**Relatório – Projeto Matriz Esparsa**

Gabrielle da Silva Barbosa RA: 18183 e Felipe Melchior de Britto RA:18200

***Objetivos do projeto:*** fazer uma lista cruzada que armazena as células não nulas (valor 0) de uma matriz esparsa (matriz cuja maioria das posições são nulas) e que faz diversas operações com ela, como somar uma constante à uma coluna, listar num gridview, multiplicar ou somar duas matrizes, etc..

***Desenvolvimento:***

***25/03***

* Início do projeto;
* Releitura da proposta e criação da aplicação;
* Criação da interface para o usuário.

***27/03***

* Criação e desenvolvimento da classe Celula:
* Atributos direita, abaixo e valor;
* LerRegistro(StreamReader arquivo);
* ToString();
* ParaArquivo().

***28/03***

* Criação da classe ListaCruzada e inicio de seu desenvolvimento:
* Construtor, recebendo como parâmetro o tamanho da matriz e criando as células cabeças correspondentes;
* Método Existe(int linha, int coluna, ref Celula abaixo, ref Celula acima, ref Celula esquerda, ref Celula direita), muito útil pois serviu para incluir, alterar, remover etc..
* Listar(DataGridView dgv), para exibir matriz no grid view correspondente.
* Problemas encontrados:
* Método existe deu um certo trabalho por causa do raciocínio necessário para percorrer a matriz e como posicionar as variáveis quando existisse (“esquerda” seria a célula anterior horizontal à encontrada e “direita” a encontrada, “acima” seria a anterior vertical e “abaixo” a encontrada) e quando não existisse (célula posicionada entre “esquerda” e “direita” e entre “acima” e “abaixo”)

***05/04***

* Na classe ListaCruzada:
* Buscar, utilizado no formulário para mostrar para o usuário o valor de uma determinada posição (l, c) que ele pesquisou;
* No formulário:
* Ler um arquivo texto e criar uma matriz com base nele;
* Exibir a matriz em um gridView.

***06/04***

* Desenvolvimento dos métodos:
* Inserir uma célula na matriz nas posições (l, c);
* Remover uma célula da matriz nas posições (l, c);
* Liberar todas as posições da matriz;
* Somar uma constante K a uma coluna y, ambos especificados pelo usuário;
* Somar duas matrizes esparsas;
* Multiplicar duas matrizes esparsas.

***07/04***

* Código da aplicação comentado;
* Problema encontrado: últimas células não apontavam pros nós cabeça, eram null, e se estava trabalhando dessa forma. Fizemos as alterações necessárias (quando se criava nós cabeça e onde se verificava se era null) e corrigimos o problema.

***09/04***

* Notamos que o método “multiplicarMatrizes” havia sido realizado de maneira errada, nós apenas multiplicávamos as células da matriz1 com as suas respectivas células da matriz2. Trabalhamos para corrigir e fazer o método com a maneira correta de multiplicação.

***09/04***

* Verificamos que havia erros no método existe em relação a percorrer horizontalmente pelas cabeças de linha, pois estávamos sempre percorrendo pelas cabeças de coluna, verticalmente, e para se multiplicar matrizes precisava-se fazer isso. Era um problema até que simples, mas demoramos a vê-lo justamente por ser em um método que julgávamos estar certo, o ExisteCelula, que retornava valores errados de esquerda e direita.
* Reduzimos a quantidade de métodos Listar(), que deixavam o programa mais lento, e optamos por métodos como inserir e remover serem bool, retornando true se houvesse sucesso, assim, ao invés de listar de novo, apenas alterávamos o elemento no próprio gridview.
* Por fim comentamos todas as mudanças e organizamos os arquivos da maneira solicitada

***Conclusão:*** Acreditamos que aprendemos bastante com o projeto, tanto em relação a listas ligadas e matrizes, desenvolvemos raciocínios não só para fazer um método, mas também para fazê-lo de forma mais eficiente e sem repetições desnecessárias. Conseguimos fazer tudo que foi pedido, além de algumas coisinhas a mais, como gravar em arquivo e poder mudar valores direto no gridView.