Foi utilizada a linguagem de programação c# onde existe maior facilidade para montar o programa que resolva o problema, grande vantagens do c# existem listas onde pode ser usadas para alocar objetos que tu não sabe a quantidade. Listas e arrays tem a função sort que é o quicksort abreviado, por essa facilidade não é necessário criar uma função só para ordenar sendo que a linguagem já possui essa função.

Usamos o Windows Forms, devido a facilidade da construção do programa e da interface conhecida e de fácil construção. Diferente do XNA que tem grande liberdade para a construção da interface, no entento a implementação é mais complicada. Então mudamos para o 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 arvore 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 arvore 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 carácteres acentuado. Em algumas pesquisas na web foi possível ver que esse problema tem origem na encouding do xml. Que estava em "ISO-8859-1" esse formato não aceita carácteres acentuados, a solução para o trabalho foi necessário trocar a encouding dos xml para “UTF-8”. Assim o problema foi solucionado. Como foi feita a troca? Ao invés de criarmos uma função que corrija as encoudings. Devido a definição do trabalho era baixar os currículos em xml. Era mais rápido trocar a encouding um por um do que pensar em um algoritmo para a trocar todas as encoudings 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 tu acessa o nodo o procura o filho (importante saber como ta a disposição dos nodos no xml) em caso de elementos iguais se prega um por um no caso de AUTORES é necessário pegar um por um. Com todas as informações dadas é só passar para a classe.

As classes utilizadas serão autores onde têm nome, referencia bibliográfica, id. Vai ser usado no programa como um array de 50, pois temos 50 xml com um autor cada um autor cada. Para os arquivos separamos os tipo de artigos: periódicos e conferencias. Artigos se refere aos periódicos. Ambos tem titulo, natureza ,id(do seu tipo),ano, quantidade de coautores, QUALIS, a id do autor e uma lista do nome dos coautores. Ambos estão sendo usados em listas, pois seus tamanhos não se sabe. Tudo tendo importância para a pesquisa e a classificação de acordo como a definição do trabalho. Uma classe estrutura onde já é construindo tudo, tendo assim a formação de todas as classes no programa podendo usar em todos os forms como uma variável global.

No programa temos tuas 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 o 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: Conferencias, Periódicos, Completos, Estendidos e Resumos. Temos também botões ordenam de Quantidade de coautores e QUALIS. 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 ordenadas no 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 coloca as 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.