

Engenharia Informática
Sistema Operativos
2018 – 2019

Fase 3: Simulação de uma Loja



Docentes:

Eduardo Marques

Luís Gaspar

Discentes:

André Telo, nº 2042316

Lisandro Marote, nº 2030315

Funchal, 5 de Janeiro 2019

Índice

	Pág.
Introdução	3
Funcionamento da loja	4
Utilização dos mecanismos de sincronização.....	7
Políticas de acesso	8
Alguns Testes	9
Conclusão.....	12
Anexos	13
Fluxogramas.....	13
Fluxograma dos Produtos	13
Fluxograma de Empregados.....	14
Fluxograma Clientes.....	15
Código	16
Monitor.c.....	16
Simulador.c.....	25
Estatísticas.c	34
Config.conf	34

Introdução

No âmbito da cadeira de Sistemas Operativos foi-nos proposto a realização de um projeto que visa a implementação de uma simulação de uma loja, onde deverá ter em conta o número de clientes, número de empregados, artigos em stock, artigos em rutura, tempos de espera etc.

Para esta terceira fase pretende-se que o grupo implemente as bibliotecas para comunicação entre o Simulador e o Monitor, a interface onde neste deverá mostrar informação pertinente ao utilizador do estado corrente da sua simulação, bem como a implementação de semáforos e trincos em zonas pertinentes de maneira a garantir a simulação da loja corra sem percalços de maior.

O grupo delineou e analisou todos os pormenores presentes no enunciado onde recorreu-se a fluxogramas e esboços com todos os pormenores captados, fazendo chegar este ao professor Luís Gaspar (*professor prática laboratorial*) dando a sua opinião acerca do plano estipulado pelo grupo se cumpria todos os requisitos pretendidos para esta terceira fase.

Funcionamento da loja

Clientes

A loja tem uma fila de atendimento, onde nesta foi deliberadamente colocado um semáforo para delimitar o número total de clientes à espera para serem atendidos (*Fig.1*), onde por sua vez são atendidas pela ordem de chegada. Os clientes vão ter um tempo de desistência ao entrar na fila de atendimento. Segundo o código implementado este valor é fixo, onde só começa a contar após o cliente entrar na fila para atendimento, onde caso o tempo que o cliente esteja à espera exceda o estipulado este desiste e sai da loja. Os clientes vão ser gerados aleatoriamente, sendo que a cada 10 segundos podem ser gerados 1 a 5 clientes (*Dependendo dos valores presentes no ficheiro de configuração*). É também na geração aleatória do cliente que é gerada a intenção do produto que cliente pretende levantar.

Implementou-se também uma faixa etária (*Adultos ou Idosos*) de maneira a ter prioridades na fila de atendimento, ou seja, caso apareça um cliente cuja faixa etária seja idoso, este deverá ser o próximo cliente a ser atendido independentemente do número total de adultos que estão na fila para serem atendidos.

```
//#####  
//case 1:          CLIENTE ENTRA NA FILA DA LOJA.  
sem_wait(&Fila_Loja);  
sleep(1);  
pthread_mutex_lock(&Trinco_Comunicacao_Socket);  
//-----  
Cliente[ID_Cliente_Recebido].Momento_Entrada_Fila_Loja = Tempo_Simulacao_Actual;  
Tempo_Actual=Tempo_Simulacao_Actual;  
Tempo_Actual=Cliente[ID_Cliente_Recebido].Momento_Entrada_Fila_Loja; //Nao va a Hora  
//-----
```

Fig.1: Incrementação do semáforo sempre que cliente entra na fila da loja.

No caso de haver mais do que 1 empregado a servir os clientes, foi também implementado no código uma função *random* onde o cliente opta desde o empregado 1 até ao N empregado gerado (*Fig.2*).

```
// Função "Random" que irá atribuir um empregado para o Cliente.  
int Define_Empregado_Atendimento(int num_empregados){  
    srand(time(NULL));  
    int random1 = (rand()%num_empregados)+1;  
    return random1;  
}
```

Fig.2: Função *random* para atribuição de empregado a cliente.

Empregados

A loja inicialmente tem 1 empregado, cada vez que está 5 clientes na fila de atendimento é alocado um novo empregado. Quando o produto estiver abaixo de uma certa quantidade o empregado a repor será o primeiro que acabar o atendimento ao cliente, tendo os empregados a prioridade de verificar o stock e depois atender os clientes (Fig.3). Caso seja um empregado único, os clientes esperam que o empregado termine a reposição do produto.

```
//***** CLIENTE É ATENDIDO *****
//case 2:
sem_post(&Fila_Loja);
pthread_mutex_lock(&Trinco_Comunicacao_Socket);
//sem_post(&Fila_Loja);
//Diminui o Numero de Clientes na Fila Dependendo do Seu Tipo
Diminui_Tipo_Cliente_Fila_Loja(Cliente[ID_Cliente_Recebido].Faixa_Etaria);
//sem_post(&Lotacao_Loja);
//*****
if(Cliente[ID_Cliente_Recebido].produto == 1){           // Produto A.
    quantidade_prodA--;
    if (quantidade_prodA < 3){
        Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente, 8, Cliente[ID_Cliente_Recebido].ID_Cliente);
        sleep(1);
        quantidade_prodA = 10;
    }
    //sleep(2);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente, 2, Cliente[ID_Cliente_Recebido].ID_Cliente);
}
}
```

Fig.3: Verificação do stock do produto quando cliente é atendido.

Ficheiro Configuração

Foi também implementado o ficheiro de configuração onde é carregado para o simulador de maneira a poder testar diferentes ambientes de simulação. Foi também implementando como extra a verificação dos valores inseridos no ficheiro de configuração de maneira a respeitar os standards exigidos.

```
Cap_Maxima_Loja = 15
Cap_Maxima_Fila = 4
Tempo_Sim = 75
tempo_serv_prod = 80
Prob_Cliente_Idoso = 50
Total_Clientes_Gerar = 100
Tempo_Maximo_Espera_Filas = 20
Tempo_Medio_Na_Loja = 10
Tempo_Medio_Chegadas = 1
Prob_Prod_A = 30
Prob_Prod_B = 35
Prob_Prod_C = 30
```

```
// ***** Erros (Restrições) *****
if(Tempo_Medio_Chegadas <= 0){
    printf("Erro: Tempo Medio de Chegadas Tem de Ser maior que 0. \n");
    Erro = 1;
};

if(tempo_simulacao < 2 ){
    printf("Erro: Tempo de Simulacao Tem de Ser maior que 2. \n");
    Erro = 1;
};

if(Capacidade_Maxima_Loja <= 0){
    printf("Erro: Capacidade Maxima da Loja Tem de Ser maior que 0. \n");
    Erro = 1;
};
```

Fig.4: Ficheiro de configuração e restrições.

Produtos

A loja contém 3 produtos nomeadamente produto A,B e C. Cada produto vai conter inicialmente uma quantidade total de 10 unidades e um tempo de serviço de 5 segundos. Quando chega a uma quantidade menor do que 3 o primeiro funcionário que acabar de servir o cliente será o funcionário que irá repor o mesmo produto. Quando o funcionário repor o produto, este fica sempre com a quantidade total de 10.

```
//-----  
//Funcao Que Define Tipo de Produto dos Clientes Com base nas Percentagens Ligas do Ficheiro Config.  
int Define_Tipo_Produto(int Percentagem_Ser_ProdA, int Percentagem_Ser_ProdB, int Percentagem_Ser_ProdC){  
  
    srand(time(NULL));  
    int produto = 0;  
    int Random = (rand() %100)+1;           // Cria Numero Random de 1 a 100  
  
    if(Random <= Percentagem_Ser_ProdA){  
        produto = 1;                       // Produto ser A.  
    }  
    else if(Random <= Percentagem_Ser_ProdA + Percentagem_Ser_ProdB){  
        produto = 2;                       // Produto ser B.  
    }  
    else{  
        produto = 3;                       // Produto ser C.  
    }  
    return produto;  
}
```

Fig.5: Função *random* que atribui que produto cliente irá optar com base nas percentagens obtidas do ficheiro de configuração.

Os produtos serão escolhidos/gerados aleatoriamente na criação dos clientes, onde é carregado a partir do ficheiro de configuração as percentagens das probabilidades de cada produto, influenciando assim a escolha dos determinados produtos durante a simulação (*Fig.4*). Cada produto vai conter uma quantidade de stock, cada vez que o cliente pede esse produto o seu valor é decrementado até chegar ao valor em que deve ser repostos.

Utilização dos mecanismos de sincronização

O projeto contém dois semáforos, um semáforo para a fila dos clientes a serem atendidos, onde delimita o total de clientes presentes na fila, e outro semáforo à entrada da loja, onde irá delimitar o número total de clientes que entraram independentemente de estarem ou não na fila para serem atendidos de maneira a garantir que exista controlo na lotação total que a loja possui (*valor esse que poderá ser alterado com base no ficheiro de configuração*).

Cada cliente só é atendido se houver um funcionário livre (*não está a atender nem a repor*). Existe prioridades entre clientes, caso exista um cliente na fila de atendimento cuja sua faixa etária seja idoso, este irá passar para a fila dos idosos a serem atendidos, cuja este fila tem uma maior prioridade face à dos adultos (*Fig.6*).

```
//Quando o Cliente Chega à Fila da Loja É Verificado o Tipo do Cliente Que Chegou (Idoso ou Adulto).  
void Atribui_Lugar_Fila_Loja_A_Cliente(int ID_Cliente){  
  
    int i;  
  
    if(Cliente[ID_Cliente].Faixa_Etaria == 1){ //Se o Cliente Que Chegou for Idoso.  
        //Clientes_ID_Fila_Loja++;  
        Cliente[ID_Cliente].Lugar_Fila_Loja = Clientes_ID_Fila_Loja; // Vai para último lugar da fila dos Idosos.  
  
        for(i = 0; i < Clientes_Criados; i++){  
            if(Cliente[i].Faixa_Etaria != 1 && Cliente[i].Lugar_Fila_Loja > 0){ // Vai para a fila normal da loja.  
                Cliente[i].Lugar_Fila_Loja++;  
            }  
        }  
    }  
  
    else if(Cliente[ID_Cliente].Faixa_Etaria == 2){ // Se o Cliente que chegou for Adulto.  
        //Clientes_AD_Fila_Loja++;  
        Cliente[ID_Cliente].Lugar_Fila_Loja = Clientes_ID_Fila_Loja + Clientes_AD_Fila_Loja; //Vai Para o Ultimo Lugar da  
    }  
}
```

Fig.6: Atribuição de lugar na fila conforme a faixa etária do cliente.

Quando um cliente está a ser atendido por determinado empregado, este possui um trinco de maneira a garantir exclusão mútua no que toca a servir apenas um cliente de cada vez.

Políticas de acesso

A comunicação entre o Simulador e Monitor é efetuada através de sockets. No simulador é feito o output da mensagem a enviar para a socket onde por sua vez é enviada para o monitor, onde neste é interpretada fazendo por fim o print da mensagem final no monitor.



Fig.7: Menu Principal monitor.

Também foi implementado uma função que interpreta os dados obtidos a partir do simulador a fim de fazer print no monitor com base nos valores recebidos.

```
//Funcao de Leitura e Interpretacao da Mensagem Recebida Pela Socket
void Interpreta_Mensagem()
{
    int* Buffer=Recebe_Mensagem(sockfd);           //(newsockfd);
    int ID_Cliente=Buffer[0];                       //Numero Processo
    int Estado=Buffer[1];                          //Ver Tabela de Estados
    int Faixa_Etaria=Buffer[2];
    int produto=Buffer[3];                         //Produto a escolher (Produto A,B ou C)
    int stock_prod=Buffer[4];
    int ID_Empregado=Buffer[5];
    int cliente_troca=Buffer[6];

    switch(Estado)
    {
        case 1:                                     // Entra Fila da Loja
            Total_Clientes_Actualmente_Na_Fila_Loja++;

            if(Faixa_Etaria==1){ // Idosos
                Total_Clientes_Idosos_Entraram_Fila_Loja++;
            }
            else if(Faixa_Etaria==2){ // Adultos
                Total_Clientes_Adultos_Entraram_Fila_Loja++;
            }
            break;

        case 2:                                     // Sai Fila da Loja / B
            Total_Clientes_Actualmente_Na_Fila_Loja--;
```

Fig.8: Função que interpreta mensagem enviada pela socket simulador – monitor.

Alguns Testes

Aqui iremos demonstrar pequenos exemplos com print's screens das simulações efetuadas, com valores diferentes no ficheiro de configuração.

Simulação 1: Probabilidade Produto A – 100%.

```
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja

Ver Eventos

* Opcao 2 - Sair Simulacao          * Opcao 4 - Ver Estatisticas

0 Cliente Nº 1 (ADULTO) Entrou na Loja
0 Cliente Nº 1 (ADULTO) Chegou a fila
0 Cliente Nº 1 (ADULTO) foi atendido pelo empregado 1 com o produto A
0 Cliente Nº 1 (ADULTO) Saiu da Loja
0 Cliente Nº 2 (ADULTO) Entrou na Loja
0 Cliente Nº 6 (ADULTO) Entrou na Loja
0 Cliente Nº 4 (ADULTO) Chegou a fila
0 Cliente Nº 4 (ADULTO) foi atendido pelo empregado 1 com o produto A
0 Cliente Nº 4 (ADULTO) Saiu da Loja
0 Cliente Nº 5 (IDOSO) Chegou a fila
0 Cliente Nº 5 (IDOSO) foi atendido pelo empregado 1 com o produto A
0 Cliente Nº 5 (IDOSO) Saiu da Loja
0 Cliente Nº 8 (ADULTO) Entrou na Loja
0 Cliente Nº 2 (ADULTO) Chegou a fila
0 Cliente Nº 2 (ADULTO) foi atendido pelo empregado 1 com o produto A
0 Cliente Nº 2 (ADULTO) Saiu da Loja
0 Cliente Nº 10 (IDOSO) Entrou na Loja
0 Cliente Nº 12 (ADULTO) Entrou na Loja
0 Cliente Nº 6 (ADULTO) Chegou a fila
0 Cliente Nº 6 (ADULTO) foi atendido pelo empregado 1 com o produto A
2
lisandro_marote@CavaloTroia:~/Documentos/proj_fase3$
```

Simulação 2: Inserção valores incorretos no ficheiro de configuração.

```
config.conf x  estatisticas.c x

1 Cap_Maxima_Loja = -1
2 Cap_Maxima_Fila = 5
3 Tempo_Sim = 75
4 tempo_serv_prod = 10
5 Prob_Cliente_Idoso = 10
6 Total_Clientes_Gerar = 100
7 Tempo_Maximo_Espera_Filas = 20
8 Tempo_Medio_Na_Loja = 10
9 Tempo_Medio_Chegadas = 1
10 Prob_Prod_A = 30
11 Prob_Prod_B = 30
12 Prob_Prod_C = 35
13

lisandro_marote@CavaloTroia:~/Documentos/proj_fase3$ make
gcc monitor.c -lpthread -o Monitor
gcc simulador.c -lpthread -o Simulador
./Simulador
Erro: Capacidade Maxima da Loja Tem de Ser maior que 0.
```

Simulação 3: Lotação máxima da loja de 5 Clientes.

```
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja

Ver Eventos

* Opcao 2 - Sair Simulacao          * Opcao 4 - Ver Estatisticas

O Cliente Nº 1 (ADULTO) Entrou na Loja
O Cliente Nº 1 (ADULTO) Chegou a fila
O Cliente Nº 1 (ADULTO) Foi Atendido Pelo Empregado 1 com o Produto C
O Cliente Nº 1 (ADULTO) Saiu da Loja
O Cliente Nº 2 (ADULTO) Entrou na Loja
O Cliente Nº 4 (ADULTO) Entrou na Loja
O Cliente Nº 3 (ADULTO) Entrou na Loja
O Cliente Nº 2 (ADULTO) Chegou a fila
O Cliente Nº 2 (ADULTO) Foi Atendido Pelo Empregado 1 com o Produto B
O Cliente Nº 2 (ADULTO) Saiu da Loja
O Cliente Nº 5 (ADULTO) Chegou a fila
O Cliente Nº 5 (ADULTO) Foi Atendido Pelo Empregado 1 com o Produto C
O Cliente Nº 5 (ADULTO) Saiu da Loja
O Cliente Nº 3 (ADULTO) Chegou a fila
O Cliente Nº 3 (ADULTO) foi atendido pelo empregado 1 com o produto A
O Cliente Nº 3 (ADULTO) Saiu da Loja
O Cliente Nº 6 (ADULTO) Entrou na Loja
O Cliente Nº 8 (ADULTO) Entrou na Loja
O Cliente Nº 4 (ADULTO) Chegou a fila
O Cliente Nº 4 (ADULTO) Foi Atendido Pelo Empregado 1 com o Produto B
O Cliente Nº 4 (ADULTO) Saiu da Loja

lisandro_marote@CavaloTroia:~/Documentos/proj_fase3$
```

```
config.conf x estatisticas.c x

1 Cap_Maxima_Loja = 5
2 Cap_Maxima_Fila = 5
3 Tempo_Sim = 75
4 tempo_serv_prod = 10
5 Prob_Cliente_Idoso = 10
6 Total_Clientes_Gerar = 100
7 Tempo_Maximo_Espera_Filas = 20
8 Tempo_Medio_Na_Loja = 10
9 Tempo_Medio_Chegadas = 1
10 Prob_Prod_A = 30
11 Prob_Prod_B = 30
12 Prob_Prod_C = 35
```

Simulação 4: Probabilidade de ser idoso ser 80%.

```
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja

Ver Eventos

* Opcao 2 - Sair Simulacao          * Opcao 4 - Ver Estatisticas

O Cliente Nº 1 (IDOSO) Entrou na Loja
O Cliente Nº 1 (IDOSO) Chegou a fila
O Cliente Nº 1 (IDOSO) foi atendido pelo empregado 1 com o produto A
O Cliente Nº 3 (ADULTO) Entrou na Loja
O Cliente Nº 5 (IDOSO) Entrou na Loja
O Cliente Nº 3 (ADULTO) Chegou a fila
O Cliente Nº 3 (ADULTO) Foi Atendido Pelo Empregado 1 com o Produto C
O Cliente Nº 3 (ADULTO) Saiu da Loja
O Cliente Nº 6 (IDOSO) Entrou na Loja
O Cliente Nº 5 (IDOSO) Chegou a fila
O Cliente Nº 5 (IDOSO) Foi Atendido Pelo Empregado 1 com o Produto B
O Cliente Nº 5 (IDOSO) Saiu da Loja
O Cliente Nº 8 (IDOSO) Entrou na Loja
O Cliente Nº 2 (IDOSO) Chegou a fila
O Cliente Nº 2 (IDOSO) foi atendido pelo empregado 1 com o produto A
O Cliente Nº 2 (IDOSO) Saiu da Loja
O Cliente Nº 6 (IDOSO) Chegou a fila
O Cliente Nº 6 (IDOSO) foi atendido pelo empregado 1 com o produto A
O Cliente Nº 6 (IDOSO) Saiu da Loja
```

```
config.conf x estatisticas.c

1 Cap_Maxima_Loja = 15
2 Cap_Maxima_Fila = 5
3 Tempo_Sim = 75
4 tempo_serv_prod = 10
5 Prob_Cliente_Idoso = 80
6 Total_Clientes_Gerar = 100
7 Tempo_Maximo_Espera_Filas = 20
8 Tempo_Medio_Na_Loja = 10
9 Tempo_Medio_Chegadas = 1
10 Prob_Prod_A = 30
11 Prob_Prod_B = 30
12 Prob_Prod_C = 35
```

Simulação 5: Tempo Máximo espera na fila de 5 segundos.

```
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja

Ver Eventos

* Opcao 2 - Sair Simulacao          * Opcao 4 - Ver Estatisticas

0 Cliente Nº 1 (ADULTO) Entrou na Loja
0 Cliente Nº 2 (ADULTO) Entrou na Loja
0 Cliente Nº 1 (ADULTO) Chegou a fila
0 Cliente Nº 3 (IDOSO) Entrou na Loja
0 Cliente Nº 5 (ADULTO) Entrou na Loja
0 Cliente Nº 4 (ADULTO) Entrou na Loja
0 Cliente Nº 3 (IDOSO) Chegou a fila
0 Cliente Nº 3 (IDOSO) foi atendido pelo empregado 1 com o produto A
0 Cliente Nº 3 (IDOSO) Saiu da Loja
0 Cliente Nº 5 (ADULTO) Chegou a fila
0 Cliente Nº 7 (ADULTO) Entrou na Loja
0 Cliente Nº 4 (ADULTO) Desistiu
0 Cliente Nº 4 (ADULTO) Saiu da Loja
0 Cliente Nº 8 (IDOSO) Entrou na Loja
0 Cliente Nº 1 (ADULTO) Saiu da Loja
0 Cliente Nº 13 (ADULTO) Chegou a fila
0 Cliente Nº 5 (ADULTO) Desistiu
0 Cliente Nº 5 (ADULTO) Saiu da Loja
0 Cliente Nº 11 (ADULTO) Entrou na Loja
0 Cliente Nº 2 (ADULTO) Desistiu
0 Cliente Nº 2 (ADULTO) Saiu da Loja
0 Cliente Nº 11 (ADULTO) Chegou a fila
0 Cliente Nº 21 (ADULTO) Entrou na Loja
0 Cliente Nº 9 (ADULTO) Entrou na Loja
0 Cliente Nº 8 (IDOSO) Chegou a fila
0 Cliente Nº 8 (IDOSO) Foi Atendido Pelo Empregado 2 com o Produto B
0 Cliente Nº 8 (IDOSO) Saiu da Loja
0 Cliente Nº 16 (ADULTO) Entrou na Loja
```

```
config.conf x  estatisticas.

1 Cap_Maxima_Loja = 20
2 Cap_Maxima_Fila = 10
3 Tempo_Sim = 75
4 tempo_serv_prod = 10
5 Prob_Cliente_Idoso = 50
6 Total_Clientes_Gerar = 100
7 Tempo_Maximo_Espera_Filas = 5|
8 Tempo_Medio_Na_Loja = 10
9 Tempo_Medio_Chegadas = 1
10 Prob_Prod_A = 30
11 Prob_Prod_B = 30
12 Prob_Prod_C = 35
```


Conclusão

Ao longo da elaboração desta fase do projeto constatou-se algumas dificuldades ao tentar implementar tudo o que era pedido / pretendido para esta terceira fase, tendo o grupo optado por dar prioridade ao fundamental desta fase que seria ter a sincronização (semáforos e trincos) a funcionar para os conceitos base para a simulação da loja online.

Não foi possível implementar todas as funcionalidades que o grupo tinha planeado, nomeadamente à possível troca que um cliente poderia efetuar na entrega de um produto, também não foi possível implementar um semáforo de acesso aos produtos para cada cliente, de maneira a garantir caso exista rutura de um determinado produto, o acesso a este fica condicionado até que seja repostado entre outros pequenos pormenores à qual não tivemos tempo de implementar.

Nesta terceira fase alteramos por completo a interface da loja online de maneira a estar mais intuitiva e objetiva, estando a decorrer fluentemente ao longo da simulação e a obter informações quer a nível estatísticos quer a nível do comportamento da loja em tempo real. Por fim também se implementou a gravação dos dados estatísticos num ficheiro onde sua interface aproxima-se quanto possível à da simulação.

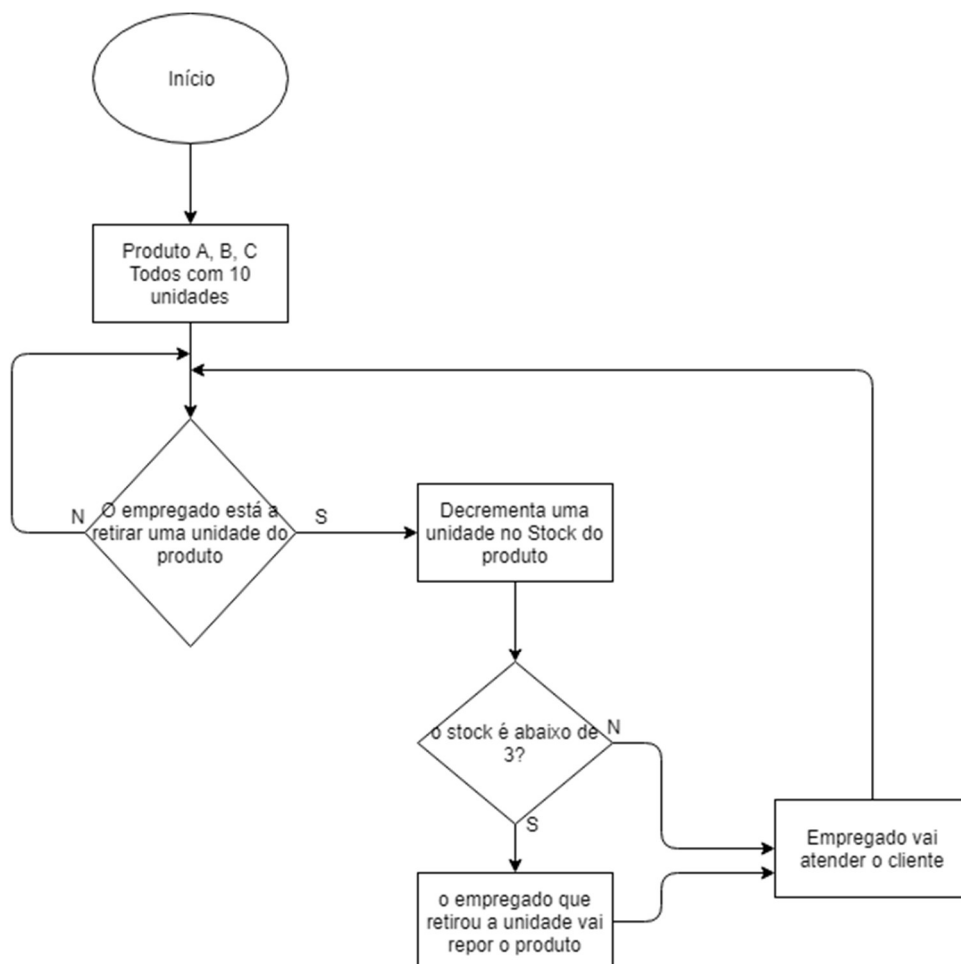
Uma das maiores dificuldades sentidas nesta terceira fase foi o tempo necessário de maneira a obter tudo o que era pedido segundo o enunciado, todas as funcionalidades do projeto a funcionar, fluxogramas, relatórios etc devido à carga excessiva de trabalhos, relatórios apresentações que foi imposta neste semestre em todas as cadeiras sendo extremamente complicado sincronizar tempo para todas estas, fazendo com que em último caso o grupo entregasse apenas o “básico” pretendido para esta fase de maneira a garantir que não saísse prejudicado.

Concluindo, com este projeto adquirimos mais experiência acerca dos mecanismos de sincronização (semáforos e trincos) bem como um contacto mais próximo com a linguagem de programação C, onde tivemos oportunidade de por em prática todos os conhecimentos captados das aulas práticas-laboratoriais de Sistemas Operativos

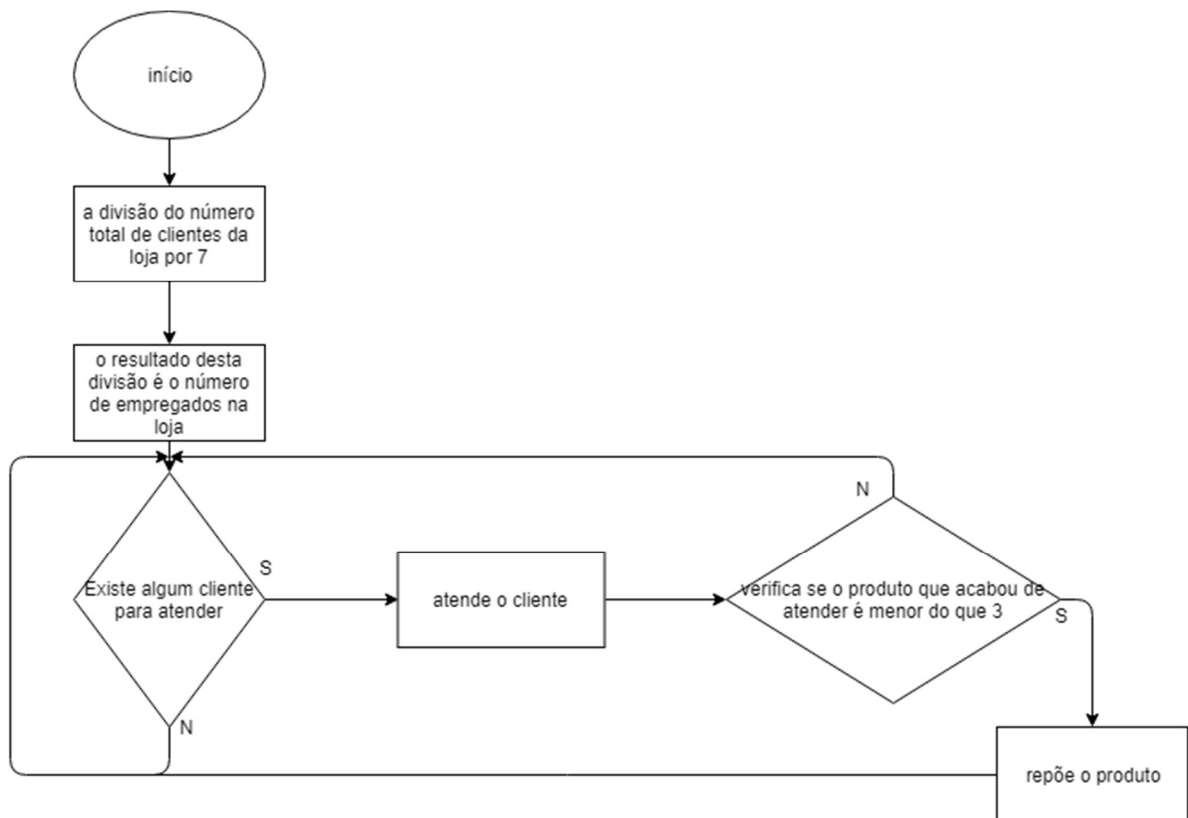
Anexos

Fluxogramas

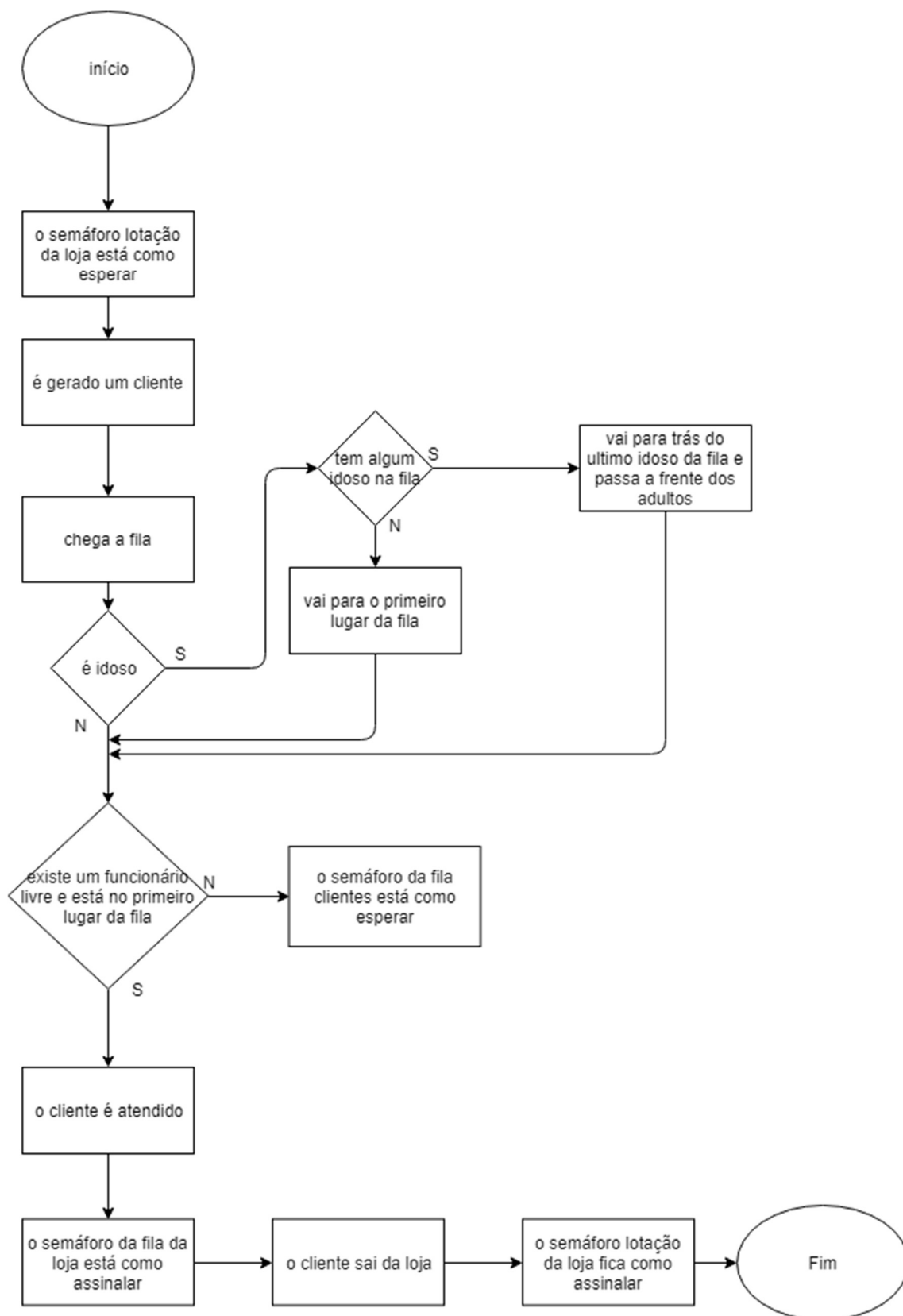
Fluxograma dos Produtos



Fluxograma de Empregados



Fluxograma Clientes



Código

Monitor.c

```
//Chamada das Livrarias.
#include "unix.h"

int sockfd;
int newsockfd;
pthread_t Thread_Recebe_Mensagens;

//-----
//Variaveis utilizadas para execucao da Simulacao:
bool Simulacao_Em_Curso =false;// Indica se a simulacao está em curso.
int Opcao_Utilizador =0;// Variavel para armazenar as opções selecionadas pelo
utilizador.
//-----
//Variaveis a Receber do Simulador.
int Capacidade_Maxima_Loja =0;

//-----
//Funcao Para Ler do Ficheiro Config:
void Leitura_Ficheiro_Config()
{
//Declaracao e Inicializacao de Variaveis Auxiliares da Funcao
char Le_Linha[100];
int Valor_Variavel_Lida;
char Nome_Variavel_Lida[100];
char Var1[]="Cap_Maxima_Loja";// Capacidade Máxima da Loja.

FILE*fp;
fp=fopen("config.conf","r");//Vai Abrir o Ficheiro Config.conf.

if(fp!=NULL){
//Ciclo de Leitura:
while(fgets(Le_Linha,100,fp)!=NULL){
sscanf(Le_Linha,"%s = %i", Nome_Variavel_Lida,&Valor_Variavel_Lida);

if(strcmp(Nome_Variavel_Lida,Var1)==0){
Capacidade_Maxima_Loja=Valor_Variavel_Lida;
}
}

fclose(fp);
}
else{
printf("Nao Foi Possivel Abrir o Ficheiro Configuracao.");
}
}

//-----
// Declaração das Variáveis Globais
int Total_Clientes_Idosos_Entraram_Fila_Loja;
int Total_Clientes_Idosos_Atendidos_Loja;
int Total_Clientes_Idosos_Desistiram_Fila_Loja;
int Total_Clientes_Adultos_Entraram_Fila_Loja;
int Total_Clientes_Adultos_Atendidos_Loja;
int Total_Clientes_Adultos_Desistiram_Fila_Loja;
int Total_Clientes_Actualmente_Na_Fila_Loja;
int Total_Clientes_Actualmente_Na_Loja;
int Total_Clientes_Sairam_Loja;
int quantidade_prodB;
int quantidade_prodB;
int quantidade_prodB;
int total_vendas_A =0;
int total_vendas_B =0;
```

```

int total_vendas_C =0;

//-----
// Interface do Menu Principal (Monitor).
void Menu_Interface(){
system("clear");
printf("\n");
printf("      Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja\n");
printf("\n");
printf("      Dados da Simulacao\n");
printf("\n");
printf("      * Opcao 1 - Iniciar Simulacao\n");
printf("      * Opcao 2 - Sair\n");
printf("      * Opcao 3 - Ver Eventos\n");
printf("      * Opcao 4 - Ver Estatisticas\n");
printf("\n");
printf("\n");
printf("\n");
printf("\n");
printf("\n");
printf("\n");
printf("\n");
}

//-----
//Funcao Para Preparar Ficheiro do Log Eventos (Invocado Quando o Utilizador Escolhe a
Opcao de Simulacao no Menu Principal):
void Prepara_Ficheiro_Eventos()
{
FILE*fp;
fp=fopen("relatorio eventos.c","w");

fprintf(fp,"Eventos Loja: \n");
fprintf(fp,"\n");

fclose(fp);

//Funcao Para Escrever Eventos no Ficheiro Log (Precisamos da Funcao de "Preparacao"
Anterior Pois Esta Funcao Vai Ser Invocada Sempre Que Houver um Evento. Dessa Forma
Temos de Usar Append em Vez de Write, Senao Apagava Sempre o Conteudo do Ficheiro):
void Escreve_Ficheiro_Eventos(int Evento,int ID_Cliente,int Faixa_Etaria)
{
char*Nome Faixa Etaria;
char Tipo_Nome_Faixa_Etarial[]="(IDOSO)";
char Tipo_Nome_Faixa_Etaria2[]="(ADULTO)";

if(Faixa_Etaria==1){
Nome_Faixa_Etaria=Tipo_Nome_Faixa_Etarial;
}
elseif(Faixa_Etaria==2){
Nome_Faixa_Etaria=Tipo_Nome_Faixa_Etaria2;
}

FILE*fp;
fp=fopen("relatorio_eventos.c","a");//"a" = Append (Ou Seja, a Linha é Adicionada ao
Ficheiro Sem Eliminar Nada)

switch(Evento)
{
case1:
fprintf(fp,"->Cliente Nº %i (%s) Entrou na Loja \n", ID_Cliente+1,Nome_Faixa_Etaria);
fprintf(fp,"-----\n");
break;

case2:
fprintf(fp,"->Cliente Nº %i Chegou a fila \n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case3:
fprintf(fp,"->Cliente Nº %i Foi atendido com o produto A\n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case4:

```

```
fprintf(fp,"->Cliente N° %i Desistiu da Fila \n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case5:
fprintf(fp,"->Cliente N° %i Saiu da Loja \n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case6:
fprintf(fp,"->Cliente N° %i Foi atendido com o produto B\n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case7:
fprintf(fp,"->Cliente N° %i Foi atendido com o produto C\n", ID_Cliente+1);
fprintf(fp,"-----\n");
break;

case8:
fprintf(fp,"O Empregado N°1 foi repor o produto A");
fprintf(fp,"-----\n");
break;

case9:
printf("| O Empregado N°1 foi repor o produto B\n");
fprintf(fp,"-----\n");
break;

case10:
printf("| O Empregado N°1 foi repor o produto C\n");
fprintf(fp,"-----\n");
break;

case50:
fprintf(fp,"*Simulacao Terminada* \n");
fprintf(fp,"\n");
fprintf(fp,"-----\n");
break;

default:
fprintf(fp,"-Erro: Evento Nao Definido\n");
fprintf(fp,"-----\n");
}

fclose(fp);
}

//-----
//Funcao que Mostra o Log de Eventos em Tempo Real
void Mostra_No_Monitor_Eventos_Tempo_Real(int Evento,int ID_Cliente,int Faixa_Etaria,int
ID_employado)
{
char*Nome_Faixa_Etaria;
char Tipo_Nome_Faixa_Etaria1[]="(IDOSO)";
char Tipo_Nome_Faixa_Etaria2[]="(ADULTO)";

if(Faixa_Etaria ==1){
Nome_Faixa_Etaria = Tipo_Nome_Faixa_Etaria1;
}

elseif(Faixa_Etaria ==2){
Nome_Faixa_Etaria = Tipo_Nome_Faixa_Etaria2;
}

switch(Evento)
{
case1:
printf("| O Cliente N° %d %s Chegou a fila \n", ID_Cliente+1,Nome_Faixa_Etaria);
break;

case2:
printf("| O Cliente N° %d %s foi atendido pelo empregado %d com o produto A\n",
ID_Cliente+1, Nome_Faixa_Etaria, ID_employado);
break;

case3:
```

```
printf("| O Cliente Nº %d %s Desistiu\n", ID_Cliente+1, Nome_Faixa_Etaria);
break;

case4:
printf("| O Cliente Nº %d %s Saiu da Loja \n", ID_Cliente+1, Nome_Faixa_Etaria);
break;

case5:
printf("| O Cliente Nº %d %s Entrou na Loja \n", ID_Cliente+1, Nome_Faixa_Etaria);
break;

case6:
printf("| O Cliente Nº %d %s Foi Atendido Pelo Empregado %d com o Produto B\n",
ID_Cliente+1, Nome_Faixa_Etaria, ID_empregado);
break;

case7:
printf("| O Cliente Nº %d %s Foi Atendido Pelo Empregado %d com o Produto C\n",
ID_Cliente+1, Nome_Faixa_Etaria, ID_empregado);
break;

case8:
printf("| O Empregado Nº %d Foi Repor o Produto A\n", ID_empregado);
break;

case9:
printf("| O Empregado Nº %d Foi Repor o Produto B\n", ID_empregado);
break;

case10:
printf("| O Empregado Nº %d Foi Repor o Produto C\n", ID_empregado);
break;

case50:
printf("| Simulacao Terminada. \n");
break;

default:
printf("| Erro: Caso Nao Definido. \n");
}
}

//-----
//Funcao Para Mostrar as Estatisticas da simulacao no Monitor:
void Mostra_Estatisticas_Monitor(int Total_Clientes_Idosos_Entraram_Fila,int
Total_Clientes_Idosos_Atendidos,int Total_Clientes_Idosos_Desistiram_Fila,int
Total_Clientes_Adultos_Entraram_Fila,int Total_Clientes_Adultos_Atendidos,int
Total_Clientes_Adultos_Desistiram_Fila,int Total_Clientes_Sairam_Loja)
{
    system("clear");
    printf("
    printf("|      Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja      |\n");
    printf("
    printf("|      Estatisticas Simulacao      |\n");

    // Variáveis Que calculam os Totais de cada parametro (Clientes).
    int Total_Clientes_Atendidos = Total_Clientes_Idosos_Atendidos +
    Total_Clientes_Adultos_Atendidos;
    int Total_Clientes_Entraram_Fila = Total_Clientes_Idosos_Entraram_Fila +
    Total_Clientes_Adultos_Entraram_Fila;
    int Total_Clientes_Desistiram_Fila = Total_Clientes_Idosos_Desistiram_Fila +
    Total_Clientes_Adultos_Desistiram_Fila;

    printf("
    printf("|* Total Clientes Atendidos Loja: %d |\n",
    Total_Clientes_Atendidos);
    printf("|* Total Clientes Sairam Loja: %d |\n",
    Total_Clientes_Sairam_Loja );
    printf("|* Total Clientes Entraram Fila Loja: %d |\n",
    Total_Clientes_Entraram_Fila);
    printf("|* Total Clientes Desistiram Fila Loja: %d |\n",
    Total_Clientes_Desistiram_Fila);
    printf("|* Total Vendas Produto A: %d |\n",
    total_vendas_A);
    printf("|* Total Vendas Produto B: %d |\n",
    total_vendas_B);
```

```
printf("|* Total Vendas Produto C: %d\n",
total_vendas_C);
printf("|\n");
printf("* Opcao 2 - Sair Simulacao * Opcao 3 - Ver Eventos\n");
}

void Escreve_Estatisticas(int Total_Clientes_Idosos_Entraram_Fila,int
Total_Clientes_Idosos_Atendidos,int Total_Clientes_Idosos_Desistiram_Fila,int
Total_Clientes_Adultos_Entraram_Fila,int Total_Clientes_Adultos_Atendidos,int
Total_Clientes_Adultos_Desistiram_Fila,int Total_Clientes_Sairam_Loja)
{
// Variáveis Que calculam os Totais de cada parametro (Clientes).
int Total_Clientes_Atendidos = Total_Clientes_Idosos_Atendidos +
Total_Clientes_Adultos_Atendidos;
int Total_Clientes_Entraram_Fila = Total_Clientes_Idosos_Entraram_Fila +
Total_Clientes_Adultos_Entraram_Fila;
int Total_Clientes_Desistiram_Fila = Total_Clientes_Idosos_Desistiram_Fila +
Total_Clientes_Adultos_Desistiram_Fila;

FILE*fp;
fp=fopen("estatisticas.c","w");

fprintf(fp,"| Estatisticas |\n");
fprintf(fp,"|\n");
fprintf(fp,"|\n");
fprintf(fp,"| Clientes Atendidos na Loja |\n");
fprintf(fp,"|\n");
fprintf(fp,"* Total Clientes Atendidos: %d\n", Total_Clientes_Atendidos);
fprintf(fp,"* Total Clientes Idosos Atendidos: %d\n", Total_Clientes_Idosos_Atendidos);
fprintf(fp,"* Total Clientes Adultos Atendidos: %d\n",
Total_Clientes_Adultos_Atendidos);

fprintf(fp,"|\n");
fprintf(fp,"| Clientes Entraram na Fila da Loja |\n");
fprintf(fp,"|\n");
fprintf(fp,"* Total Clientes Entraram Fila: %d\n", Total_Clientes_Entraram_Fila);
fprintf(fp,"* Total Clientes Idosos Entraram Fila: %d\n",
Total_Clientes_Idosos_Entraram_Fila_Loja);
fprintf(fp,"* Total Clientes Adultos Entraram Fila: %d\n",
Total_Clientes_Adultos_Entraram_Fila);

fprintf(fp,"|\n");
fprintf(fp,"| Clientes Desistiram da Fila da Loja |\n");
fprintf(fp,"|\n");
fprintf(fp,"* Total Clientes Desistiram da Fila: %d\n",
Total_Clientes_Desistiram_Fila);
fprintf(fp,"* Total Clientes Idosos Desistiram da Fila: %d\n",
Total_Clientes_Idosos_Desistiram_Fila);
fprintf(fp,"* Total Clientes Adultos Desistiram da Fila: %d\n",
Total_Clientes_Adultos_Desistiram_Fila_Loja);

fprintf(fp,"|\n");
fprintf(fp,"| Vendas de Produtos |\n");
fprintf(fp,"|\n");
fprintf(fp,"* Total de Vendas Produto A: %d\n", total_vendas_A);
fprintf(fp,"* Total de Vendas Produto B: %d\n", total_vendas_B);
fprintf(fp,"* Total de Vendas Produto C: %d\n", total_vendas_C);
fprintf(fp,"|\n");
fprintf(fp,"|\n");

fprintf(fp,"Dados Impressos Terminados.\n");
fprintf(fp,"|\n");

fclose(fp);
}

//-----
//Funcao Que Recebe Mensagem da Socket e Passa Cada Campo da Mensagem Para Uma Variavel
Propria
```

```

int* Recebe_Mensagem(int sockfd)
{
    int Mensagem_Recebida=0;
    int ID_Cliente=0;
    int Estado=0;
    int Faixa_Etaria=0;
    int produto=0;
    int desistencias=0;
    int pac_min=0;
    int pac_max=0;
    int stock_prod=0;
    int ID_Empregado=0;
    int cliente_troca=0;

    char recvline[MAX_LINE+1];
    static int Mensagem[9]; // Conversao dos Valores da Mensagem Para Array a Ser Enviado Para Interpretacao

    Mensagem_Recebida=read(sockfd, recvline, MAX_LINE);

    if(Mensagem_Recebida < 0) perror("Erro ao ler mensagem no servidor\n");
    recvline[Mensagem_Recebida]=0;

    sscanf(recvline,"%d %d %d %d %d %d %d %d %d %d",
    &ID_Cliente,&Estado,&Faixa_Etaria,&produto,&desistencias,&pac_min,&pac_max,&stock_pr
    od,&ID_Empregado,&cliente_troca);

    Mensagem[0]= ID_Cliente;
    Mensagem[1]= Estado;
    Mensagem[2]= Faixa_Etaria;
    Mensagem[3]= produto;
    Mensagem[4]= desistencias;
    Mensagem[5]= pac_min;
    Mensagem[6]= pac_max;
    Mensagem[7]= stock_prod;
    Mensagem[8]= ID_Empregado;
    Mensagem[9]= cliente_troca;

    return Mensagem;
}

//-----
// Funcao de Leitura e Interpretacao da Mensagem Recebida Pela Socket
void Interpreta_Mensagem()
{
    int* Buffer=Recebe_Mensagem(sockfd); // (newsockfd);
    int ID_Cliente=Buffer[0]; // Numero Processo
    int Estado=Buffer[1]; // Ver Tabela de Estados
    int Faixa_Etaria=Buffer[2];
    int produto=Buffer[3]; // Produto a escolher (Produto A,B ou C)
    int stock_prod=Buffer[4];
    int ID_Empregado=Buffer[5];
    int cliente_troca=Buffer[6];

    switch(Estado)
    {
        case 1: // Entra Fila da Loja
            Total_Clientes_Actualmente_Na_Fila_Loja++;

            if(Faixa_Etaria==1) { // Idosos
                Total_Clientes_Idosos_Entraram_Fila_Loja++;
            }
            elseif(Faixa_Etaria==2) { // Adultos
                Total_Clientes_Adultos_Entraram_Fila_Loja++;
            }
            break;

        case 2: // Sai Fila da Loja / É atendido com o produto A
            Total_Clientes_Actualmente_Na_Fila_Loja--;
            quantidade_prodA--;
            total_vendas_A++;

            if(Faixa_Etaria ==1) { // Idosos
                Total_Clientes_Idosos_Atendidos_Loja++;
            }
    }
}

```

```
elseif(Faixa_Etaria ==2){ // Adultos
    Total_Clientes_Adultos_Atendidos_Loja++;
}
break;

case3:// Desiste Fila Loja
    Total_Clientes_Actualmente_Na_Fila_Loja--;
if(Faixa_Etaria ==1){ // Idosos
    Total_Clientes_Idosos_Desistiram_Fila_Loja++;
    Total_Clientes_Sairam_Loja++;
}
elseif(Faixa_Etaria ==2){ // Adultos
    Total_Clientes_Adultos_Desistiram_Fila_Loja++;
    Total_Clientes_Sairam_Loja++;
}
break;

case4:// Sai Loja
    Total_Clientes_Actualmente_Na_Loja--;
    Total_Clientes_Sairam_Loja++;

break;

case5:// Entra Loja
    Total_Clientes_Actualmente_Na_Loja++;

break;

case6:// Sai Fila da Loja / É atendido com o produto B
    Total_Clientes_Actualmente_Na_Fila_Loja--;
    quantidade_prodB--;
    total_vendas_B++;

if(Faixa_Etaria ==1){ // Idosos
    Total_Clientes_Idosos_Atendidos_Loja++;
}
elseif(Faixa_Etaria ==2){ // Adultos
    Total_Clientes_Adultos_Atendidos_Loja++;
}
break;

case7:// Sai Fila da Loja / É atendido com o produto C
    Total_Clientes_Actualmente_Na_Fila_Loja--;
    quantidade_prodC--;
    total_vendas_C++;

if(Faixa_Etaria ==1){ // Idosos
    Total_Clientes_Idosos_Atendidos_Loja++;
}
elseif(Faixa_Etaria ==2){ // Adultos
    Total_Clientes_Adultos_Atendidos_Loja++;
}
break;

case8:// Sai Fila da Loja / É atendido com o produto C
    quantidade_prodA =10;

break;

case9:// Sai Fila da Loja / É atendido com o produto C
    quantidade_prodB =10;

break;

case10:// Sai Fila da Loja / É atendido com o produto C
    quantidade_prodC =10;

break;

case50:// Fim Simulação
    Simulacao_Em_Curso = false;
// Escreve dados das estatísticas quando a simulação termina.
    Escreve_Estatisticas(Total_Clientes_Idosos_Entraram_Fila_Loja,
    Total_Clientes_Idosos_Atendidos_Loja, Total_Clientes_Idosos_Desistiram_Fila_Loja,
    Total_Clientes_Adultos_Entraram_Fila_Loja, Total_Clientes_Adultos_Atendidos_Loja,
    Total_Clientes_Adultos_Desistiram_Fila_Loja, Total_Clientes_Sairam_Loja);
break;

default:printf("Erro: Estado do Processo Errado. \n");
break;

} //Fim Switch
```



```

if(Opcao_Utilizador ==4){// Ver estatísticas em Tempo real da simulação.
    Mostra_Estatisticas_Monitor(Total_Clientes_Idosos_Entraram_Fila_Loja,
    Total_Clientes_Idosos_Atendidos_Loja, Total_Clientes_Idosos_Desistiram_Fila_Loja,
    Total_Clientes_Adultos_Entraram_Fila_Loja, Total_Clientes_Adultos_Atendidos_Loja,
    Total_Clientes_Adultos_Desistiram_Fila_Loja, Total_Clientes_Sairam_Loja);
}
elseif(Opcao_Utilizador==3){// Ver os eventos da simulação em tempo real.
    Mostra_No_Monitor_Eventos_Tempo_Real(Estado, ID_Cliente, Faixa_Etaria,
    ID_Empregado);
}
}

//-----
//Funcao da tarefa Que Recebe Mensagens.
void Activa_Ciclo_Recebe_Mensagens(){

while(Simulacao_Em_Curso!=false){
    Interpreta_Mensagem();
}
}

//-----
//Menu Principal:
void Menu_Principal(){

if(Opcao_Utilizador!=3){
printf("Opcao Seleccionada: ");
}
//verifica a escolha do menus
scanf("%d",&Opcao_Utilizador);

switch(Opcao_Utilizador){

case1://Se Nao Houver Nenhuma Simulacao Activa, começa uma nova.
if(Simulacao_Em_Curso == false){

        Prepara_Ficheiro_Eventos();
        system("clear");

printf("
printf(" |      Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja      | \n");

printf(" |_____| \n");
printf(" |              Simulacao Iniciada              | \n");

printf(" |_____| \n");
printf(" | * Opcao 2 - Sair                               | \n");
printf(" | * Opcao 3 - Ver Eventos                         | \n");
printf(" | * Opcao 4 - Ver Estatisticas                    | \n");

printf(" |_____| \n");

        Simulacao_Em_Curso = true;

}

if(pthread_create(&(Thread_Recebe_Mensagens),NULL,(void*)&Activa_Ciclo_Recebe_Mensagens,
NULL)!=0){
printf("Erro na criacao da tarefa\n");
exit(1);
}
}
else{//Se Houver uma Simulacao Activa Devolve Erro.
printf("Erro: Simulacao Activa Neste Momento \n");
}
break;

case2://Escreve Estatisticas (do Momento em Que a Simulacao Foi Interrompida) no
Ficheiro
if(Simulacao_Em_Curso == true){
        Escreve_Estatisticas(Total_Clientes_Idosos_Entraram_Fila_Loja,
        Total_Clientes_Idosos_Atendidos_Loja, Total_Clientes_Idosos_Desistiram_Fila_Loja,
        Total_Clientes_Adultos_Entraram_Fila_Loja, Total_Clientes_Adultos_Atendidos_Loja,
        Total_Clientes_Adultos_Desistiram_Fila_Loja, Total_Clientes_Sairam_Loja);
}
}

```

```

        close(newsockfd);
        close(sockfd);

exit(0);
break;

case3://Ver Eventos em tempo Real
if(Simulacao_Em_Curso == true){
    system("clear");

printf("
Sistemas Operativos 2018 - 2019 - Simulacao de uma Loja\n");
printf("
Ver Eventos\n");

printf("
Opcao 2 - Sair Simulacao      * Opcao 4 - Ver Estatisticas\n");
printf("
}
else{
printf("Erro: Nenhuma Simulacao Activa Neste Momento.\n");
}
break;

case4://Funcao Para Mostrar as Estatisticas da Feira no Monitor:
if(Simulacao_Em_Curso == true){

Mostra_Estatisticas_Monitor(Total_Clientes_Idosos_Entraram_Fila_Loja,
Total_Clientes_Idosos_Atendidos_Loja, Total_Clientes_Idosos_Desistiram_Fila_Loja,
Total_Clientes_Adultos_Entraram_Fila_Loja, Total_Clientes_Adultos_Atendidos_Loja,
Total_Clientes_Adultos_Desistiram_Fila_Loja, Total_Clientes_Sairam_Loja);
}
else{
printf("Erro: Nenhuma Simulacao Activa Neste Momento. Selecione Outra Opcao. \n");
}
break;

default:printf("Erro: Opcao Inserida Invalida. \n");
break;
}
}
//-----
intmain(){

    Leitura_Ficheiro_Confg();
    int servlen;
    struct sockaddr_un serv_addr;

    if((sockfd=socket(AF_UNIX, SOCK_STREAM,0))<0){
        perror("Falha ao criar socket Cliente.n");
    }

    bzero((char*)&serv_addr,sizeof(serv_addr));
    serv_addr.sun_family = AF_UNIX;

    strcpy(serv_addr.sun_path, UNIXSTR_PATH);
    servlen = strlen(serv_addr.sun_path)+sizeof(serv_addr.sun_family);

    if(connect(sockfd,(struct sockaddr *)&serv_addr, servlen)<0){
        perror("Falha ao Estabelecer Conexao.\n");
    }

    system("clear");
    Menu_Interface();
    while(Opcao_Utilizador!=2){
        Menu_Principal();
    }
    return0;
}

```

Simulador.c

```
// Chamada das bibliotecas.
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "unix.h"
#include <unistd.h>

//-----
// Declaração das Variáveis Globais
int Tempo_Medio_Chegadas =0;
int Capacidade_Maxima_Loja =0;
int Capacidade_Maxima_Fila_Loja =0;
int tempo_simulacao =0;
int tempo_serv_prod =0;
int Percentagem_Ser_Cliente_Idoso =0;
int Numero_Clientes_A_Criar =0;
int Tempo_Maximo_Espera_Filas =0;
int Tempo_Medio_Na_Loja =0;

int Percentagem_Ser_ProdA =0;
int Percentagem_Ser_ProdB =0;
int Percentagem_Ser_ProdC =0;

int quantidade_prodA =10;
int quantidade_prodB =10;
int quantidade_prodC =10;
int tipo_produto;

int newsockfd;
int sockfd;
int Fecha_Loja =0;//Quando Todos os Cliente Forem Embora, Acaba a Thread da loja.
int Valor_Semaforo_fila;
int Pessoas_No_Semaforo_fila =0;
int IDEmpregado =1;

//-----
//Funcao Que Recebe o Valor Inserido (Tempo Medio de Qualquer Coisa) no Ficheiro Config
e Devolve Valor Random Baseado no Valor Lido
int Calcula_Tempos_Medios_Random(int Valor_Lido_Config){

    srand(time(NULL));
    int Desvio;
    int Random;
    int Valor_Lido_Mais_Um;//Para Ser, Por Exemplo, de 0 a 10 e nao de 0 a 9

    if(Valor_Lido_Config >1){//Se For Impar Arredondar Para Tras. Isto é, 1.5->1
        Desvio = Valor_Lido_Config/2;
        Valor_Lido_Mais_Um = Valor_Lido_Config+1;
        Random =(rand()%Valor_Lido_Mais_Um)+Desvio;
    }
    elseif(Valor_Lido_Config ==1){//Caso Especial Se For 1. Temos de Deixar o Random a 1
        Random =1;
    }
    return Random;
}

//-----
//Funcao Que Define Tipo dos Clientes Baseado nas Percentagens Lidas do Ficheiro Config.
int Define_Tipo_Cliente_Random(int Percentagem_Ser_Cliente_Idoso){

    srand(time(NULL));
    int Faixa_Etaria =0;
    int Random=(rand()%100)+1;//Cria Numero Random de 1 a 100

    if(Random <= Percentagem_Ser_Cliente_Idoso){
        Faixa_Etaria =1;// Idoso.
    }
    else{
        Faixa_Etaria=2;// Adulto.
    }
    return Faixa_Etaria;
}
```

```
//-----
//Funcao Que Define Tipo Tipo de Produto dos Clientes Com base nas Percentagens Ligas do
Ficheiro Config.
int Define_Tipo_Produto(int Percentagem_Ser_ProdA,int Percentagem_Ser_ProdB,int
Percentagem_Ser_ProdC){

    srand(time(NULL));
int produto =0;
int Random =(rand()%100)+1;// Cria Numero Random de 1 a 100

if(Random <= Percentagem_Ser_ProdA){
    produto =1;// Produto ser A.
}
elseif(Random <= Percentagem_Ser_ProdA + Percentagem_Ser_ProdB){
    produto =2;// Produto ser B.
}
else{// Produto ser C.
    produto =3;
}
return produto;
}

//-----
// Função que Defini a Probalidade de um cliente desistir.
int Define_Desistencia(int percentagem_desistencia){

    srand(time(NULL));
int percentagem_des =0;
int Random=(rand()%100)+1;//Cria Numero Random de 1 a 100

// Produto ser A
if(Random<=percentagem_desistencia){
    percentagem_des=1;
}
return percentagem_des;
}

//-----
// Função "Random" que irá atribuir um empregado para o Cliente.
int Define_Empregado_Atendimento(int num_empregados){
    srand(time(NULL));
int random1 =(rand()%num_empregados)+1;
return random1;
}

//-----
//Funcao Que Escreve Mensagem Para Enviar Pela Socket
void Escreve_Mensagem(int newsockfd,int ID_Cliente,int Estado,int Faixa_Etaria,int
produto,int stock_prod,int empregados,int cliente_troca){

char Buffer[MAX_LINE];
int Tamanho_Mensagem;

sprintf(Buffer,"%d %d %d %d %d %d %d ", ID_Cliente, Estado, Faixa_Etaria, produto,
stock_prod, empregados, cliente_troca);
    Tamanho_Mensagem = strlen(Buffer)+1;// Add one character more because of \n

if(write(newsockfd,Buffer,Tamanho_Mensagem)!= Tamanho_Mensagem)
perror("Erro ao escrever no cliente.\n");
}

//-----
//Funcao Para Ler do Ficheiro Config:
void Ler_Ficheiro_Config(){
//Declaracao e Inicializacao de Variaveis Auxiliares da Funcao
char Le_Linha[100];
int Valor_Variavel_Lida;
char Nome_Variavel_Lida[100];

char Var1[]="Cap_Maxima_Loja";//"Tempo_Medio_Chegadas";
char Var2[]="Cap_Maxima_Fila";// Capacidade Máxima que a Fila da Loja Pode Tomar.
char Var3[]="Tempo_Sim";// Tempo de Simulação.
char Var4[]="tempo_serv_prod";// Tempo Serviço por Produto.
char Var5[]="Prob_Cliente_Idoso";// Percentagem do Cliente ser Idoso.
char Var6[]="Total_Clientes_Gerar";// Total Clientes a Gerar.
char Var7[]="Tempo_Maximo_Espera_Filas";// Tempo Máximo de Espera na Fila.
char Var8[]="Tempo_Medio_Na_Loja";// Tempo Médio dos Clientes Permanecer na Loja.
char Var9[]="Tempo_Medio_Chegadas";// Tempo Médio de Chegada dos Clientes.
char Var10[]="Prob_Prod_A";// Percentagem do Cliente ser ProdutoA.
```

```

char Var11[]="Prob_Prod_B";// Percentagem do Cliente ser ProdutoB.
char Var12[]="Prob_Prod_C";// Percentagem do Cliente ser ProdutoC.

int Erro=0;//Se Uma Das Restricoes For Violada, a Variavel Passa Para 1

FILE*fp;
fp=fopen("config.conf","r");//Vai Abrir o Ficheiro Config.conf

if(fp!=NULL){
//Ciclo de Leitura:
while(fgets(Le_Linha,100,fp)!=NULL){

sscanf(Le_Linha,"%s = %i", Nome_Variavel_Lida,&Valor_Variavel_Lida);

if(strcmp(Nome_Variavel_Lida,Var1)==0){
    Capacidade_Maxima_Loja = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var2)==0){
    Capacidade_Maxima_Fila_Loja = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var3)==0){
    tempo_simulacao = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var4)==0){
    tempo_serv_prod = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var5)==0){
    Percentagem_Ser_Cliente_Idoso = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var6)==0){
    Numero_Clientes_A_Criar = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var7)==0){
    Tempo_Maximo_Espera_Filas = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var8)==0){
    Tempo_Medio_Na_Loja = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var9)==0){
    Tempo_Medio_Chegadas = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var10)==0){
    Percentagem_Ser_ProdA = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var11)==0){
    Percentagem_Ser_ProdB = Valor_Variavel_Lida;
}

elseif(strcmp(Nome_Variavel_Lida,Var12)==0){
    Percentagem_Ser_ProdC = Valor_Variavel_Lida;
}
}

fclose(fp);

else{
printf("Falha ao Abrir Ficheiro Configuracao.");
}

// ***** Erros (Restrições) *****

if(Tempo_Medio_Chegadas <=0){
printf("Erro: Tempo Medio de Chegadas Tem de Ser maior que 0. \n");
Erro =1;
};

if(tempo_simulacao <2){
printf("Erro: Tempo de Simulacao Tem de Ser maior que 2. \n");
Erro =1;
};

```

```

};

if(Capacidade_Maxima_Loja <=0){
printf("Erro: Capacidade Maxima da Loja Tem de Ser maior que 0. \n");
    Erro =1;
};

if(Tempo_Maximo_Espera_Filas <0){
printf("Erro: Tempo Maximo de Espera nas Filas Tem de Ser maior ou igual a 0. \n");
    Erro =1;
};

if(Percentagem_Ser_Cliente_Idoso <0|| Percentagem_Ser_Cliente_Idoso >100){
printf("Erro: Percentagem de Ser Cliente Idoso Tem de Ser >0%% e <=100%%. \n");
    Erro =1;
};

if(Numero_Clientes_A_Criar <=0|| Numero_Clientes_A_Criar >9999){
printf("Erro: Numero de Clientes a Criar Tem de Ser maior que 0 e menor que 10000. \n");
    Erro =1;
};

if(Tempo_Medio_Na_Loja <=0){
printf("Erro: Tempo Medio na Loja Tem de Ser maior que 0. \n");
    Erro =1;
};
if(Capacidade_Maxima_Fila_Loja <=0){
printf("Erro: Capacidade Maxima da Fila na Loja Tem de Ser maior que 0. \n");
    Erro =1;
};
//Se Alguma das Restricoes Anteriores Acontecer, Programa Termina
if(Erro ==1){
exit(0);
};
}

//-----
//Estrutura de Dados dos Clientes
typedef struct{
int Estado_Seguinte;
int ID_Cliente;
int Faixa_Etaria;
int Tempo_Espera_Maximo_Filas;
int Tempo_Estadia_Loja;
int Lugar_Fila_Loja;
int Tempo_Fica_Zona;
int produto;
int Momento_Entrada_Fila_Loja;
int Momento_Vai_Desistir_Fila_Loja;
int empregado;
}

//-----
Struct_Clientes;

int Clientes_Criados=0;
int Tempo_Simulacao_Actual =0;//Guarda o Instante Actual da Simulacao. Comeca a 0. Vai Incrementando Ate Acabar a Simulacao

Struct_Clientes Cliente[10000];//[Numero_Clientes_A_Criar]; ->Numero de Structs de Clientes Criadas

//Declaracao Trincos:
pthread_mutex_t Trinco_Comunicacao_Socket;//Trinco Que Impede Atribuir Instrucoes a Varios Clientes ao Mesmo Tempo

//Declaracao Threads:
pthread_t Thread_Ciclo_Tempo_Simulacao;
pthread_t Thread_Cliente[10000];//[Numero_Clientes_A_Criar]; ->Numero de Tarefas Criadas

//      Declaracao Semaforos:f
sem_t Lotacao_Loja;
sem_t Fila_Loja;

int Clientes_AD_Fila_Loja=0;
int Clientes_ID_Fila_Loja=0;

```

```

int num_empregados =1;

//-----
void Cria_Empregados(){

int Total_Clientes = Clientes_ID_Fila_Loja + Clientes_AD_Fila_Loja;
int Divisao_Clientes;

    Divisao_Clientes = Total_Clientes /2;
    num_empregados = Divisao_Clientes +1;
//  if(Total_Clientes == 0){
//      num_empregados = 1;
//}
}
// Prioridades dos Clientes (Idoso - Adulto)
void Aumenta Tipo Cliente Fila Loja(int Faixa Etaria){

if(Faixa_Etaria ==1){// Idoso.
    Clientes_ID_Fila_Loja++;
}
elseif(Faixa_Etaria ==2){// Adultos.
    Clientes AD Fila Loja++;
}
}

//-----
//Diminui o Numero de Clientes na Fila Dependendo do Seu Tipo
void Diminui_Tipo_Cliente_Fila_Loja(int Faixa_Etaria){

if(Faixa_Etaria ==1){// Idoso
    Clientes_ID_Fila_Loja--;
}
elseif(Faixa Etaria ==2){// Adulto
    Clientes_AD_Fila_Loja--;
}
}
//-----
//Quando o Cliente Chega à Fila da Loja É Verificado o Tipo do Cliente Que Chegou (Idoso
ou Adulto).
void Atribui_Lugar_Fila_Loja_A_Cliente(int ID_Cliente){

int i;

if(Cliente[ID_Cliente].Faixa_Etaria ==1){//Se o Cliente Que Chegou for Idoso.
//Clientes ID Fila Loja++;
    Cliente[ID_Cliente].Lugar_Fila_Loja = Clientes_ID_Fila_Loja;// Vai para último
lugar da fila dos Idosos.

for(i =0; i < Clientes_Criados; i++){
if(Cliente[i].Faixa_Etaria !=1&& Cliente[i].Lugar_Fila_Loja >0){// Vai para a fila
normal da loja.
    Cliente[i].Lugar_Fila_Loja++;
}
}
}
elseif(Cliente[ID_Cliente].Faixa_Etaria ==2){// Se o Cliente que chegou for Adulto.
//Clientes_AD_Fila_Loja++;
    Cliente[ID_Cliente].Lugar_Fila_Loja = Clientes_ID_Fila_Loja +
Clientes_AD_Fila_Loja;//Vai Para o Ultimo Lugar da Fila Composta Pela Soma de Todas as
Outras "Sub Filas MR, PG e Cri"
}
}
//-----
//Funcao Que Faz o Cliente Recebido (1º da Fila) Sair da Fila e Entrar na Zona. E Todos
Avancam 1 Posicao
void Avanca_Um_Lugar_Fila_Loja(){

int i;

for(i =0; i < Clientes_Criados; i++){
if(Cliente[i].Lugar_Fila_Loja >0){// Se está na fila da loja.
    Cliente[i].Lugar_Fila_Loja--;
}
}
}
//-----

```



```
//Funcao Que Faz o Cliente Recebido Desistir da Fila e Todos Atras Dele Avancam 1
Posicao
void Desiste_Fila_E_Avancam_Um_Lugar_Fila_Loja(int ID_Cliente){

int i;
    Cliente[ID_Cliente].Lugar_Fila_Loja =0;// Cliente Que Desiste Sai da Fila
(Poiscao=0)

for(i =0; i < Clientes_Criados; i++){// Todos os Clientes Atras Dele Avancam 1 Posicao

if(Cliente[i].Lugar_Fila_Loja >0){// Se Esta na Fila da Loja.
    Cliente[i].Lugar_Fila_Loja--;
}
}
}
//-----
void Gestor_Eventos_Clientes(int ID_Cliente_Recebido){

    Cliente[ID_Cliente_Recebido].ID_Cliente = ID_Cliente_Recebido;
    Cliente[ID_Cliente_Recebido].produto=
Define_Tipo_Produto(Percentagem_Ser_ProdA,Percentagem_Ser_ProdB,Percentagem_Ser_ProdC);
    Cliente[ID_Cliente_Recebido].Faixa_Etaria =
Define_Tipo_Cliente_Random(Percentagem_Ser_Cliente_Idoso);
    Cliente[ID_Cliente_Recebido].Tempo_Espera_Maximo_Filas =
Tempo_Maximo_Espera_Filas;//Variavel Lida do Config. //Tirar em Principio da Mensagem
    Cliente[ID_Cliente_Recebido].Tempo_Estadia_Loja =
Calcula_Tempos_Medios_Random(Tempo_Medio_Na_Loja);//Tirar em Principio da Mensagem
    Cliente[ID_Cliente_Recebido].Lugar_Fila_Loja =0;
    Cliente[ID_Cliente_Recebido].Momento_Vai_Desistir_Fila_Loja =0;//novo

//Tempo Que Vai Ficar na Zona:
    Cliente[ID_Cliente_Recebido].Tempo_Fica_Zona =
Calcula_Tempos_Medios_Random(Tempo_Medio_Na_Loja);
    Cliente[ID_Cliente_Recebido].empregado=
Define_Empregado_Atendimento(num_empregados);

int Tempo_Actual;
//Variavel Que Guarda Se Houve Desistencia ou Nao
int Desistencia_Fila_Loja =0;
    Cria_Empregados();

#####
// Caso 5:          CLIENTE ENTRA NA LOJA.
// Semáforo Para Controlar o total de clientes que estão dentro da loja.
    sem_wait(&Lotacao_Loja);
    pthread_mutex_lock(&Trinco_Comunicacao_Socket);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente,5,
Cliente[ID_Cliente_Recebido].Faixa_Etaria, Cliente[ID_Cliente_Recebido].produto,10,
Cliente[ID_Cliente_Recebido].empregado,0);
    pthread_mutex_unlock(&Trinco_Comunicacao_Socket);

#####
//case 1:          CLIENTE ENTRA NA FILA DA LOJA.
    sem_wait(&Fila_Loja);
sleep(1);
    pthread_mutex_lock(&Trinco_Comunicacao_Socket);
//-----
    Cliente[ID_Cliente_Recebido].Momento_Entrada_Fila_Loja = Tempo_Simulacao_Actual;
    Tempo_Actual=Tempo_Simulacao_Actual;
    Tempo_Actual=Cliente[ID_Cliente_Recebido].Momento_Entrada_Fila_Loja;//Nao va a Hora
Mudar Entre Estas 2 Instrucoes e Haver Diferencas.
//-----
//Define o momento de Desistencia:

Cliente[ID_Cliente_Recebido].Momento_Vai_Desistir_Fila_Loja=Cliente[ID_Cliente_Recebido]
.Momento_Entrada_Fila_Loja+Cliente[ID_Cliente_Recebido].Tempo_Espera_Maximo_Filas;

//Aumenta o Numero de Clientes n Fila Dependendo do Seu Tipo
    Aumenta_Tipo_Cliente_Fila_Loja(Cliente[ID_Cliente_Recebido].Faixa_Etaria);
sleep(1);
//pthread_mutex_lock(&Trinco_Comunicacao_Socket);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente,1,
Cliente[ID_Cliente_Recebido].Faixa_Etaria, Cliente[ID_Cliente_Recebido].produto,10,
Cliente[ID_Cliente_Recebido].empregado,0);
//pthread_mutex_unlock(&Trinco_Comunicacao_Socket);
```

```

sleep(1);
Atribui_Lugar_Fila_Loja_A_Cliente(Cliente[ID_Cliente_Recebido].ID_Cliente);
pthread_mutex_unlock(&Trinco_Comunicacao_Socket);

while(Cliente[ID_Cliente_Recebido].Lugar_Fila_Loja !=1 && Desistencia_Fila_Loja
!=1){//Enquanto o Cliente da Tarefa Nao For 1º da Fila, Espera
if(Tempo_Simulacao_Actual >=
Cliente[ID_Cliente_Recebido].Momento_Vai_Desistir_Fila_Loja)
{

#####
#####
//case 3:          CLIENTE DESISTE FILA DA LOJA
    sem_post(&Fila_Loja);
    pthread_mutex_lock(&Trinco_Comunicacao_Socket);
//sem_post(&Fila_Loja);
//Diminui o Numero de Clientes na Fila Dependendo do Seu Tipo
    Diminui_Tipo_Cliente_Fila_Loja(Cliente[ID_Cliente_Recebido].Faixa_Etaria);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente,3,
Cliente[ID_Cliente_Recebido].Faixa_Etaria, Cliente[ID_Cliente_Recebido].produto,10,
Cliente[ID_Cliente_Recebido].empregado,0);
sleep(1);
    Desiste_Fila_E_Avancam_Um_Lugar_Fila_Loja(Cliente[ID_Cliente_Recebido].ID_Cliente);
    pthread_mutex_unlock(&Trinco_Comunicacao_Socket);

//Se Cliente Desistir, Activa Sinal Para Indicar Que Nao Vai Haver Entrada na Zona em
Questao nem Saida
    Desistencia_Fila_Loja =1;
}
} // Fim do ciclo While.
if(Desistencia_Fila_Loja ==0){

#####
#####
//case 2:          CLIENTE É ATENDIDO
    sem_post(&Fila_Loja);
    pthread_mutex_lock(&Trinco_Comunicacao_Socket);
//sem_post(&Fila_Loja);
//Diminui o Numero de Clientes na Fila Dependendo do Seu Tipo
    Diminui_Tipo_Cliente_Fila_Loja(Cliente[ID_Cliente_Recebido].Faixa_Etaria);
//sem_post(&Lotacao_Loja);
//*****
*****
if(Cliente[ID_Cliente_Recebido].produto==1){// Produto A.
    quantidade_proda--;
if(quantidade_proda <3){
    Escreve_Mensagem(newsockfd,Cliente[ID_Cliente_Recebido].ID_Cliente,8,
Cliente[ID_Cliente_Recebido].Faixa_Etaria,1,10,
Cliente[ID_Cliente_Recebido].empregado,0);
sleep(1);
        quantidade_proda =10;
}
//sleep(2);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente,2,
Cliente[ID_Cliente_Recebido].Faixa_Etaria, Cliente[ID_Cliente_Recebido].produto,10,
Cliente[ID_Cliente_Recebido].empregado,0);
}
//*****
*****
elseif(Cliente[ID_Cliente_Recebido].produto==2){// Produto B.
    quantidade_prodB--;
if(quantidade_prodB <3){
Escreve_Mensagem(newsockfd,Cliente[ID_Cliente_Recebido].ID_Cliente,9,
Cliente[ID_Cliente_Recebido].Faixa_Etaria,1,10,
Cliente[ID_Cliente_Recebido].empregado,0);
sleep(1);
        quantidade_prodB =10;
}
//sleep(2);
    Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente,6,
Cliente[ID_Cliente_Recebido].Faixa_Etaria,
Cliente[ID_Cliente_Recebido].produto,10,Cliente[ID_Cliente_Recebido].empregado,0);
}
//*****
*****
elseif(Cliente[ID_Cliente_Recebido].produto==3){// Produto C.

```

```

        quantidade_prodC--;
    if(quantidade_prodC <3){

        Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente, 10,
        Cliente[ID_Cliente_Recebido].Faixa_Etaria, 1, 10,
        Cliente[ID_Cliente_Recebido].empregado, 0);
        sleep(1);

        quantidade_prodC = 10;

    }
    //sleep(2);

        Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente, 7,
        Cliente[ID_Cliente_Recebido].Faixa_Etaria,
        Cliente[ID_Cliente_Recebido].produto, 10, Cliente[ID_Cliente_Recebido].empregado, 0);
    }

    sleep(1);

        Avanca_Um_Lugar_Fila_Loja(); //int ID_Cliente);
        pthread_mutex_unlock(&Trinco_Comunicacao_Socket);
        srand(time(NULL));
    } // Fim do If se Houve Desistencia.
    //#####
    //#####
    //case 4:          CLIENTE SAI DA LOJA
        pthread_mutex_lock(&Trinco_Comunicacao_Socket);
        Escreve_Mensagem(newsockfd, Cliente[ID_Cliente_Recebido].ID_Cliente, 4,
        Cliente[ID_Cliente_Recebido].Faixa_Etaria, Cliente[ID_Cliente_Recebido].produto, 10,
        Cliente[ID_Cliente_Recebido].empregado, 0);
        sleep(1);
        pthread_mutex_unlock(&Trinco_Comunicacao_Socket);
        sleep(1);
        sem_post(&Lotacao_Loja);
    }
    //-----
    //-----
    //Vai Criar Clientes Ate Alcançar o Tempo de Simulacao OU Criar Todos os Clientes
    Definidos
    void Criacao_Clientes()
    {
        srand(time(NULL));
        long Clientes_Criados2 = 0;

        while(Tempo_Simulacao_Actual < tempo_simulacao && Clientes_Criados <=
        Numero_Clientes_A_Criar){

            if(pthread_create(&(Thread_Cliente[Clientes_Criados]), NULL, (void*)&Gestor_Eventos_Clientes, (void*)Clientes_Criados2) != 0){
                printf("Erro na criação da tarefa de criação dos clientes\n");
                exit(1);
            }

            sleep(Calcula_Tempos_Medios_Random(Tempo_Medio_Chegadas));
            Clientes_Criados++; //Variavel Global
            Clientes_Criados2++; //Variavel Local -> Senao, Para Eliminar Esta Variavel,
            Talvez Por a Variavel Global Como Long!
        }
    }

    //-----
    //-----
    //Relógio Funciona Ate Todos os Clientes Sairem
    void Funcao_Incrementa_Tempo_Actual(){

        while(Fecha_Loja != 1){
            Tempo_Simulacao_Actual = Tempo_Simulacao_Actual + 1;
            sleep(1);
        }
    }
    //-----
    //-----
    //Quando Recebe Mensagem do Monitor Com Sinal Para Começar a Simulacao
    void Comeca_Simulacao(){

        if(pthread_create(&(Thread_Ciclo_Tempo_Simulacao), NULL, (void*)&Funcao_Incrementa_Tempo_Actual, NULL) != 0){
            printf("Falha na Criação da tarefa do ciclo tempo.\n");
            exit(1);
        }
    }

```

```

}
//-----
int main(int argc, char* argv[])
{
    // Leitura do ficheiro configuração.
    Ler_Ficheiro_Config();
    pthread_mutex_init(&Trinco_Comunicacao_Socket, NULL);

    //Semaforos Para o Gestor da Loja:
    sem_init(&Lotacao_Loja, 0, Capacidade_Maxima_Loja);
    sem_init(&Fila_Loja, 0, Capacidade_Maxima_Fila_Loja);

    int clilen, servlen, i;
    struct sockaddr_un cli_addr, serv_addr;

    // Cria Socket
    if((sockfd = socket(AF_UNIX, SOCK_STREAM, 0)) < 0)
        perror("Falha ao criar socket stream servidor.\n");

    unlink(UNIXSTR_PATH);
    bzero((char*)&serv_addr, sizeof(serv_addr));
    serv_addr.sun_family = AF_UNIX;
    strcpy(serv_addr.sun_path, UNIXSTR_PATH);
    servlen = strlen(serv_addr.sun_path) + sizeof(serv_addr.sun_family);

    if(bind(sockfd, (struct sockaddr *)&serv_addr, servlen) < 0)
        perror("Falha ao atribuir nome socket servidor.\n");

    listen(sockfd, 1);
    clilen = sizeof(cli_addr);
    newsockfd = accept(sockfd, (struct sockaddr *)&cli_addr, &clilen);

    if(newsockfd < 0)
        perror("Falha ao criar ligacao dedicada.\n");

    printf("Pressione SPACE Para Começar Simulação. \n");

    char Tecla=0;

    while(Tecla != ' '){
        Tecla=getchar();
    }

    //Começa Simulação
    Começa_Simulação();
    Criação_Clientes();
    sleep(2);

    int qwe;

    for(qwe = 0; qwe < Clientes_Criados; qwe++){
        pthread_join(Thread_Cliente[qwe], NULL);
    }

    printf("Acabou Join Threads dos Clientes\n");
    sleep(2);
    //Quando Termina Todas as Threads Criadas, a Loja Começa a Fechar
    Fecha_Loja = 1;
    pthread_join(Thread_Ciclo_Tempo_Simulação, NULL);
    printf("Acabou Join Thread do Ciclo do Tempo de Simulação.\n");
    sleep(2);

    //Mensagem de Fim de Simulação
    pthread_mutex_lock(&Trinco_Comunicacao_Socket);
    Escreve_Mensagem(newsockfd, 0, 50, 0, 0, 0, 0);
    printf("Enviou Mensagem Fim Simulação!\n");
    pthread_mutex_unlock(&Trinco_Comunicacao_Socket);
    sleep(2);

    //Fechar o Socket
    close(newsockfd);
    return 0;
}

```

Estatísticas.c

Estatísticas

Clientes Atendidos na Loja

- * Total Clientes Atendidos:67
- * Total Clientes Idosos Atendidos:34
- * Total Clientes Adultos Atendidos:33

Clientes Entraram na Fila da Loja

- * Total Clientes Entraram Fila:74
- * Total Clientes Idosos Entraram Fila:35
- * Total Clientes Adultos Entraram Fila:39

Clientes Desistiram da Fila da Loja

- * Total Clientes Desistiram da Fila:6
- * Total Clientes Idosos Desistiram da Fila:1
- * Total Clientes Adultos Desistiram da Fila:5

Vendas de Produtos

- * Total de Vendas Produto A:27
- * Total de Vendas Produto B:18
- * Total de Vendas Produto C:22

Dados Impressos Terminados.

Config.conf

```
Cap_Maxima_Loja = 15
Cap_Maxima_Fila = 4
Tempo_Sim = 75
tempo_serv_prod = 80
Prob_Cliente_Idoso = 50
Total_Clientes_Gerar = 100
Tempo_Maximo_Espera_Filas = 20
Tempo_Medio_Na_Loja = 10
Tempo_Medio_Chegadas = 1
Prob_Prod_A = 30
Prob_Prod_B = 35
Prob_Prod_C = 30
```