

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](#_Toc534814578)

[Funcionamento da loja 4](#_Toc534814579)

[Utilização dos mecanismos de sincronização 7](#_Toc534814580)

[Políticas de acesso 8](#_Toc534814581)

[Alguns Testes 9](#_Toc534814582)

[Conclusão 12](#_Toc534814583)

[Anexos 13](#_Toc534814584)

[Fluxogramas 13](#_Toc534814585)

[Fluxograma dos Produtos 13](#_Toc534814586)

[Fluxograma de Empregados 14](#_Toc534814587)

[Fluxograma Clientes 15](#_Toc534814588)

[Código 16](#_Toc534814589)

[Monitor.c 16](#_Toc534814590)

[Simulador.c 25](#_Toc534814591)

[Estatísticas.c 34](#_Toc534814592)

[Config.conf 34](#_Toc534814593)

# **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.

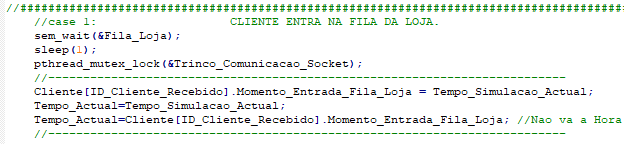
Para esta terceira fase pretende-se que o grupo implemente as bibliotecas para comunicação entre o Simulador e o Monitor, a interface onde neste deverá mostrar informação pertinente ao utilizador do estado corrente da sua simulação, bem como a implementação de semáforos e trincos em zonas pertinentes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

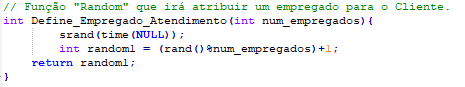
**Clientes**

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 saí 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.

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

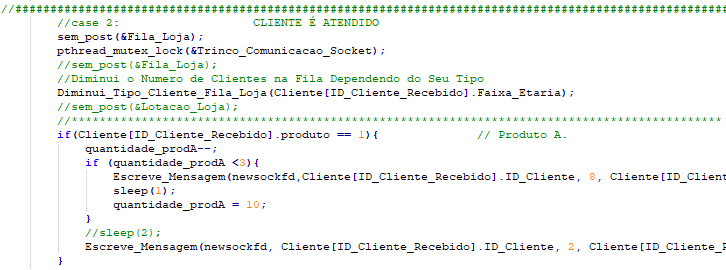
No caso de haver mais do que 1 empregado a servir os clientes, foi também implementado no código uma função *random* onde o cliente opta desde o empregado 1 até ao N empregado gerado (*Fig.2*).



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

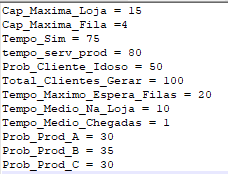
**Empregados**

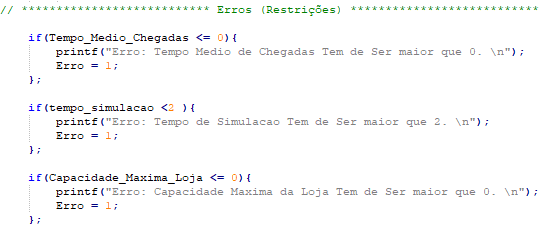
A loja inicialmente tem 1 empregado, cada vez que está 5 clientes na fila de atendimento é alocado um novo empregado. Quando o produto estiver abaixo de uma certa quantidade o empregado a repor será o primeiro que acabar o atendimento ao cliente, tendo os empregados a prioridade de verificar o stock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



**Fig.3:** Verificação do stock do produto aquando cliente é atendido.

**Ficheiro Configuração**

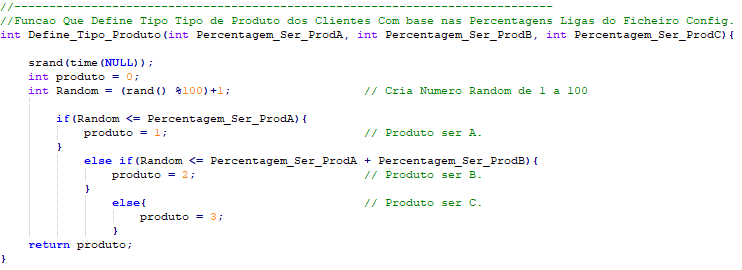
Foi também implementado o ficheiro de configuração onde é carregado para o simulador de maneira a poder testar diferentes ambientes de simulação. Foi também implementando como extra a verificação dos valores inseridos no ficheiro de configuração de maneira a respeitar os standards exigidos.



**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 menordo 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.



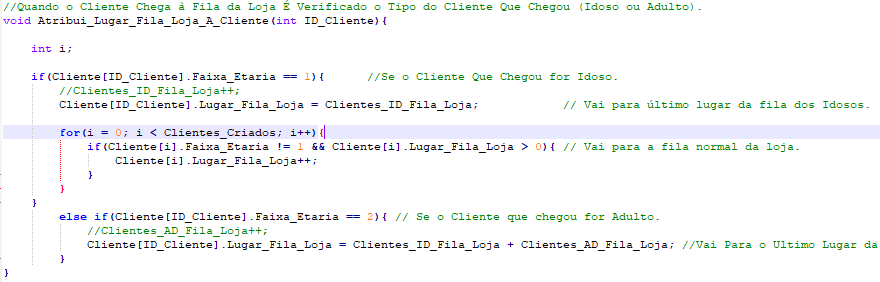
**Fig.5:** Função *random* que atribui que produto cliente irá optar com base nas percentagens obtidas do ficheiro de configuração.

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 reposto.

# **Utilização dos mecanismos de sincronização**

O projeto contem 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*).

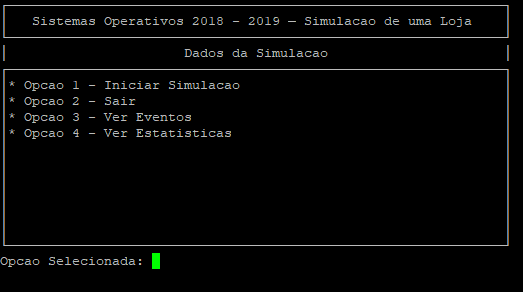


**Fig.6:** Atribuição de lugar na fila conforme a faixa etária do cliente.

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

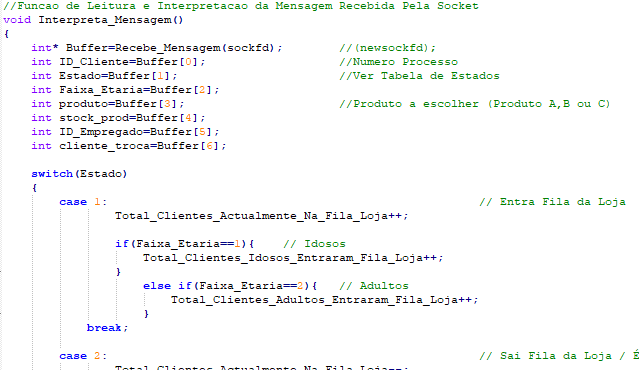
# **Políticas de acesso**

A comunicação entre o Simulador e Monitor é efetua 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.

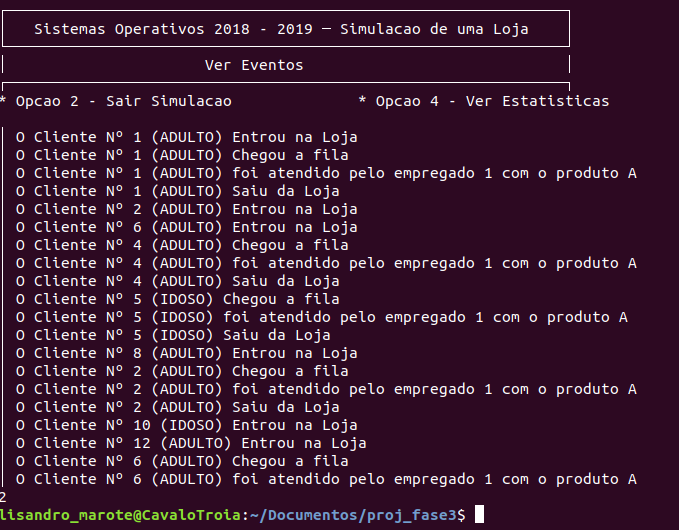
Também foi implementado uma função que interpreta os dados obtidos a partir do simulador a fim de fazer print no monitor com base nos valores recebidos.

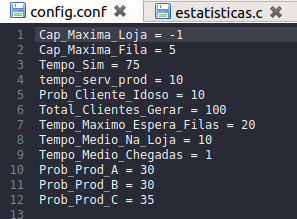


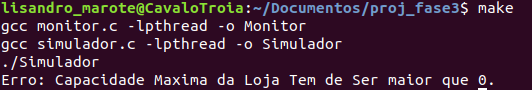
**Fig.8:** Função que interpreta mensagem enviada pela socket simulador – monitor.

# **Alguns Testes**

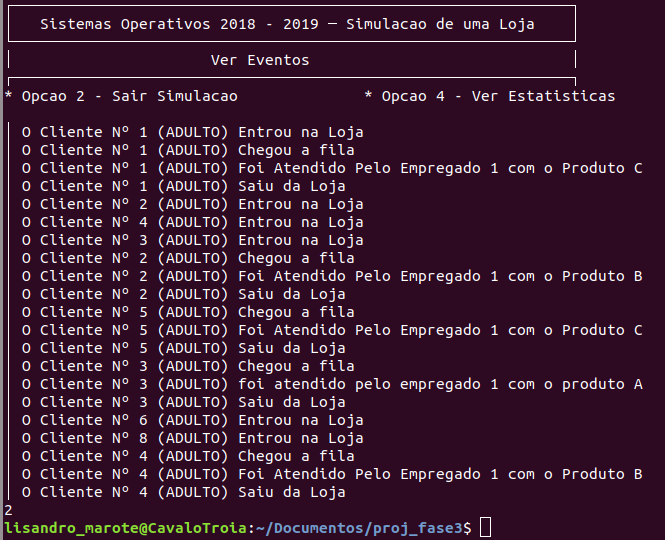
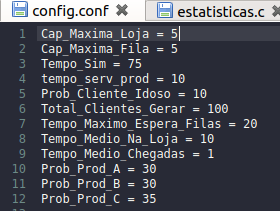
Aqui iremos demostrar pequenos exemplos com print´s screens das simulações efetuadas, com valores diferentes no ficheiro de configuração.

**Simulação 1:** Probabilidade Produto A – 100%.

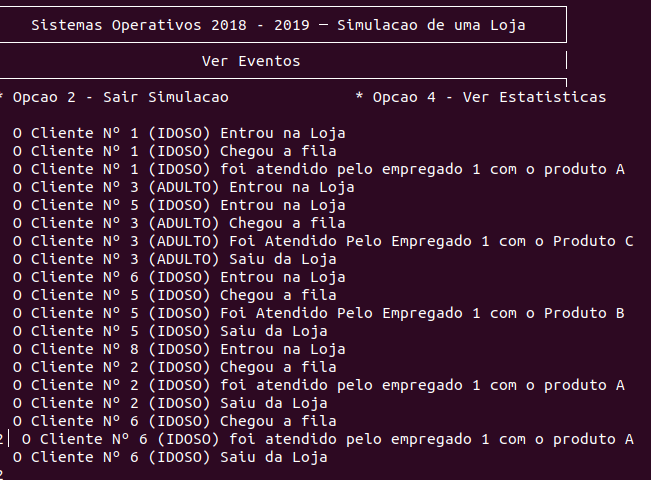
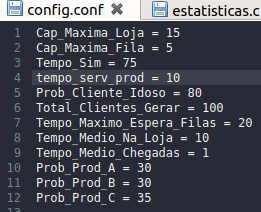
**Simulação 2:** Inserção valores incorretos no ficheiro de configuração.



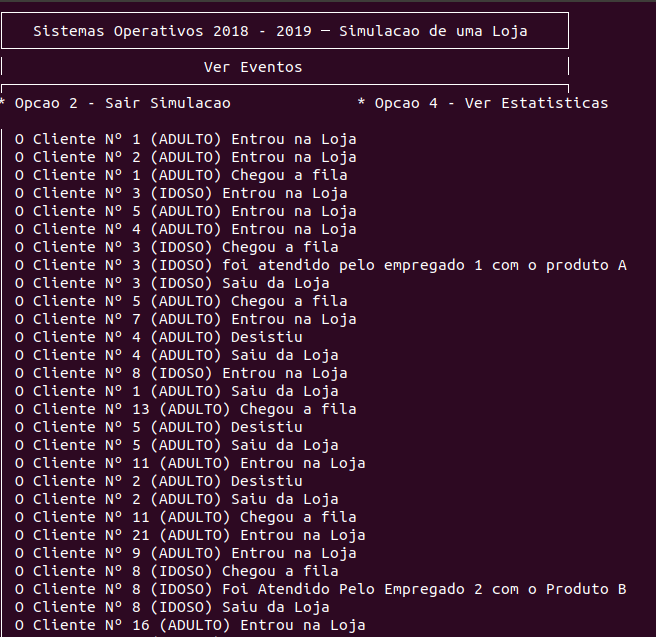
**Simulação 3:** Lotação máxima da loja de 5 Clientes.

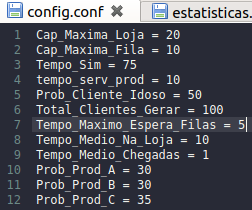


**Simulação 4:** Probabilidade de ser idoso ser 80%.



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





# **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 reposto 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.

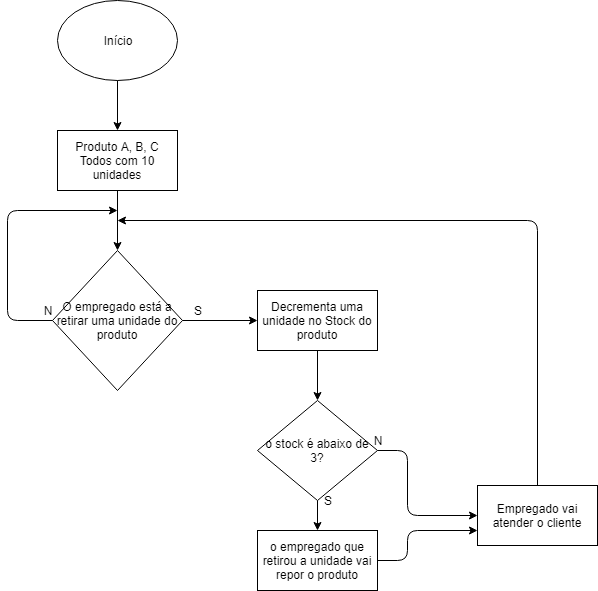
Uma das maiores dificuldades sentidas nesta terceira fase foi o tempo necessário de maneira a obter tudo o era pedido segundo o enunciado, todas as funcionalidades do projeto a funcionar, fluxogramas, relatórios etc devido à carga excessiva de trabalhos, relatórios apresentações que foi imposta neste 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

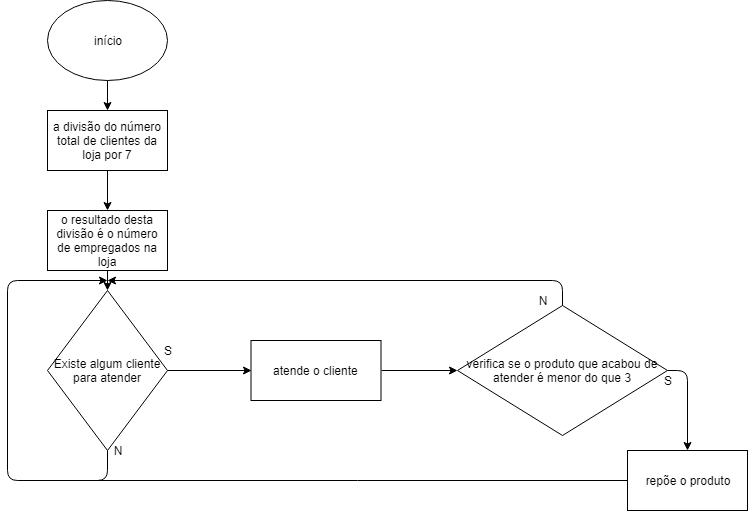
# **Anexos**

## Fluxogramas

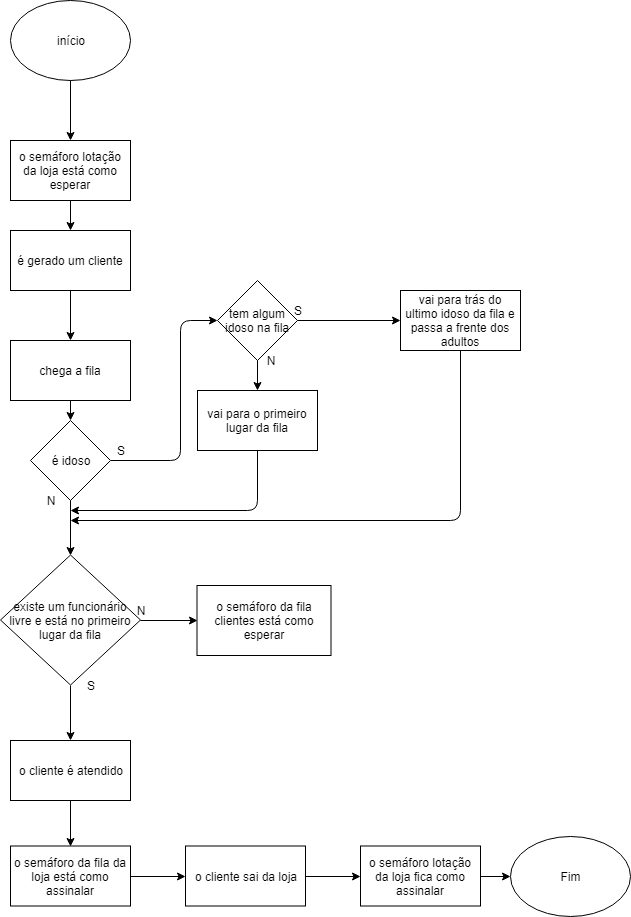
### Fluxograma dos Produtos



### 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 execução da Simulacao:

bool Simulacao\_Em\_Curso **=**false**;**// Indica se a simulação 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\_Confg**()**

**{**

//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\_prodA**;**

int quantidade\_prodB**;**

int quantidade\_prodC**;**

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\_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**;**

**}**

FILE**\***fp**;**

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

**switch(**Evento**)**

**{**

**case**1**:**

fprintf**(**fp**,**"->Cliente Nº %i (%s) Entrou na Loja \n"**,** ID\_Cliente**+**1**,**Nome\_Faixa\_Etaria**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**2**:**

fprintf**(**fp**,**"->Cliente Nº %i Chegou a fila \n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**3**:**

fprintf**(**fp**,**"->Cliente Nº %i Foi atendido com o produto A\n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**4**:**

fprintf**(**fp**,**"->Cliente Nº %i Desistiu da Fila \n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**5**:**

fprintf**(**fp**,**"->Cliente Nº %i Saiu da Loja \n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**6**:**

fprintf**(**fp**,**"->Cliente Nº %i Foi atendido com o produto B\n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**7**:**

fprintf**(**fp**,**"->Cliente Nº %i Foi atendido com o produto C\n"**,** ID\_Cliente**+**1**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**8**:**

fprintf**(**fp**,**"O Empregado Nº1 foi repor o produto A"**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**9**:**

printf**(**"│ O Empregado Nº1 foi repor o produto B\n"**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**10**:**

printf**(**"│ O Empregado Nº1 foi repor o produto C\n"**);**

fprintf**(**fp**,**"-----------------------------------------------------\n"**);**

**break;**

**case**50**:**

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\_empregado**)**

**{**

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**)**

**{**

**case**1**:**

printf**(**"│ O Cliente Nº %d %s Chegou a fila \n"**,** ID\_Cliente**+**1**,**Nome\_Faixa\_Etaria**);**

**break;**

**case**2**:**

printf**(**"│ O Cliente Nº %d %s foi atendido pelo empregado %d com o produto A\n"**,** ID\_Cliente**+**1**,** Nome\_Faixa\_Etaria**,** ID\_empregado**);**

**break;**

**case**3**:**

printf**(**"│ O Cliente Nº %d %s Desistiu\n"**,** ID\_Cliente**+**1**,** Nome\_Faixa\_Etaria**);**

**break;**

**case**4**:**

printf**(**"│ O Cliente Nº %d %s Saiu da Loja \n"**,** ID\_Cliente**+**1**,** Nome\_Faixa\_Etaria**);**

**break;**

**case**5**:**

printf**(**"│ O Cliente Nº %d %s Entrou na Loja \n"**,** ID\_Cliente**+**1**,**Nome\_Faixa\_Etaria**);**

**break;**

**case**6**:**

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;**

**case**7**:**

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;**

**case**8**:**

printf**(**"│ O Empregado Nº %d Foi Repor o Produto A\n"**,** ID\_empregado**);**

**break;**

**case**9**:**

printf**(**"│ O Empregado Nº %d Foi Repor o Produto B\n"**,** ID\_empregado**);**

**break;**

**case**10**:**

printf**(**"│ O Empregado Nº %d Foi Repor o Produto C\n"**,** ID\_empregado**);**

**break;**

**case**50**:**

printf**(**"│ Simulacao Terminada. \n"**);**

**break;**

**default:**

printf**(**"│ Erro: Caso Nao Definido. \n"**);**

**}**

**}**

//------------------------------------------------------------------------------------------------------------------

//Funcao Para Mostrar as Estatisticas da simulação 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**(**"┌──────────────────────────────────────────────────────────────┐\n"**);**

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

printf**(**"└──────────────────────────────────────────────────────────────┘\n"**);**

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**(**"┌──────────────────────────────────────────────────────────────┐\n"**);**

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**,**"\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**,**"┌──────────────────────────────────────────────────────────────┐\n"**);**

fprintf**(**fp**,**"│ Clientes Entraram na Fila da Loja │ \n"**);**

fprintf**(**fp**,**"└──────────────────────────────────────────────────────────────┘\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**,**"┌──────────────────────────────────────────────────────────────┐\n"**);**

fprintf**(**fp**,**"│ Clientes Desistiram da Fila da Loja │ \n"**);**

fprintf**(**fp**,**"└──────────────────────────────────────────────────────────────┘\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**,**"┌──────────────────────────────────────────────────────────────┐\n"**);**

fprintf**(**fp**,**"│ Vendas de Produtos │ \n"**);**

fprintf**(**fp**,**"└──────────────────────────────────────────────────────────────┘\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**];**

staticintMensagem**[**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\_prod**,&**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;**

**case**3**:**// 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;**

**case**4**:**// Sai Loja

Total\_Clientes\_Actualmente\_Na\_Loja**--;**

Total\_Clientes\_Sairam\_Loja**++;**

**break;**

**case**5**:**// Entra Loja

Total\_Clientes\_Actualmente\_Na\_Loja**++;**

**break;**

**case**6**:**// 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;**

**case**7**:**// 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;**

**case**8**:**// Sai Fila da Loja / É atendido com o produto C

quantidade\_prodA **=**10**;**

**break;**

**case**9**:**// Sai Fila da Loja / É atendido com o produto C

quantidade\_prodB **=**10**;**

**break;**

**case**10**:**// Sai Fila da Loja / É atendido com o produto C

quantidade\_prodC **=**10**;**

**break;**

**case**50**:**// Fim Simulação

Simulacao\_Em\_Curso **=** false**;**

// Escreve dados das estatisticas 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 estatisticas 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 Selecionada: "**);**

**}**

//verifica a escolha do menus

scanf**(**"%d"**,&**Opcao\_Utilizador**);**

**switch(**Opcao\_Utilizador**){**

**case**1**:**//Se Nao Houver Nenhuma Simulacao Activa, começa uma nova.

**if(**Simulacao\_Em\_Curso **==** false**){**

Prepara\_Ficheiro\_Eventos**();**

system**(**"clear"**);**

printf**(**"┌──────────────────────────────────────────────────────────────┐\n"**);**

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;**

**case**2**:**//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;**

**case**3**:**//Ver Eventos em tempo Real

**if(**Simulacao\_Em\_Curso **==** true**){**

system**(**"clear"**);**

printf**(**"┌──────────────────────────────────────────────────────────────┐\n"**);**

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

printf**(**"└──────────────────────────────────────────────────────────────┘\n"**);**

printf**(**"│ Ver Eventos │\n"**);**

printf**(**"┌──────────────────────────────────────────────────────────────┐\n"**);**

printf**(**"\* Opcao 2 - Sair Simulacao \* Opcao 4 - Ver Estatisticas \n"**);**

printf**(**" \n"**);**

**}**

**else{**

printf**(**"Erro: Nenhuma Simulacao Activa Neste Momento.\n"**);**

**}**

**break;**

**case**4**:**//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**();**

**}**

**return**0**;**

**}**

### 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 Seriço 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 Permancer 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 Alcancar 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 criacao da tarefa de criacao 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!

**}**

**}**

//------------------------------------------------------------------------------------------------------

//Relogio 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 Comecar a Simulacao

void Comeca\_Simulacao**(){**

**if(**pthread\_create**(&(**Thread\_Ciclo\_Tempo\_Simulacao**),NULL,(**void**\*)&**Funcao\_Incrementa\_Tempo\_Actual**,NULL)!=**0**){**

printf**(**"Falha na Criacao da tarefa do ciclo tempo.\n"**);**

exit**(**1**);**

**}**

**}**

//------------------------------------------------------------------------------------------------------

intmain**(**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 Comecar Simulacao. \n"**);**

char Tecla**=**0**;**

**while(**Tecla**!=**' '**){**

Tecla**=**getchar**()**

**;}**

//Comeca Simulacao

Comeca\_Simulacao**();**

Criacao\_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 Comeca a Fechar

Fecha\_Loja **=**1**;**

pthread\_join**(**Thread\_Ciclo\_Tempo\_Simulacao**,NULL);**

printf**(**"Acabou Join Thread do Ciclo do Tempo de Simulacao.\n"**);**

sleep**(**2**);**

//Mensagem de Fim de Simulacao

pthread\_mutex\_lock**(&**Trinco\_Comunicacao\_Socket**);**

Escreve\_Mensagem**(**newsockfd**,**0**,**50**,**0**,**0**,**0**,**0**,**0**);**

printf**(**"Enviou Mensagem Fim Simulacao!\n"**);**

pthread\_mutex\_unlock**(&**Trinco\_Comunicacao\_Socket**);**

sleep**(**2**);**

//Fechar o Socket

close**(**newsockfd**);**

**return**0**;**

**}**

### Estatísticas.c

│ Estatisticas │

┌──────────────────────────────────────────────────────────────┐

│ 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