

UNIVERSIDADE ANHEMBI MORUMBI

CAMPUS PAULISTA

ESTRUTURA DE DADOS

SIMULAÇÃO DE ATENDIMENTO BANCÁRIO

Integrantes:

Eduarda Tobias Fernandes - RA: 12522216514;

Jennifer Reis Sicherolli De Souza - RA: 125221110547;

Katerine Linda Witkoski - RA: 1252221362;

Nathalie Mori Taylor Zampieri - RA: 1252224816;

São Paulo/SP - 2024

Relatório de Aplicação:

O objetivo do trabalho foi criar um algoritmo que determine o tempo médio que um cliente permanece na fila de uma agência bancária.

Foi definido que:

Na simulação a ser realizada, há dois eventos importantes:

- Um cliente chega à agência e entra na fila.
- Um guichê é liberado, alguém sai da fila e o utiliza.

Definição das classes:

Para resolução do problema criamos as seguintes classes:

Atendimento:

Classe Main, contendo uma verificação de ``GuicheLivre()`` e se ``ChegouCliente()``.

Pontos de lógica que consideramos:

- Se está dentro do tempo de expediente.
- Se a fila não está vazia.
- Se os guichês estão livres e qual guichê está livre.
- Realiza as transações (saque, depósito e pagamento).

A principal funcionalidade dessa classe é realizar os cálculos para a saída desejada. Contendo essas informações:

- Número total de clientes atendidos.
- Número de clientes que fizeram saque, depósito e pagamento.
- Tempo médio de espera na fila.
- Tempo extra de expediente.

Cliente:

A classe Cliente possui um atributo ``horaDeEntrada`` que armazena a hora de entrada de um cliente.

Há dois construtores:

Um construtor padrão que não inicializa a hora de entrada.

Um construtor que permite definir a hora de entrada no momento da criação do objeto Cliente.

FilaCliente:

Estrutura de lista encadeada, onde cada nó representa um cliente.

- `qtdeElementos`: Um inteiro que mantém o número de elementos na fila.
- `inicio`: Referência para o primeiro nó da lista (início da fila), a lista precisa começar vazia.

Métodos FilaCliente:

- `enfileirar`: adiciona um novo cliente na fila.
- `desenfileirar`: remove e retorna o tempo de entrada do cliente no início da fila.
- `filaEstaVazia`: verificar se a fila está vazia.
- `tamanhoFila`: retorna o número de elementos da fila.
- `mostrarInformacoes`: imprime as horas de entrada de todos os clientes.

Node

A classe Node encapsula um objeto `Cliente` e a referência para o próximo nó na lista encadeada. Ela é essencial para a implementação da lista encadeada usada pela classe `FilaCliente`.

Guiche:

A classe Guiche possui dois atributos:

- `livre`: Um booleano que indica se o guichê está disponível (true) ou não (false).
- `tempoOcupado`: Um inteiro que indica o tempo em que o guichê estará ocupado.

O construtor inicializa livre com o valor passado como parâmetro e tempoOcupado com 0, indicando que o guichê não está ocupado inicialmente.

ListaGuiche

A classe ListaGuiche possui quatro atributos:

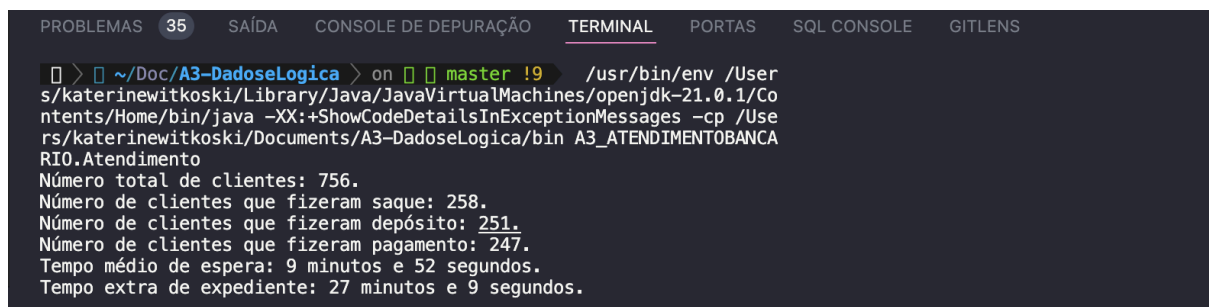
- `qtdeElementos`: Um inteiro que mantém o número de guichês na lista.
- `lista`: Um array de objetos `Guiche` que armazena os guichês. (classe Guiche)
- `inicio`: Índice que aponta para o início da lista.
- `fim`: Índice que aponta para o fim da lista.

Métodos ListaGuiche:

- `inserir`: adiciona um guichê à lista.
- `checar`: verifica se o guichê no índice especificado está livre.
- `obter`: retorna o guichê na lista.
- `tamanho`: retorna a quantidade de guichês na lista.
- `listaEstaVazia`: retorna se a lista está vazia.
- `listaEstaCheia`: retorna se a lista está cheia.

Resultado:

Resposta do console:



```
PROBLEMAS 35 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS SQL CONSOLE GITLENS
> ~/Doc/A3-DadoseLogica > on master !9 /usr/bin/env /User
s/katerinewitkoski/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Co
ntents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Use
rs/katerinewitkoski/Documents/A3-DadoseLogica/bin A3_ATENDIMENTOBANCA
RIO.Atendimento
Número total de clientes: 756.
Número de clientes que fizeram saque: 258.
Número de clientes que fizeram depósito: 251.
Número de clientes que fizeram pagamento: 247.
Tempo médio de espera: 9 minutos e 52 segundos.
Tempo extra de expediente: 27 minutos e 9 segundos.
```

Estrutura do projeto

O projeto foi estruturado em um pacote contendo todas as classes.

Para utilizar do sistema compile o arquivo `Atendimento`.

Conclusão

Através deste trabalho, conseguimos criar uma simulação realista do atendimento em uma agência bancária considerando eventos essenciais de modo a permitir calcular métricas importantes para a gestão do atendimento na agência.

Usar um cenário verossímil nos permitiu ter a percepção da importância de uma boa estrutura e planejamento na programação, além de vislumbrar resultados práticos.