

Problema 01: Pesquisa de opinião

Uellington Da Conceição Damasceno

¹Curso de Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)

Abstract. *This report should be presented as regards the steps required for the implementation of opinion polling software, whose main purpose is to receive data, process and transform the data into treated and meaningful data. Remembering that the software aims at agility in the interview and in the overall result. It should also be noted that the software was developed in C programming language.*

Resumo. *Neste relatório serão mostrados, as etapas necessárias para a implementação de um software de pesquisa de opinião, cujo o principal objetivo receber alguns dados, processá-los e transformá-los em informações tratadas e significativas. Lembrando que o software visa a agilidade na entrevista e no resultado geral. Deve-se também destacar que o software foi desenvolvido em linguagem de programação C.*

1. Introdução

Com o decorrer dos anos, foi-se descobrindo que a metodologia tradicional de ensino e aprendizagem não estava contemplando todos os alunos. Como solução para esse problema, novas abordagens pedagógicas foram criadas. Consequentemente, novas opiniões com relação a eficácia dessas metodologias surgiram. Diante disso, o professor da disciplina de MI de algoritmo resolveu descobrir qual era a opinião dos alunos do curso de engenharia de computação com relação a metodologia PBL.

O público alvo da pesquisa são os alunos veteranos e ou egressos do curso de engenharia de computação. Tendo em vista a grande quantidade de alunos que seriam entrevistados, notou-se que a computação dos dados da pesquisa se tornaria exaustiva caso a pesquisa fosse feita utilizando o método tradicional (Papel e caneta).

Decidiu-se que, a maneira mais rápida e eficaz para resolver o problema com relação a agilidade no processamento dos dados era o desenvolvimento de um *software* de pesquisa de opinião que pudesse fazer um número de pesquisas ilimitadas e ao final exibisse os resultados de acordo com os critérios utilizados adotados pelo professor.

Vale também ressaltar que o *software* foi desenvolvido em linguagem de programação C para o sistema operacional Windows. Utilizando o DEV-C++ como IDE.

Neste relatório, iremos apresentar como foi feito, e quais os critérios foram considerados para a implementação deste produto, para um melhor entendimento de como foi desenvolvido este *software*, este relatório foi dividido em quatro partes, sendo elas: Introdução, metodologia, resultados e discussões, conclusão e por fim apresentar as referências que foram utilizadas ao longo do projeto.

2. Metodologia

Após algumas reuniões, existiram algumas ideias e questões que foram discutidas e ao final ficou decidido alguns fatos que deveriam ser implementados no programa. Menu ao final de cada pesquisa, ordem das questões, cálculo, processamento, relatório e por fim funções complementares. Essas foram algumas ideias discutidas durante alguns encontros que ocorreram durante o desenvolvimento do *software*.

2.1. pesquisa ilimitada

Pelo fato de não sabermos a quantidade exata de alunos que terminaram e que ainda fazem o curso, foi implementado um laço de repetição que permite ao usuário fazer quantas pesquisas forem necessárias, dessa maneira não será necessário que o usuário retorne ao menu principal sempre que decida fazer outra pesquisa.

2.2. Ordem das questões

Para um melhor funcionamento do programa, as questões que seriam utilizadas na pesquisa foram separadas em dois grupos dos quais foram chamados de grupo de questões sendo elas as: obrigatórias (sexo e idade) e as relativas (ano de formatura e grau de escolaridade).

O grupo de questões obrigatórias tem prioridades sob as relativas. Pois as questões pertencentes ao grupo obrigatório são feitas para todos os usuários. Já o grupo de questões relativas contém algumas peculiaridades determinadas de acordo com algumas características. Por esse motivo essas questões ficaram em segundo plano se compararmos com as questões do primeiro grupo.

2.3. Cálculos, processamento e relatório

Como já foi dito anteriormente o programa deve além de receber os dados de forma individuais como devem processá-los e transformá-los em informações significativas para o interessado pela pesquisa.

Os problemas com os cálculos foram consideravelmente simples de serem resolvidos. Apesar que, em determinados momentos alguns bugs apareceram devido a funcionalidade de interação entre os menus (função complementar individual). Por esse motivo os cálculos foram deixados para serem realizados ao final de cada pesquisa. Pois, desta maneira impediria que existisse algum erro surgissem caso o usuário quisesse retornar para o menu principal como exemplo.

Após a conclusão da primeira pesquisa e com todos os cálculos devidamente efetuados o programa desbloqueia a opção de “Exibir relatório” (que estava bloqueada devido à falta de dados necessário para obtenção do relatório completo), permitindo ao usuário ler as informações descritas no relatório.

2.4. Complementos

Mesmo utilizando confirmações de respostas foi implementado uma função responsável por manter uma interação entre os menus, e assim consequentemente permitindo ao usuário voltar de um menu para o outro, caso exista algum erro ou arrependimento. Desta maneira o programa se tornou mais dinâmico e usual.

3. Resultados e discussões

De como simplório o algoritmo consiste em variáveis de processamento, variáveis de controle, laço de repetição e por fim estruturas condicionais.

No algoritmo as variáveis de controle podem assumir dois valores sendo eles 0 e 1. Elas receberam esse nome pois são responsáveis por controlar o funcionamento dos laços de repetições que mantêm o vínculo entre os menus. Para melhor identificação dessas variáveis no código, houve uma padronização na nomenclatura onde todas as variáveis começam com “repetir” e o nome do menu que repete.

As variáveis de processamento recebem esse nome pois são utilizadas para armazenar valores que posteriormente serão calculados, os resultados gerados pelo processamento dessas variáveis será exibido no relatório.

Os demais laços de repetições (os que não repetem os menus) foram utilizados em todas as perguntas do programa. Pois, eles têm como função tratar de erros que são gerados a partir de uma escolha indevida do usuário.

No início do algoritmo foi implementado um menu principal composto por quatro opções sendo elas: Nova pesquisa, exibir relatório, sobre e sair. O menu principal utiliza um switch. Que fica responsável por verificar a qual foi a escolha do usuário e direcioná-lo para o caso escolhido.

Toda a estrutura do menu fica envolta a um laço de repetição (do-while), que por sua vez impede que o usuário continue caso digite um valor diferente dos valores permitidos.

Caso o usuário deseje realizar novo cadastro, o programa começará a recolher as informações pessoais do entrevistado. Scanf foi a função utilizada para efetuar a leitura do teclado e assim consequentemente armazenar as respostas do usuário nas variáveis de processamento.

Ao final da pesquisa um novo menu irá aparecer para o usuário permitindo que ele possa escolher entre fazer um novo cadastro, exibir um relatório gerado com os valores dos dados recolhidos até aquele momento ou voltar ao menu principal.

Vale ressaltar que o programa possui tratamento de string. Por esse motivo, caso o usuário digite alguma letra por engano ou até mesmo por falta de atenção, o programa simplesmente irá exibir uma mensagem de erro e em seguida permitirá que o usuário digite novamente.

Com relação às saídas do *software* podemos dizer que, tem como grande maioria textos informativos responsáveis por direcionar o usuário, como por exemplo as mensagens de erro que aparecem caso o usuário digite um valor inválido.

Durante o desenvolvimento foram efetuados diversos testes que visavam a segurança e a eficácia do programa, os principais testes efetuados foram a inserção de valores inválidos como strings, valores acima ou abaixo dos permitidos e até mesmo valores altos.

Após alguns testes foi descoberto que havia um mal funcionamento com o laço de repetição que englobava o menu “grau de escolaridade”, esse mal funcionamento se deu devido ao fato de que o programa continuava a execução dos comandos mesmo quando o

usuário pedia para retornar ao menu anterior. Para resolver esse problema foi implementado uma verificação que impedia que o programa continuasse executando caso o usuário desejasse retornar ao menu anterior.

Outro grande problema encontrado foi no laço de repetição responsável por repetir os menus. Pois, sempre que retornava pela primeira vez, o trecho de código que ficava envolto pelo laço entrava em loop infinito.

Foi descoberto que o bug acontecia por causa do valor da variável de controle, que durante a primeira execução teria o valor alterado para 1 e assim consequentemente tornando a verificação do laço de repetição sempre verdadeira.

A solução para esse problema foi simples. Pois, só foi necessário fazer a variável retornar ao valor 0 sempre que o laço fosse executado.

4. Conclusão

Após todo o trabalho de desenvolvimento, o programa chegou em um ponto onde consegue executar com excelência todos os requisitos mínimos necessários. Requisitos esses que varia desde um simples cálculo da média de idade até a emissão do relatório completo.

Na atual versão do programa não existe nenhum bug encontrado. Entretanto, é possível que exista algum problema ainda não foi descoberto, que pode ser desencadeado depois de alguma sequência de inserção de entradas por exemplo.

Para um melhor desempenho do programa, poderia ser implementado um banco de dados que ficaria responsável por armazenar informações dos usuários entrevistados e assim, consequentemente permitindo ao usuário fazer outras verificações como por exemplo o mês em que mais foram entrevistadas mulheres egressas.

5. Bibliografia Consultada

Nenhuma fonte foi consultada.