

Problema 2: Enquetes flexíveis

Problema

O Núcleo de Estudos da Democracia da UEFS (NED-UEFS) gostou muito do trabalho realizado pelos estudantes do primeiro período do curso de Engenharia de Computação da UEFS e resolveu ampliar a parceria. Após o projeto piloto realizado com o questionário sobre autoritarismo, a coordenação do núcleo resolveu contratar os estudantes para fazer um sistema mais geral para atender a quase todas as necessidades que o núcleo tem para realizar enquetes com a comunidade.

Segundo Angela Pimenta, coordenadora de TI do núcleo, a maior dificuldade em relação ao projeto piloto foi a necessidade de encerrar a enquete toda em um mesmo dia, por não se poder salvar os dados antes de terminar a enquete. Ela pede aos futuros engenheiros que providenciem um modo de escrever os dados brutos das entrevistas já realizadas em um arquivo de texto “estilo bloco de notas”. Neste arquivo, cada dado demográfico de cada respondente (curso de graduação, sexo e idade) deve ter sua própria linha do arquivo de texto, além de haver um linha adicional com todas as respostas à cada pergunta da enquete. Assim, o pesquisador responsável pode recuperar os dados das entrevistas anteriores realizadas até então e continuar com entrevistas adicionais, salvando-as ao final de cada entrevista. Ela diz ainda que “nunca fizemos uma enquete com mais de 500 entrevistados, mas era bom colocar um limite generoso de até 1000 entrevistados, para não ter problemas no futuro”.

Angela menciona ainda que ela gostaria que o sistema tivesse enquetes mais genéricas, com um número flexível de questões. Ela não menciona limites, mas lembra que a maior enquete que já usou tinha 90 perguntas. E diz: “Se o sistema aceitar até 100 perguntas, já está bom demais”. Ela sugere que as perguntas sejam recuperadas de outro arquivo de texto “estilo bloco de notas”, onde cada linha do arquivo contenha o texto de uma questão inteira. “Assim, o pesquisador não precisa se preocupar de pedir ao programador para ficar mudando o sistema o tempo todo. Basta o programador fazer um sistema flexível que ‘adivinha’ quantas questões têm a enquete a partir do número de perguntas do arquivo”.

Ela explica ainda que o pesquisador deve colocar no arquivo, após cada linha de pergunta, uma linha adicional com o código do padrão a ser buscado para as respostas. Por exemplo, se o pesquisador escolheu o código 1, será usada uma escala com 3 níveis {“Não”, “Mais ou menos”,

“Sim”}. Se escolheu o código 2, será usada a escala de 5 níveis {“Não é verdade”, “Um pouco verdadeiro”, “Moderadamente verdadeiro”, “Muito verdadeiro”, “Totalmente verdadeiro”}. Se escolheu o código 3, será usada a escala de 4 níveis {“Discordo totalmente”, “Discordo parcialmente”, “Concordo Parcialmente”, “Concordo totalmente”}. Ela diz ainda: “Obviamente, essas escalas podem ser buscadas de dentro de um terceiro arquivo, com o código da escala, o número de itens da escala e os itens da escala. Isso que eu disse é apenas um exemplo, já que existem várias possíveis escalas de Likert”.

O sistema deve ser organizado em menus. Por exemplo, ela quer que o sistema apresente uma opção de menu que mostre a frequência absoluta de cada nível de concordância, para cada questão respondida, de todas as entrevistas realizadas até o momento presente. Além deste exemplo, ela também quer outras opções de menu para entrar novas entrevistas, para recuperar dados de entrevistas anteriores através de arquivo, para calcular e mostrar algumas estatísticas como:

- Média aritmética de cada questão;
- Moda, ou seja, o valor mais frequente de resposta de cada questão.

Ela fala também de opção de menu para escores, sugere que eles podem querer computar escores a partir de um subconjunto de questões da enquete: “Geralmente, os escores são simples, o pesquisador diz quais são as questões que fazem parte do escore e é computada uma média aritmética das respostas de um entrevistado do dado conjunto de questões. É interessante também gerar não só o escore individual de cada respondente, mas as mesmas estatísticas anteriores para o conjunto dos respondentes, ou seja, a média do escore e a moda do escore.”

Finalmente, ela sugere que pode querer filtrar e apresentar os resultados categorizados, por exemplo, para um dado curso de graduação, para um dado sexo ou para uma dada faixa etária estabelecida pelo pesquisador.

E, para terminar, ela diz: “Olha, da última vez que a gente contratou um pessoal de informática para fazer algo parecido, os danados fizeram um software muito bagunçado. Quando a gente contratou outra pessoa para dar manutenção no código, era mais fácil achar uma agulha no palheiro. Quando forem fazer o software, não esqueçam de modularizar direitinho tudo, separar as opções de menu, separar entrada de dados de cálculos de estatísticas, separar o cálculo de cada estatística, separar a gravação dos dados de saída, essas coisas

para facilitar a vida de quem for manter o software depois, mais na frente, senão o trabalho acaba sendo em vão.”

Produto e Relatório

Você deve desenvolver o código fonte do sistema e um relatório final, no formato de artigo da SBC, conforme modelo e instruções disponibilizados no site do MI. O código e relatório final deverão ser enviados até ao meio dia do dia **14/11/2018** (a entrega impressa do relatório final pode ser solicitada pelo tutor). O relatório final só será aceito mediante entrega do código fonte.

O desempenho nas sessões tutoriais equivale a 30% da nota no Problema. O relatório equivale a 30% da nota e o código fonte (produto) equivale a 40%. Haverá penalidade de 1 **ponto** por descumprimento do prazo de entrega e 1 ponto por dia de atraso na entrega, até o máximo de cinco dias. Após este prazo, o trabalho não será mais aceito.

Tanto o código fonte quanto o relatório devem ser desenvolvidos **individualmente**. Deve constar no código fonte declaração de ausência de plágio.

Recursos para Aprendizagem

MANZANO, J. A. N. G., OLIVEIRA, J. F.

Algoritmos: Lógica para Desenvolvimento de Programação. São Paulo: Érica, 1996.

MANZANO, J. A. N. G. **Estudo Dirigido de Linguagem C.** São Paulo: Érica: 1997.

FORBELLONE, A. V. L., EBERSPACHER, H. F.

Lógica de Programação : A Construção de Algoritmos e Estrutura de Dados. 2. ed. São Paulo: Makron Books, 2000.

MIZRAHI, V. V. **Treinamento em Linguagem C: Módulo 1.** São Paulo : Makron Books, 1990.

MIZRAHI, V. V. **Treinamento em Linguagem C: Módulo 2.** São Paulo : Makron Books, 1990.

SCHILDT, H. **C Completo e Total.** 3. Ed. São Paulo: Makron Books, 1996.

Cronograma

Data	Sessão Tutorial
11/10/18	Sessão Tutorial Inicial - Problema 2
18/10/18	Sessão Tutorial - Problema 2
25/10/18	Sessão Tutorial - Problema 2
01/11/18	Sessão Tutorial - Problema 2
08/11/18	Sessão Tutorial - Problema 2
14/11/18	Entrega do Código-Fonte e Relatório.
15/11/18	FERIADO
22/11/18	Apresentação do Problema 3