

TRABALHO 1 – ENUNCIADO

INE 5410-03208 (Turmas A e B)

Programação Concorrente (2022.1)

RESTAURANTE UNIVERSITÁRIO

Suponha que você foi contratado(a) pelo Departamento de Nutrição (DN) para desenvolver um sistema (em linguagem C) que simula o comportamento de um novo restaurante universitário (RU). O objetivo do DN é o de apresentar a sua simulação em uma reunião com a reitoria da Universidade para validar o fluxo de trabalho e convencer o Departamento Financeiro Universitário a criar um novo restaurante para os alunos do Centro Tecnológico (CTC), visto que, o restaurante principal está muito distante, as filas estão muito grandes e os alunos do CTC estão sempre se atrasando para as aulas no período vespertino.

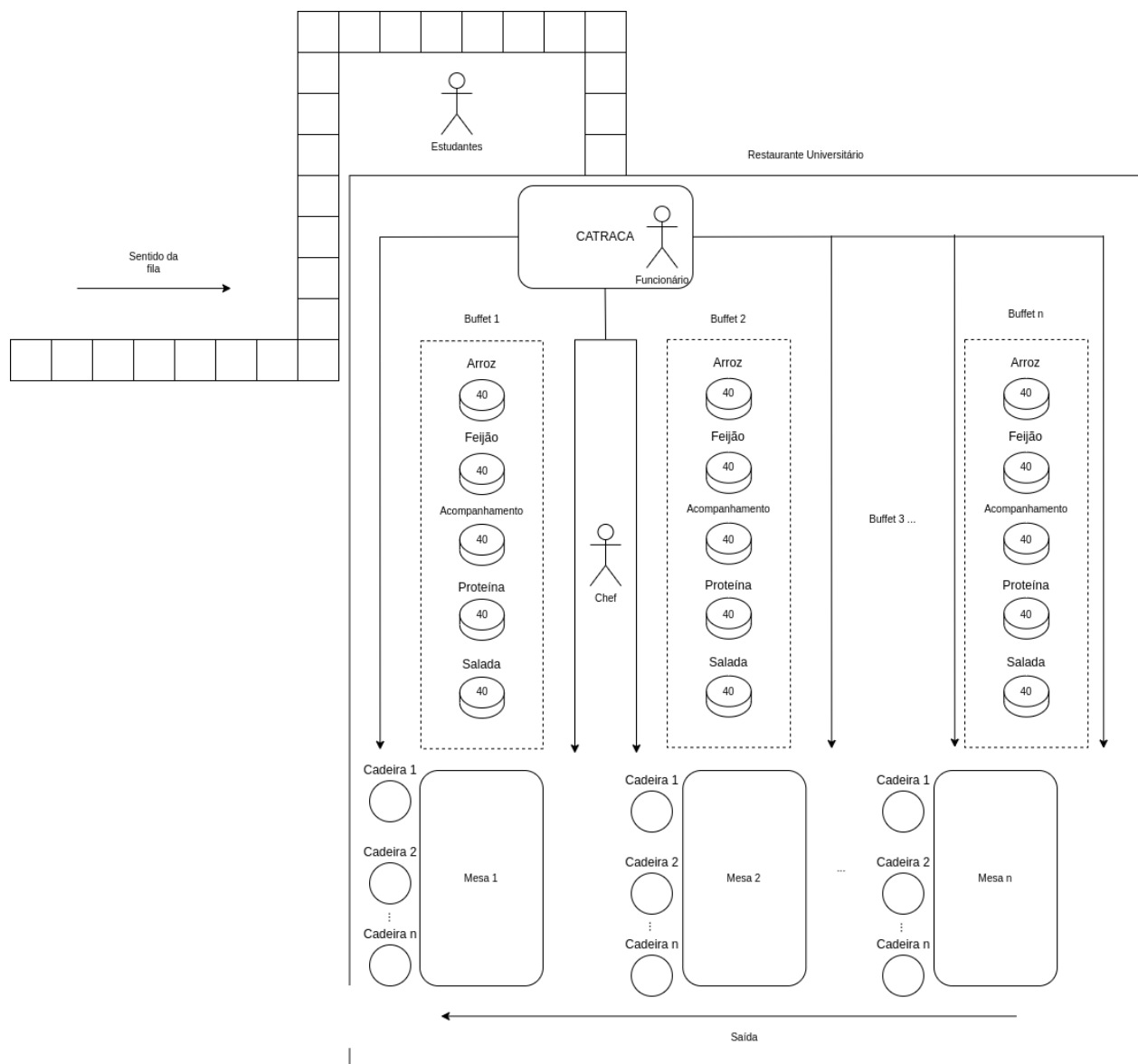
O seguinte comportamento é o descrito pelo DN para você:

- 1) O(a) aluno(a) entra em uma fila externa única para acessar o RU.
- 2) Uma catraca controla os acessos dos(as) alunos(as) ao RU. Somente um(uma) aluno(a) entra por vez no RU.
- 3) Após entrar no restaurante, o(a) aluno(a) vai para um dos *buffets* e escolhe sua refeição, entre as 5 bacias disponíveis (o DN quer simular vários cenários para saber quantos *buffets* seriam necessários construir para melhor atender aos universitários). O estudante escolhe no mínimo uma opção entre as 5 disponíveis.
- 4) Cada *buffet* possui duas filas distintas com um tamanho limitado, de 5 lugares (um lugar para cada bacia do *buffet*).
- 5) Depois de se servir, o(a) aluno(a) vai em busca de um lugar livre para se sentar. Somente após encontrar um lugar livre, o aluno(a) se senta e come por um tempo. Após comer, o(a) aluno(a) sai do restaurante.

Algumas restrições são impostas:

- Dois (duas) alunos(as) de filas diferentes no mesmo *buffet* podem pegar o mesmo tipo de comida de uma mesma bacia simultaneamente. Porém, dois alunos na mesma fila não podem pegar a mesma comida simultaneamente.
- Quando não há mais nenhum lugar em nenhuma fila dos *buffets* (ou seja, os *buffets* estão lotados), o funcionário que está na catraca não deixa nenhum aluno entrar. Quando não há mais nenhum aluno(a) na fila externa única, ele fecha a porta de entrada do restaurante e vai embora.
- Há um *chef* que sempre verifica se as bacias de comida estão vazias. Se elas estão vazias, o(a) aluno(a) que quer aquela porção deve esperar. Obviamente, isso atrasa a fila, pois, nenhum aluno(a) vai furar a fila. Após todos os(as) alunos(as) se servirem, o *chef* vai embora para preparar um novo cardápio para o dia seguinte.

É apresentado abaixo uma ilustração do funcionamento do novo complexo construído no CTC.



Há oito arquivos em C que simbolizam o RU e que foram fornecidos para você por um acadêmico que inicialmente havia aceitado implementar a simulação mas que desistiu, pois não teria tempo para finalizar o programa. Contudo, o DN pagou para ele por cada linha programada. Nesse sentido, há áreas que não necessitam ser alteradas e que estão indicadas no código. Você prometeu ao DN que não iria alterá-las.

Nesses arquivos, ele implementou partes do sistema:

- **buffet.h**
É o arquivo que contém a thread do buffet e todas as suas ações no restaurante.
- **chef.h**

É o arquivo que contém a thread do chef e todas as suas ações no restaurante.

- **config.h**

É o arquivo que contém configurações e macros para o sistema.

- **globals.h**

É o arquivo que contém as variáveis globais do sistema.

- **queue.h**

É o arquivo que contém uma lista encadeada (ligada) que armazenam estudantes (student_t).

- **student.h**

É o arquivo que representa o(a) aluno(a) e todas as suas ações no restaurante.

- **table.h**

É o arquivo que simboliza uma mesa no restaurante.

- **worker_gate.h**

É o arquivo que representa o colaborador na catraca e todas as suas ações nas filas.

No código fornecido, há também um *Makefile*. Para usá-lo no terminal, basta digitar **make** e pressionar Enter. Ele compila o programa e recebe como argumento(s):

`./program -n [número de alunos] -b [número de buffets] -t [número de mesas] -s [número de assentos por mesa]`

Atenção: O programa cria uma *thread* para cada aluno(a) e uma *thread* para cada *buffet*. Cuidado com a quantidade de *threads* que você irá colocar, pois esse programa foi criado para ser executado em um super-computador (e não em um desktop!).

O DN solicitou que esse sistema que simule o restaurante seja entregue, assim que finalizado, para que seja liberado o pagamento. O DN também solicita que o seu código seja **bem comentado**, pois o sistema passará por futuras expansões. Portanto, não se esqueça desse importante detalhe (ele será avaliado)!

Bom trabalho!