|  |  |  |
| --- | --- | --- |
| **Departamento de Matemática da Universidade de Coimbra** | | |
| **2021 - 2022** | **Programação Orientada para os Objetos** | **Projeto 2** |

**Relatório do Projeto 2**

**Trabalho realizado por:** Margarida Biscaia e André Couto, Grupo 7.

**Especificação:** Uma empresa proprietária de uma cadeia de hipermercados, pretende construir um simulador de um sistema de filas de espera das caixas num supermercado. A empresa pretende abrir uma nova unidade e está interessada em saber quantas caixas necessita de instalar para que seja sempre possível atender os seus clientes sem que haja longas filas de espera (filas com quatro ou mais clientes). Para isso necessita de saber qual o número mínimo de caixas que serão necessárias de forma a evitar filas longas. Um dos objetivos do simulador é determinar esse número.

**Utilização:** A interface devolve os resultados da simulação na própria interface e os eventos de cada segundo escritos no ficheiro *dados.cvc*. Para alterar os parâmetros: DuraçãoDaSimulação e Lambda, o utilizador tem de alterar as constantes definidas em *eventos.hpp*:



**Diagrama UML:**

**Uma imagem com objeto de exterior, céu noturno

Descrição gerada automaticamente**

**Algoritmo:**

**- Class Evento**

**. Private:** Definimos uma fila f de clientes, um ponteiro para o cliente que está a ser atendido, o tempo para gerir o tempo da simulação, o número de clientes, o número de clientes atendidos, a soma dos tempos de espera e a soma total dos comprimentos da fila.

**. Public:**

A função *NumAleatorio* gera um número aleatório x com uma distribuição exponencial negativa de média , tal como descrito na secção “Geração de números aleatórios” do enunciado do projeto.

A função *ExecutaSimulacao* inicializa as variáveis todas a zero e o ponteiro para o cliente atendido a NULL. Enquanto o tempo de simulação não esgotar, coloca-se um cliente na fila (de acordo com o lambda definido) e processa-se a fila.

A função *ColocaClienteNaFila* incrementa o número de clientes da fila, e inicializa um novo cliente na fila. Gera também um tempo aleatório de atendimento. Para colocar o cliente na fila utilizamos a função push do contentor <queue>. Por fim, escreve num ficheiro o evento: entrada de cliente na fila.

A função *AtendeCliente* retira o primeiro cliente da fila e coloca o ponteiro de cliente atendido nesse cliente. O número de atendidos aumenta e os tempos de espera também. Por fim, escreve num ficheiro o evento: entrada de cliente na caixa.

A função *DispensaCliente* escreve num ficheiro o evento: cliente deixa a caixa e coloca o ponteiro de cliente atendido a NULL.

A função *ProcessaFila* vai processar a fila de clientes. Se nenhum cliente estiver a ser atendido, atende-se o primeiro cliente da fila. Caso contrário, se o tempo de atendimento for 0 então dispensamos o cliente, e senão, o tempo de atendimento diminui. Por fim, incrementa-se a soma dos comprimentos.

As funções *VerNumAtendidos, VerSomaTemposEspera* e *VerSomaComprimentos* devolvem a variável correspondente.

**- Class Cliente**

A classe Cliente define um tipo cliente com: número de cliente (numCliente), instante de entrada na fila (instanteEntrada) e tempo de atendimento (tempoAtendimento).

**Particularidades:** A simulação está feita apenas para uma caixa.

**Exemplo:**

**interface**

**Uma imagem com texto

Descrição gerada automaticamente**

**dados.cvc**

**Uma imagem com mesa

Descrição gerada automaticamente**