

Sistemas Operativos

Projeto prático 2013/2014

(2º Fase)

Simulação de uma Loja

Docentes: Elsa Carvalho, Eduardo Marques

Licenciatura Engenharia Informática

Grupo 22

Cristina Hart, 2017911

Denis Vieira, 2012704

Frederico Jesus, 2063211

Índice

[Introdução 3](#_Toc377832157)

[Desenvolvimento 4](#_Toc377832158)

[Descrição do funcionamento 4](#_Toc377832159)

[Simulador 4](#_Toc377832160)

[Comunicação 4](#_Toc377832161)

[Monitor 5](#_Toc377832162)

[Comunicação 6](#_Toc377832163)

[Mensagens 6](#_Toc377832164)

[Implementação e Sincronização 7](#_Toc377832165)

[Fluxogramas 9](#_Toc377832166)

[Tarefa Cliente 9](#_Toc377832167)

[Tarefa Empregado 10](#_Toc377832169)

[Tarefa Gerente 11](#_Toc377832170)

[Conclusão 12](#_Toc377832171)

[Anexos 13](#_Toc377832172)

# Introdução

Neste projeto prático é-nos proposto a conceção e implementação da simulação de uma loja no âmbito da cadeira de Sistemas Operativos.

Para esta última fase foi-nos pedido que obtivéssemos todo o projeto funcional atendendo à descrição do projeto fornecido no enunciado.

Vamos descrever de forma simples e concreta o protocolo de comunicação entre o cliente e o servidor. A comunicação entre o monitor e o simulador é realizada através de sockets como aprendemos nas aulas práticas.

Além disto, teremos ainda uma descrição de como funciona toda a loja e como foi implementado todo o projeto.

Em anexo vamos entregar também um relatório de um teste de 5 minutos.

# Desenvolvimento

## Descrição do funcionamento

Quando a loja abre aos clientes, estes vão chegando de X em X tempo, podendo estes ser prioritários ou não.

Existem duas filas para o balcão de atendimento: uma prioritária e outra normal. Quando o cliente chega, existe uma probabilidade de ele desistir e ir-se embora por constatar que a fila está muito grande, senão, coloca-se numa das filas até ser chamado pelo empregado. Quando chega à sua vez, o empregado espera que o cliente escolha o seu produto. Depois de tomada a decisão, o cliente informa o empregado e aguarda enquanto este encarrega-se de traze-lo ao cliente. Por sua vez, o cliente poderá mudar a sua escolha e pedir outro tipo de produto, sendo atendido pelo mesmo empregado que espera sempre pela tomada de decisão.

Depois do cliente sair, se o stock do produto que o cliente escolheu for menor que X, o empregado que o atendeu é responsável por fazer a sua reposição.

A loja tem, também, um gerente que encarrega-se colocar ou retirar empregados extra ao balcão, dependendo do número de clientes na fila. De modo a evitar a constante colocação e retirada de empregados, o gerente não está constantemente a verificar o tamanho da fila.

## Simulador

No início, o simulador lê o ficheiro de configuração que contém:

|  |  |
| --- | --- |
| Ficheiro | Descrição |
| TEMPO\_MEDIO\_CHEGADA\_CLIENTES=5 | Tempo médio de chegada de clientes |
| NUMERO\_PRODUTOS=6 | Número total de tipo de produtos |
| STOCK\_INICIAL\_PRODUTOS=7 | Stock inicial de cada produto |
| TEMPO\_PRODUTO\_UM=2 | Tempo 1 de atendimento de produto |
| TEMPO\_PRODUTO\_DOIS=4 | Tempo 2 de atendimento de produto |
| MAX\_CLIENTES\_FILA=5 | Nº máximo de clientes na fila |
| MIN\_STOCK\_PRODUTO=2 | Mínimo de stock de cada produto |
| PROB\_CLIENTE\_DESISTIR =25 | Probabilidade de um cliente desistir |
| TAMANHO\_FILA\_DESISTIR=4 | Tamanho mínimo da fila para desistir |
| PROB\_CLIENTE\_ALTERAR\_PRODUTO=5 | Probabilidade de um cliente trocar de produto |
| PROB\_CLIENTE\_SER\_PRIO=10 | Probabilidade de um cliente ser prioritário |
| NUMERO\_MAX\_EMPREGADOS=5 | Número máximo de empregados ao balcão |
| TEMPO\_REPOSIÇAO=2 | Tempo que o empregado demora a repor |
| RONDA\_GERENTE=15 | Intervalo de tempo das rondas do gerente |

***Tabela 1*** *- Ficheiro de configuração*

### Comunicação

A comunicação simulador 🡪 monitor é feita através de um código. Esse código é dividido em sete elementos sendo que o primeiro e o segundo têm sempre a mesma representação. O primeiro é o tempo e o segundo é o evento. Os cinco últimos variam dependendo do evento.

Para uma melhor explicação, observaremos as duas tabelas seguintes:

|  |  |
| --- | --- |
| Evento | Descrição |
| 1 | Abrir loja (1), fechar loja (0) e fechar por terminar tempo de simulação (2) |
| 2 | Entrada de cliente na loja |
| 3 | Cliente escolhe produto |
| 4 | Cliente é atendido e sai da loja |
| 5 | Cliente desiste por aborrecimento |
| 6 | Cliente trocou de produto |
| 7 | Novo empregado ao balcão |
| 8 | Reposição de produto |
| 9 | Retirado empregado do balcão |

***Tabela 2*** *- Descrição dos eventos*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Evento | Código | | | | |
| 1 | **Cod1** | **Cod2** | **Cod3** | **Cod4** | **Cod5** |
| 1 | Nº de produtos | Quantidade de produtos | - | - |
| 0 | - | - | - | - |
| 2 | - | - | - | - |
| 2 | Nº do Cliente | Tipo de cliente | - | - | - |
| 3 | Nº do Cliente | Tipo de Produto | Tipo de cliente | Tempo de espera | Nº do Empregado |
| 4 | Nº do Cliente | Tipo de cliente | - | - | - |
| 5 | Nº do Cliente | Tipo de cliente | - | - | - |
| 6 | Nº do Cliente | Tipo de Produto | Tipo de Produto | - | - |
| 7 | - | - | - | - | - |
| 8 | Tipo de Produto | Quantidade | Nº do Empregado | - | - |
| 9 | - | - | - | - | - |

***Tabela 3*** *- Codificação*

## Monitor

Como o nome indica, o monitor servirá de interface para o utilizador. Será apresentado, no início, uma série de opções que o utilizador poderá usar antes, durante e depois da simulação. Sendo estas:

|  |  |
| --- | --- |
| Opção | Descrição |
| 1 | Começar simulação |
| 2 | Mostrar modo estatística |
| 3 | Retomar modo simulação |
| 4 | Sair da simulação |

***Tabela 4*** *- Opções monitor*

Sendo que a estatística será mostrada da seguinte maneira:

|  |  |
| --- | --- |
| Estatística | Descrição |
| 1 | Estado da simulação |
| 2 | Nº de clientes na fila |
| 3 | Nº de clientes na fila prioritária |
| 4 | Nº de empregados no balcão |
| 5 | Quantidade de produto A |
| 6 | Quantidade de produto B |
| ... | Quantidade de produto ... |
| 4 + N | Quantidade de produto N |
| 4 + (N+1) | Nº total de clientes\* |
| 4 + (N+2) | Nº total de clientes prioritários |
| 4 + (N+3) | Nº de desistências |
| 4 + (N+4) | Nº de clientes que trocaram de produto |
| 4 + (N+4) | Tempo médio de espera na fila |
| 4 + (N+5) | Tempo médio de espera na fila prioritária |

***Tabela 5*** *- Demonstração de estatísticas*

\*De notar que o número total de clientes engloba tanto clientes normais como prioritários.

### Comunicação

A comunicação monitor 🡪 simulador é feita, simplesmente, através de um número. Este número representa a opção da *tabela 4* escolhida pelo utilizador e irá ter o efeito escolhido no simulador apenas nas opções começar, mostrar estatística, retomar e sair.

**Exemplo:** O utilizador escolhe 1 para começar, é enviado um socket para o simulador com esta opção. O simulador reconhece a opção e começa a simulação.

### Mensagens

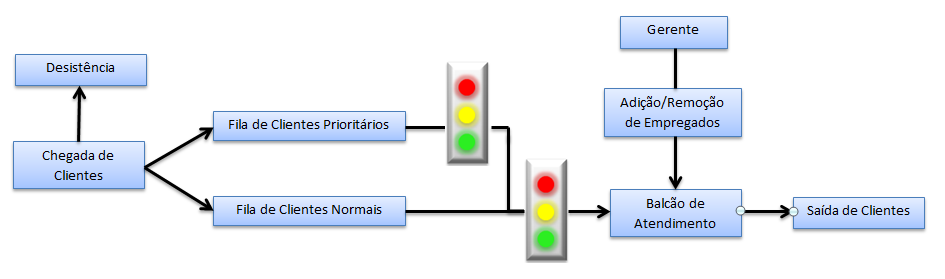
|  |  |
| --- | --- |
| Evento | Mensagem |
| 1 | 1: “TEMPO” - A loja abriu aos clientes.  0: “TEMPO” - A loja fechou.  2: “TEMPO” – A loja vai fechar. Não podem entrar mais clientes. |
| 2 | “TEMPO” - Entrou um cliente na loja. (Cliente nº “cod1”) |
| 3 | “TEMPO” - O cliente “cod1” escolheu o produto “cod2” e vai ser atendido pelo empregado “cod5”. |
| 4 | “TEMPO” - O cliente “cod1” foi atendido e saiu da loja. |
| 5 | “TEMPO” - O cliente “cod1” desistiu por achar que a fila está muito grande. |
| 6 | “TEMPO” - O cliente “cod1” decidiu trocar o produto “cod2” pelo produto “cod3”. |
| 7 | “TEMPO” - O gerente chamou um novo empregado para o balcão. |
| 8 | “TEMPO” – O empregado “cod3” foi fazer reposição do produto “cod1”. |
| 9 | “TEMPO” - O gerente retirou um empregado do balcão. |

Depois de receber os códigos do simulador, o monitor trata-os da seguinte maneira:

***Tabela 6*** *– Mensagens*

Estas mensagens são impressas no monitor e no ficheiro “relatorio.txt”.

## Implementação e Sincronização

 De acordo com a figura 1, os clientes, quando chegam à loja, podem desistir de forma aleatória dependendo do tamanho da fila (se esta estiver muito grande). Também de forma aleatória, o cliente pode ser prioritário, colocando-se no semáforo “fila\_prioritaria”, ou não, colocando-se no semáforo “fila\_normal”. Este fica à espera que chegue à sua vez para ser atendido pelo empregado.

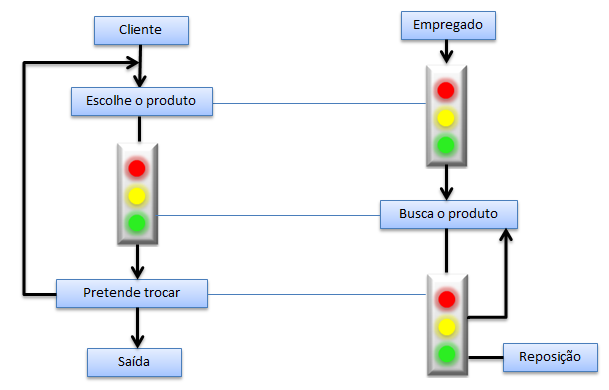
“fila\_prioritaria”

“fila\_normal”

***Figura 1 –*** *Sincronização geral da loja*

Protegido por um trinco, o empregado coloca o seu id numa variável global “empregadoDisponivel” e então assinala o semáforo “fila\_normal” se não existirem clientes prioritários na fila e de seguida assinala o semáforo “fila\_prioritaria”. O cliente passa então para o atendimento, deixando o empregado à espera da sua escolha. Esta parte do código foi protegida de modo a que só um empregado consiga assinalar a fila de cada vez e que o cliente saiba sempre que empregado lhe atendeu através da variável global “empregadoDisponivel”. Entretanto, o cliente preenche alguns campos da estrutura à frente explicada.

Para a interação clientes-empregados, foi criado um array de estruturas de dados com o tamanho do número máximo de empregados. Cada estrutura representa a comunicação entre um determinado cliente e um empregado. Essa estrutura contém a seguinte informação: o id do cliente, o tempo de chegada ao atendimento do cliente, o produto escolhido pelo cliente, uma variável booleana que indica se o cliente pretende trocar de produto e dois semáforos para a sincronização, semáforo “espera\_atendimento” e semáforo “espera”. Sendo que o índice da estrutura representa o id do empregado.



“espera”

“espera\_atendimento”

***Figura 2 –*** *Sincronização do atendimento*

Como podemos observar com a Figura 2, o empregado está à espera que o cliente escolha um produto. Quando este o escolhe, assinala o semáforo “espera” e fica à espera que o empregado vá buscar o produto. Após ter ido busca-lo, assinala o semáforo “espera\_atendimento” e o cliente decide, então, se está satisfeito ou se quer trocar o produto. De notar que o cliente apenas pode trocar de produto uma vez de modo a evitar um loop demasiado grande. O empregado está à espera que o cliente decida a sua satisfação. Se o cliente decidir que está satisfeito, avisa o empregado (assinala o semáforo “espera”) e sai da loja. Caso pretenda trocar, escolhe o novo produto e pede ao empregado para o ir buscar (assinala semáforo “espera”).

Já o empregado, no caso em que o cliente está satisfeito e abandona a loja, faz a reposição do stock do produto escolhido pelo cliente se necessário.

Quanto à adição ou remoção de funcionários, o gerente realiza rondas de verificação do tamanho da fila. Se este achar que a fila está muito grande chama um empregado extra ao balcão. Caso ache também que tem muitos empregados para um certo número de clientes manda um empregado se retirar.

É depois do atendimento de um cliente que o empregado verifica se o gerente o mandou retirar, se não, volta a atender outro cliente que esteja à espera na fila.

Quanto a trincos, decidimos criar 3, um para as variáveis clientes, um para as variáveis empregado e um para o stock, isto porque, assim evitamos que uma tarefa fique bloqueada desnecessariamente quando quer mexer só num tipo de variáveis. Há mais distribuição e mais liberdade nas execuções.

Para melhorar a compreensão da implementação, apresentamos a seguir fluxogramas das tarefas.

## Fluxogramas

### Tarefa Cliente

### 



### Tarefa Empregado



### Tarefa Gerente



# Conclusão

Para a última fase podemos dizer que o nosso trabalho está concluído de acordo com o pedido.

Desde a última fase realizamos algumas mudanças, sendo uma delas o ficheiro de configuração em que acrescentamos algumas configurações que achamos necessárias e adequadas. Mudamos também algumas das frases dos eventos por acharmos mais adequadas e acrescentamos uma no evento 1, “A loja vai fechar. Não podem entrar mais clientes.”, de maneira a que se saiba quando acabou o tempo de simulação.

Em anexo vamos entregar também um relatório de um teste de 5 minutos.

Na segunda fase foi apresentada uma ideia de sincronização que não foi a usada visto que os professores não concordaram porque simplificava o problema da sincronização. Então, como pedido, alteramos o nosso pensamento de modo a que cada entidade fosse uma tarefa.

Concluímos ainda que conseguimos aprender o funcionamento da sincronização. Achamos, também, que foi um projeto interessante.

# Anexos

Código do programa.

Read Me que explica como usar na linha de comandos.

Ficheiro com relatório de teste de 5 minutos com as suas estatística no fim do ficheiro.