



### **Problema 1 - Estruturas de Dados Lineares**

#### **Motivação:**

O atendimento ao cliente é um dos pontos críticos de uma série de segmentos da economia, desde supermercados, passando por bancos e repartições públicas. Um controle de acesso aos guichês de atendimento eficiente eleva o grau de satisfação do usuário do serviço pois reduz o tempo de espera no atendimento.

Uma das soluções mais empregadas para o controle de atendimento é o sistema de senhas, que permite que diversos guichês sejam gerenciados de forma transparente, sem que o cliente tenha que escolher um e tenha a sensação que o dele é sempre o mais lento.

#### **O Problema:**

Você foi contratado por uma *startup* que deseja desenvolver um **Sistema de Controle de Atendimento** baseado em **Senhas**. Sua primeira tarefa é desenvolver um protótipo que simula o funcionamento do sistema. A partir de um levantamento de requisitos preliminar os donos da empresa determinaram que o sistema deverá funcionar da seguinte forma:

1. O sistema é baseado em guichês de atendimento e clientes;
2. O cliente precisa baixar o *app* e instalar em seu dispositivo móvel;
3. Clientes solicitam um atendimento (senha) através do *app* de seu dispositivo móvel, lendo um QRCode que identifica o estabelecimento. A senha (não sequencial) é gerada automaticamente pelo sistema e enviada ao cliente pelo *app*;
4. atendimentos são classificados em expressos, normais ou complexos, relacionados ao tempo médio necessário. Essa informação é solicitada ao cliente após sua identificação e antes da geração da sua senha;
5. A partir da geração da senha o sistema aloca o cliente a um guichê. O critério é o que leve ao menor tempo de atendimento possível;
6. A ordem de atendimento deve respeitar, tanto quanto possível, a ordem de chegada dos clientes;
7. Haverá sempre ao menos um guichê para atendimento de clientes com prioridade por lei;
8. O sistema deve reduzir ao máximo a ociosidade dos guichês;
9. Por questões legais, os atendentes tem direito a um intervalo após um número X de atendimentos consecutivos ou a cada período Y de tempo. O sistema deve controlar esse processo de forma transparente.
10. Todo atendente tem um “botão de pânico” que pode ser acionado se alguma intercorrência acontecer, que o leve a suspender seu atendimento. Nesses casos, o sistema deve redirecionar os clientes alocados a esse guichê para os demais guichês, respeitando o item 5.

Seu simulador deve permitir que os donos da empresa visualizem o processo de alocação de clientes aos guichês ativos, para avaliar a solução proposta por você.

Para o item 10 dos requisitos os donos da empresa sugeriram que a realocação do atendimento no guichê “em pânico” seja feita redirecionando os clientes que chegaram mais recentemente aos guichês com maior tempo de espera, enquanto que os que estão a mais tempo aguardando atendimento sejam priorizados para os guichês de menor tempo de espera.

### **Objetivos:**

O objetivo principal do trabalho é aplicar os conceitos de **Tipos Abstratos de Dados (TAD) Lineares** [1][2][3] apresentados em sala de aula, no contexto de um problema real e prático.

Os alunos deverão analisar o problema proposto, e definir qual o melhor **TADs** para a resolução do problema. Mais de um **TAD** pode ser necessário.

Escolhidos os **TADs**, sua implementação deverá ser integrada ao problema e uma solução computacional codificada em linguagem C.

### **A Implementação:**

A implementação deve ser desenvolvida em linguagem C ANSI (independente de qualquer IDE ou SO).

Você deverá definir seus **TADs** baseados no conceito de modularização, utilizado nas aulas de laboratório.

Os trabalhos deverão ser desenvolvidos individualmente. O código fonte gerado deve ser comentado e legível. Acompanhando o código fonte um breve relatório técnico (README.txt) deve ser entregue, descrevendo os TADs utilizados e justificando suas escolhas.

### **A Entrega:**

O trabalho deverá ser submetido via *Moodle*, respeitando a data e hora limite para entrega. Em caso de atraso, será aplicado um fator de penalização de 1,0 ponto por dia de atraso. Qualquer problema de arquivos corrompidos ou similar o trabalho será considerado não entregue. Portanto, verifique bem o que for entregar.

Os arquivos devem ser enviados seguindo o seguinte padrão: arquivo compactado (zip, rar, tgz ou gzip apenas) contendo um diretório com o nome do(s) aluno(s) e seu arquivos. Arquivos fora desse padrão sofrerão penalização de 0,5 ponto na nota final. Códigos com erros de compilação ou qualquer outra pendência que impeça a compilação não serão avaliados.

A cooperação entre alunos e grupos é considerada salutar. No entanto, trabalhos com alto grau de similaridade serão tratados como “plágio”, o que resultará em avaliação **zero** para **TODOS os envolvidos**.

**Qualquer dúvida adicional, evite problemas: não presuma nada, procure o professor para esclarecimentos.**

### **Referencias Bibliográficas:**

- [1] Ziviani, Nivio. **Projeto de Algoritmos: com Implementações em Pascal e C**. Vol. 2. Thomson, 2004.
- [2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. **Introdução a algoritmos**. 2001.
- [3] Sedgewick, Robert. **Algorithms in C++**. Vol. 2. Pearson Education India, 2003.