



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados I
Professor: Mário Luiz Rodrigues Oliveira
Atividade: 2º Trabalho Prático
Formiga, MG, 5 de agosto de 2014

INSTRUÇÕES:

1. Esta atividade pode ser resolvida em grupo composto por no máximo 2 integrantes.
2. Caso você ache que falta algum detalhe nas especificações, você deverá fazer as suposições que julgar necessárias e escrevê-las no seu relatório. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.
3. Como produtos da atividade serão gerados dois artefatos: códigos fontes da implementação e documentação da atividade.
4. Cada arquivo-fonte deve ter um cabeçalho constando no mínimo as seguintes informações: nome(s) e matrícula do(s) autor(es) do trabalho.
5. O arquivo contendo a documentação da atividade (relatório) deve ser devidamente identificado com o(s) nome(s) e matrícula do(s) autor(es) do trabalho. O arquivo contendo o relatório deve, obrigatoriamente, estar no formato **PDF**.
6. Devem ser entregues os arquivos contendo os códigos-fontes e o arquivo contendo a documentação da atividade (relatório). Compacte todos os artefatos gerados **num único arquivo no formato RAR**. Envie apenas um arquivo por grupo.
7. O trabalho deve ser implementado usando-se, exclusivamente, a linguagem de programação Pascal.
8. O prazo final para entrega desta atividade é até as **23:59:00** do dia **19/08/2014** via portal acadêmico acessado pela **URL: <https://meu.ifmg.edu.br/>**.
9. O envio é de total responsabilidade do aluno. **Não serão aceitos trabalhos enviados fora do prazo estabelecido.**
10. **Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos.**
11. O valor desta atividade é 10 pontos.



1. Objetivo e Descrição do Trabalho

O objetivo deste trabalho é realizar a simulação de padrões de fila numa agência bancária que possui 5 caixas atendendo. A simulação deve determinar valores para os seguintes parâmetros:

1. tempo médio de espera do cliente para ser atendido
2. quantidade de clientes atendidos num expediente bancário
3. quantidade de transações realizadas
4. quantas vezes cada transação foi realizada
5. quantidade de clientes que ainda estavam na fila quando do fim do expediente bancário

Adicionalmente procura-se uma resposta para o seguinte questionamento: é melhor manter uma única fila de clientes ou filas individuais para cada atendente? Para responder tal questionamento simule dois 2 cenários, a saber:

1. fila única
2. uma fila para cada atendente

Após as simulações compare os 5 parâmetros especificados acima e decida qual a melhor alternativa.

Uma das maneiras de realizar a simulação solicitada é ficar parado na porta do banco e perguntar aos clientes que saem a hora em que chegaram no banco e marcar a hora de sua saída, subtrair a primeira da segunda para cada cliente e calcular a média de todos os clientes. Deve-se, também, manter contadores que informem a quantidade de clientes atendidos, a quantidade de transações realizadas e a quantidade de clientes na fila quando do fim do expediente bancário. É necessário questionar cada cliente o tipo de transação realizada de forma a contabilizar a frequência de cada tipo de transação. Entretanto, isso não seria muito prático. É difícil assegurar que todo cliente saiba, por exemplo, a hora exata de sua chegada. Para contornar tal dificuldade, pode-se escrever um programa que simule as ações dos clientes numa agência bancária.

Para realizar essa simulação considere que o cliente ao entrar no banco encaminha-se para o final da fila única ou para o final da fila mais curta (se as filas forem individuais para cada atendente) e aguarda o momento do seu atendimento. Cada cliente realizará uma e apenas uma das transações especificadas na Tabela 1. À medida que o atendente fica livre ele chama o próximo cliente da fila se fila única ou o próximo cliente de sua respectiva fila (na hipótese de filas individuais para cada atendente). Quando o cliente entra na



fila, o horário de entrada é anotado. Quando ele sai, verifica-se quanto tempo ele aguardou na fila. O tempo que cada cliente vai demorar no caixa é função da transação a ser realizada. O tempo de cada transação é conforme indicado na Tabela 1. No processo de simulação, essa transação deverá ser escolhida aleatoriamente. Use um cronômetro para simular o tempo. Quando terminar o expediente (a ser definido pelo usuário e controlado pelo cronômetro), o processo de atendimento é imediatamente encerrado. O horário de chegada do cliente e a quantidade de clientes que chegam no banco num determinado momento são definidos de maneira aleatória.

Tabela 1: Transações Bancárias

TRANSAÇÃO	CÓDIGO	TEMPO (EM MINUTOS)
Saldo	0	1
Saque	1	2
Depósito	2	2
Extrato	3	1
Pagamento em dinheiro	4	2
Pagamento em cheque	5	3
Malote	6	10

O programa implementado deve atender, também, as seguintes especificações:

- o código fonte do programa deve ser portátil, ou seja, o mesmo código fonte deve ser compilado corretamente e gerar código executável para as seguintes plataformas: sistemas operacionais Windows e GNU/Linux e arquiteturas *Intel* x86/x86-64 e *AMD64*/x86-64
- o código fonte do programa deve ser compatível com o seguinte compilador: Free Pascal (FPC versão 2.6.2)

2. Artefatos

Esta seção descreve o que deve ser gerado como produto final do trabalho. Ao final do trabalho deve ser gerado além das implementações, um relatório documentando seu programa, com as seguintes informações:

1. introdução: apresente o problema a ser resolvido. Descreva sucintamente a solução proposta e dê uma visão geral do que será apresentado no relatório;
2. implementação: descrição sobre as decisões de projeto e implementação do programa. Essa parte da documentação deve incluir uma descrição das estruturas de dados usadas no



programa; funcionamento das principais funções e procedimentos utilizados; o formato de entrada e saída dos dados, como executar o programa e as decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado;

3. análise das medidas realizadas: descreva a metodologia usada para realizar o estudo, incluindo uma descrição do sistema computacional, a linguagem de programação e o compilador usado na implementação dos algoritmos. Apresente e discuta comparativamente os resultados obtidos. Faça uso de tabelas e gráficos para facilitar suas explicações. Procure responder claramente o seguinte questionamento: é melhor uma fila única ou uma fila para cada atendente?
4. conclusão: explicita as recomendações do grupo indicando em quais situações cada um dos algoritmos de ordenação comparados e avaliados devem ser utilizados. Inclua alguns comentários/avaliação sobre o trabalho considerando: a experiência adquirida, a contribuição para o aprendizado da disciplina, as principais dificuldades encontradas ao implementá-lo e como tais dificuldades foram superadas;
5. bibliografia: cite as fontes consultadas na resolução do trabalho;
6. listagem dos códigos-fontes do programa.

Todos os artefatos (códigos fontes e relatório) devem ser entregue conforme as instruções contidas nesse documento.

3. Critérios de Correção

Conforme descrito no plano de aula, os critérios de avaliação do trabalho são:

- somente serão corrigidos os trabalhos com códigos fontes portáteis e sem de erros de compilação;
- apresentação (30%);
- análise código fonte: modularização, uso adequado de comentários, legibilidade, corretude e indentação do código, e (40%)
- documentação (30%).

4. Bibliografia

EDELWEISS, Nina; GALANTE, Renata. Estrutura de Dados. Porto Alegre: Bookman, 2009.