**Universidade Federal do Rio Grande do Sul**

**Instituto de Informática**

**Professor: Leandro Krug Wives**

**Nomes: Bruno Soares Fillmann e Inatan Hertzog Cartões: 219420 e 219432**

**Relatório do Trabalho Final sobre Organização de Curriculum Lattes.**

O problema: Conseguir pegar os currículos do site CNPQ de alguns professores da INF da UFRGS e separar todos e listar seus artigos publicados e separar por tipo, natureza, quantidade de coautores e QUALIS.

Foi utilizada a linguagem de programação C# no programa Visual Studio, nesta existe maior facilidade para montar o programa que resolve o problema proposto. Além de sua grande semelhança com o C tradicional, linguagem que ambos os integrantes já conheciam, entre as vantagens do C# estão as listas, as quais podem ser utilizadas para alocar objetos não se sabendo a quantidade. Listas e arrays têm em C# a função sort que é um quicksort abreviado, por essa facilidade e pela grande eficiência do quicksort para casos aleatórios, não é necessário criar uma função apenas para ordenar sendo que a linguagem já possui uma perfeitamente adequada.

Usamos o Windows Forms, devido a sua facilidade de construção do programa, e a interface conhecida e de fácil construção. Diferente do XNA que tem grande liberdade para a construção da interface, o Windows Forms demonstrou-se uma iniciação melhor para criação de interfaces devido a grande complexidade de implementação do XNA. Optou-se então realizar o trabalho com Windows Forms.

O que é um XML? É uma linguagem de marcação que tem o objetivo de ser lida por computadores e pessoas. Ela utiliza nodos como se fosse uma árvore onde cada marcador pode ter um filho e ou ter um pai. Assim o acesso é dado como uma forma indireta passando pelo pai até o filho procurado. Temos uma raiz também como qualquer árvore estrutural.

Primeiro problema encontrado: foi a acentuação dos XML, abrindo ele com browser é possível encontrar uma palavra que costuma ter acentuação estava impressa com lixo ao invés de imprimir o cacteres acentuado. Em algumas pesquisas na web foi possível ver que esse problema tem origem na encoding do XMLl. Que estava em "ISO-8859-1" esse formato não aceita caracteres acentuados, a solução para o trabalho foi necessário trocar a encoding dos XML para “UTF-8”. Assim o problema foi solucionado. Como foi feita a troca? Ao invés de criarmos uma função que corrija as encodings. Devido a definição do trabalho era baixar os currículos em XML. Era mais rápido trocar a encoding um por um do que pensar em um algoritmo para a trocar todas as encodings sempre quando for lido o XML.

Segundo problema: Leitura do XML devido ao seu formato utilizando a linguagem C# a grande dificuldade é achar um método eficiente que leia as informações necessárias do xml. Com algumas pesquisas tem uma solução que não pode ser utilizada no trabalho pois utiliza banco de dados database. Mas a forma que foi garantida para a busca do xml é a a procura dos nodos no XML e buscar de acordo com a tag procurada o elemento desejado e assim obtendo as formações necessárias, essa busca é mais trabalhosa, mas garante a informação que se deseja buscar. No caso se acessa o nodo e se procura o filho (importante saber como está a disposição dos nodos no XML) .Com todas as informações são então passadas para a classe.

Uma das classes utilizadas será “autores” a qual possui nome, referencia bibliográfica, id. Será utilizado no programa um array de 50 autores, pois temos 50 XML com um autor cada. Para os arquivos serão utilizadas duas outras estruturas, separando os tipos de artigos: periódicos e conferências. A estrutura “artigo” refere-se aos periódicos publicados. Enquanto a “conferencia” refere-se aos artigos de conferências. Ambas possuem título, natureza, id(do seu tipo),ano, quantidade de coautores, QUALIS, e a id do autor como campos do registro lógico. Ambas estão sendo usadas como listas, pois seus tamanhos não podem ser previstos. Levando em consideração a racionalização e normalização de arquivos. Em uma classe estrutura é tudo construindo, contendo assim a formação de todas as classes no programa podendo usar em todos os forms como uma variável global.

No programa temos duas listas, uma contendo o nome dos autores em ordem alfabética, outra contendo o nome de todos os artigos pegos. Temos uma barra de pesquisa para os autores, a pesquisa retira os acentos da estrutura e deixa em letra minúscula procura no array o autor que tem o nome com essa substring, coloca em um auxiliar o nome e nesse auxiliar retira os acentos e deixa em letras minúsculas, fazendo assim o usuário ter maior liberdade de busca. Quando for encontrado nomes que contem essa substring eles serão listados. Existe outra caixa de texto para procurar também os artigos, pesquisando nas lista de conferencias e na lista de periódicos e os lista.

Temos os botões utilizados para listar só aqueles artigos procurados: conferências ou periódicos. Temos também botões que ordenam pela quantidade de coautores, QUALIS, e pela natureza da publicação, dados armazenados nas próprias estruturas. Todos usam a função sort. No caso dos botões que listam artigos de um determinado tipo eles são ordenados em ordem alfabética e depois são listados só do tipo determinado. Já no caso da ordenação, primeiramente são ordenados em ordem alfabética, depois ordenados pelo critério procurado.

Outro diferencial é que quando é dado duploclick em um dos elementos da lista de autores, abre-se um novo form contendo a informação dele e duas listas dos artigos que ele é autor e dos artigos que ele é coautor. A pesquisa é feita primeiro procurando o autor selecionado depois colocando suas informações no novo form . O mesmo processo é feito com a lista de artigos é aberto um novo form com as informações do artigo e uma lista de coautores.

A leitura do CSV teria sido simples, porém encontramos dificuldades com a encoding do arquivo, mas após uma pesquisa realizada, conseguimos traduzi-la para UTF-8, não havendo maiores dificuldades depois.

As estruturas são salvas em binário, podendo-se, após ler-se o arquivo XML, lê-las também por seus respectivos arquivos. Desses existem quatro, cada um contendo apenas um dos seguintes itens: autores, conferências, periódicos, e um final que armazena as produções de cada autor.

Esse último é organizado contendo uma lista de códigos formada por duplas de autor seguido por uma produção sua, podendo várias instâncias do mesmo autor ocorrerem para cada uma de suas produções. Planejava-se também criar-se um índice para esse arquivo, mas o tempo se mostrou insuficiente. Não foram encontradas grandes dificuldades para a leitura e gravação de arquivos binários, dadas as bibliotecas presentes no C#.

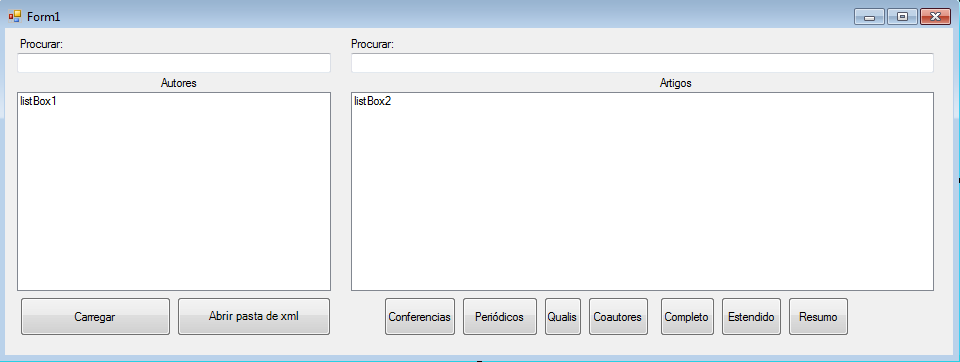
Manual de instruções

Ao abrir o programa caso você não tenha os arquivos necessários para a formatação é necessário você ter uma pasta com 50 xml dos currículos dos autores. E usar a função abrir XML. Caso você possua os arquivos clique em carregar.

As instruções procuraram o titulo ou o nome do autor procurado assim é listado só o nome que pode ser.

Ao clicar duas vezes em um autor. Abre-se uma janela de autores

Ao clicar duas vezes em um artigo. Abre-se uma janela de artigos.



Conferencias: lista apenas as conferências

Periódicos: lista apenas os periódicos

Qualis: Ordena por nota qualis os artigos.

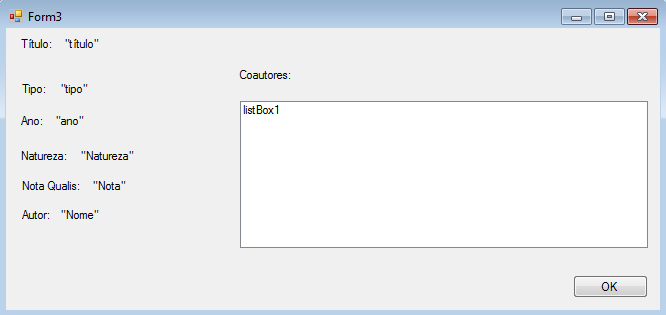
Coautores: Ordena por quantidade de coautores os artigos

Completo: lista apenas os de natureza completo

Estendido:lista apenas os de natureza estendido

Resumo: lista apenas os de natureza Resumo

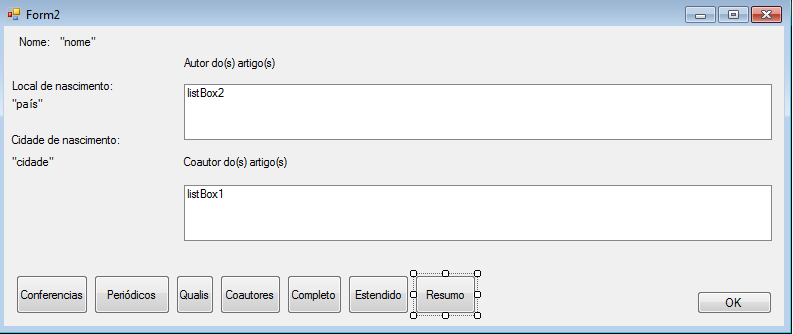
Ao clicar em um artigo abre-se essa janela



Nela se tem o titulo do artigo, tipo, seu ano (caso for 0 é um ano não encontrado ou inreal),sua natureza, sua nota QUALIS, o nome do autor. E a lista de coautores.

Ok fecha a janela.

Ao abrir a janela de autor têm-se as informações do autor e artigos que ele é autor e artigos que ele é coautor



Janela da o nome do autor, o pais onde nasceu, e a cidade de nascimento.

Ao clicar em um artigo qualquer de qualquer lista será aberto a janela de artigos que dá as informações dos artigos.

Os botões têm as mesmas funções do menu principal no entanto ordena as duas listas.

Considerações finais:

Realizando o trabalho tivemos a oportunidade de aprender uma nova linguagem de programação, melhor capacitada para a solução do problema. Também reforçamos diversos conceitos discutidos em aula. Em primeiro lugar, contudo, depois de muito discutir sobre o trabalho, acreditamos ter agora uma maior habilidade para ler, salvar e planejar arquivos.