

# Implementação Tattoo e Piercing

Recentemente a grande franquia mundial de tatuagens e *body piercing* foi agraciada com um estúdio na região. Trata-se da Esperta (**Estúdio de Piercing e Tatuagem**), um local onde os apreciadores desses adornos e decorações corporais podem se sentir a vontade e realizar todos os seus desejos. Já na inauguração estão previstas muitas pessoas interessadas em contratar e realizar seus serviços, devido a grande fama mundial da franquia. Nesse sentido, os organizadores estão preocupados em causar uma boa impressão, desde o primeiro minuto de abertura do Estúdio, dessa forma resolveram desenvolver um simulador de estúdios de realização tatuagens e colocação de *body piercing*, a fim de avaliar o fluxo de clientes e profissionais no recinto.

De acordo com as normas da franquia, a Esperta terá uma sala de entrada, onde as pessoas serão atendidas pela recepcionista. Após o(a) cliente informar o que deseja, deve-se encaminhar ele(a) para uma das ante-salas disponíveis: uma de tatuagem e outra de *body piercing*. Em cada ante-sala há uma porta que leva a sala exclusiva do profissional (privacidade é um dos pilares da franquia) e um número definido de *pufs*, para uma melhor acomodação dos clientes enquanto esperam sua vez. O pressuposto é que se a porta da sala está aberta, o profissional está disponível e o cliente dirigi-se a sala, fechando a porta e iniciando seu atendimento. Se a porta estiver fechada (o profissional está ocupado ou ainda não chegou), o(a) cliente deve se acomodar num dos *pufs* e esperar sua vez. No caso de não ter *pufs* disponíveis (todos ocupados), o(a) cliente vai embora, um tanto frustrado(a) o que a Esperta não quer que aconteça.

O atendimento dos profissionais a(o) cliente dura uma certa quantidade de minutos, que varia de acordo com a complexidade do adorno corporal a ser realizado. Não é permitido (normas da franquia) que um(a) cliente realize no mesmo ato, solicitação para tatuagem e *body piercing*. Caso o(a) cliente deseje as duas especialidades da Esperta, deve repetir o processo duas vezes.

O simulador deve levar em consideração a jornada de trabalho dos profissionais, medidos no simulador de acordo com o número de clientes atendidos. Dessa forma, quando for atingido o número de clientes especificado para o dia, o simulador deve encerrar suas atividades de recepção de clientes, sendo apenas os clientes aguardando nos *pufs* atendidos e encerrar sua execução.

## Entrada:

As entradas devem vir como argumentos via linha de comando.

- CLI: número de clientes atendidos no dia que impede a chegada de novos clientes.  $100 > CLI > 10$ ;
- PT: quantidade de *pufs* para espera de tatuagem.  $10 > PT > 0$ ;
- PBP: quantidade de *pufs* para espera de *body piercing*.  $10 > PBP > 0$ ;
- PAT: probabilidade de chegada de um cliente para tatuagem.  $0,99 > PAT > 0$ ;
- MINATEN: tempo mínimo de atendimento de um profissional.  $100 > MINATEN > 0$ ;
- MAXATEN: tempo máximo de atendimento de um profissional.  $500 > MAXATEN > MINATEN$ ;

## Saída:

A saída deve ser mostrada na saída padrão. Deve-se mostrar a atividade dos profissionais e a chegada dos clientes.

Mostrar as seguintes ações dos profissionais:

- Quando iniciou o atendimento de um cliente;
- Quando finalizou o atendimento de um cliente;

Mostrar as seguintes ações de cada cliente:

- Chegou e foi atendido imediatamente;
- Chegou e foi sentar num *puf* de espera;
- Chegou e foi embora pois não foi atendido e não podia sentar (não há *pufs* disponíveis);

Ao final do programa, deve ser mostrado:

- Quantidade de clientes que foram atendidos (total);
- Quantidade de clientes atendidos pelo tatuador e pelo *body piercer*;
- Quantidade de clientes que chegaram e foram embora sem serem atendidos;
- Quantidade de clientes que foram atendidos sem esperar nos *pufs*;

## Funcionamento

O problema consiste em programar o problema da Esperta e fazê-lo funcionar de forma concorrente. Deve ser implementado em C/C++, sendo cada um dos profissionais implementados por meio de uma *thread* e outra *thread* deve ser usada para a chegada de clientes. O atendimento pode ser simulado por meio da função `usleep()`, sendo gerado um valor aleatório entre MINATEN e MAXATEN, representando o tempo em minutos que demora para um profissional realizar um atendimento. A chegada de um cliente deve ser aleatória, sendo que a cada intervalo de 100 minutos deve chegar entre 1 e 10 clientes.

Na chegada de um cliente deve-se definir o que ele deseja fazer. Para tanto, deve-se gerar um número aleatório entre 0 e 1. Caso o número gerado seja menor ou igual a PAT, o cliente quer realizar atendimento com um tatuador. Caso contrário, ele quer o atendimento de um *body piercer*.

Lembrar de iniciar o gerador de números aleatórios de modo a gerar sequencias diferentes de números a cada execução.