vale salientar que a função não distingue o nome que está sendo inserido dos nomes que já estão cadastrados na lista. Sendo assim, se já houver um nome “João” na lista e o usuário tentar inserir um outro usuário com o mesmo nome “João”, o sistema permitirá e o interpretará como um usuário diferente.

**Estática:** Para cadastrar o aluno, verifica-se caso a lista esteja cheia ou o aluno já estava no sistema para cancelar o processo de cadastrar o aluno. Caso a lista de alunos esteja vazia. O cadastramento funciona copiando as informações que entram na função como ponteiros, e arruma a lista que está em formato de banco de memória.

**Cadastrar Livro**

**Dinâmica:** A implementação dinâmica consistiu em receber o nome do livro e um conjunto de informações deste. Tais informações foram armazenadas em uma estrutura do tipo Livro e guardadas em uma lista. A inserção nessa lista foi sempre no fim, de forma a não se preocupar com a ordem dos livros.

Vale salientar que a função é capaz de distinguir o nome do livro que está sendo inserido dos livros que já estão cadastrados na lista. Sendo assim, se já houver um livro com um determinado nome e o usuário efetuar um cadastro de um outro livro com o mesmo nome, apenas a quantidade do livro existente no cadastro será alterada, terminando a inserção.

**Estática:** Assim como na função cadastrar aluno, a função de cadastrar livro, verifica se a lista não está cheia ou se o livro já estava no sistema. Depois a função copia as informações passadas por referência para o livro na lista. Tudo ajeitando o banco de memória.

**Retirar Livro**

**Dinâmica:** A implementação dinâmica consistiu em receber o nome de um aluno cadastrado no sistema e o título de um livro, também cadastrado. Dado que o livro tem quantidade suficiente disponível, o livro é emprestado, caso contrário, o aluno em questão entra em uma fila de espera, que flui conforme ocorrem as devoluções do livro em questão.

Vale salientar que tanto o nome do aluno quanto o título do livro devem ser válidos, já que o sistema não efetua uma verificação de existência dos nomes. Casos de erro podem ocorrer se um dos nomes for digitado incorretamente.

**Estática:** Primeiramente a função verifica se existem ambos aluno e livro. Caso haja livros disponíveis, o livro é emprestando. Caso não hajam livros disponíveis o aluno entra na fila de espera do livro.

**Retornar Livro**

**Dinâmica:** A implementação dinâmica consistiu em receber o nome de um livro cadastrado no sistema e devolver uma unidade à Biblioteca, simulando a devolução de um determinado livro que estava emprestado a um aluno do sistema. A retorno de um livro implica verificar se a fila de espera possui algum aluno. Se houver, o livro automaticamente passa para o primeiro aluno da lista, fazendo-a fluir. Se não houver, a quantidade do livro na biblioteca é incrementada, apenas.

Vale salientar que o título do livro deve ser válido, já que o sistema não efetua uma verificação de existência do nome. Casos de erro podem ocorrer se o título for digitado incorretamente.

**Estática:** A função verifica se o livro existe no sistema. Se a fila de espera do livro estiver vazia é adicionado um para o total de livros disponíveis, caso contrário uma mensagem é armazenada na caixa de mensagem do próximo aluno na fila de espera, que é removido da fila, e o livro continua indisponível.

**Remover Livro**

**Dinâmica:** A implementação dinâmica consistiu em, dado o título de um livro existente na biblioteca, efetuar a remoção desse livro. Isso implica que não só o livro será removido, mas toda a fila de espera existente nele.

Vale salientar que o título do livro deve ser válido, já que o sistema não efetua uma verificação de existência do nome. Casos de erro podem ocorrer se o título for digitado incorretamente.

**Estática:** Se o livro for o primeiro da lista, a função altera o inicio do banco de dados, e torna aquela parte da sequência em primeiro vazio. Caso não seja o primeiro, ele continua verificando para o próximo do livro que está no laço, dessa forma é possível alterar o inteiro “próximo” do elemento anterior ao livro a ser removido.

**Remover Aluno**

**Dinâmica:** A implementação dinâmica consistiu em, dado o nome de um aluno existente na biblioteca, efetuar a remoção desse aluno do sistema. Isso implica que o aluno será removido da relação de alunos, bem como de todas as filas de espera em que ele estiver contido.

Vale salientar que o nome do aluno deve ser válido, já que o sistema não efetua uma verificação de existência do nome. Casos de erro podem ocorrer se o nome for digitado incorretamente.

**Estática:** Funciona de forma semelhante à função de remover livos, porém esta função conta com a função AlunoFilaRemove, que remove o aluno da fila de espera de todos os livros. A tal função funciona dentro de um laço que usa o valor “total” da lista para limitar o laço.

**IMPRIMIR MENSAGEM**

**Estática:** Função que imprime as mensagens do aluno em forma de pilha, a mensagem que podem haver é de retirar o livro que o aluno estava na fila de espera. Após imprimir as mensagens a pilha esvazia.