Software de gerenciamento de restaurante

Bruno de Oliveira Jucá - 201965013A
Thales Brito de Souza Fonseca Rodrigues - 201965191A
Rômulo Chrispim de Mello - 201935038
Lowrran Durao Matias - 201935036
Pedro Henrique Moreira Raad - 201965215A

15 de Março de 2021

Resumo

O software de gerenciamento de restaurante busca simplificar e automatizar processos relacionados a gestão de um restaurante. Ele tem como objetivo gerenciar informações relacionadas ao funcionamento do estabelecimento e manter essas informações organizadas, realizando cálculos automáticos e auxiliando em processos de controle de caixa, controle de estoque, gerenciamento de comandas e pedidos, visualização de dados e armazenamento dos mesmos em arquivos de texto, para que se necessário, sejam carregados nas próximas utilizações do software.

Sumário

1 Objetivo e especificações

O objetivo do desenvolvimento deste software é atender aos requisitos do Trabalho Prático exigido pela disciplina de Orientação à Objetos, ministrada pelo professor Gleiph Ghiotto Lima De Menezes. A linguagem utilizada é Java e o projeto foi configurado de forma a ser construído através do Maven. Além disso, foi utilizado o sistema de gerenciamento de versão GIT, e o GitHub foi utilizado como local para o repositório remoto (`https://github.com/Grupo-de-OO/trabalho`).

2 Executando o código

Para compilar e executar o código, é necessário ter versão igual ou superior ao JDK 1.8. Além disso, é necessário ter o Maven instalado.

1. Execute o comando `maven install` no diretório raiz do projeto (onde se encontra o arquivo `pom.xml`). Este comando deve gerar uma pasta target, onde será armazenado o arquivo executável pela máquina virtual java, já com suas dependências.
2. Execute o comando no diretório raíz do projeto (é importante que o programa seja executado nesse diretório para que não haja problemas com os caminhos dos arquivos utilizados):
   `java -jar /target/trabalho-1.0-SNAPSHOT-jar-with-dependencies.jar`

3 Pacotes e Classes

3.1 Restaurante

O pacote Restaurante agrupa as classes responsáveis pelo gerenciamento das informações gerais do sistema.

3.1.1 Estoque

O Estoque surgiu da ideia de monitorar os ingredientes disponíveis e permitir consulta para identificar a possibilidade da construção de determinado prato ou bebida a partir dos dados contidos no estoque. Além de gerenciar as entradas de ingredientes e o gasto dos mesmos para a confecção dos pedidos.

3.1.2 ItemEstoque

A classe ItemEstoque tem o objetivo de auxiliar a criação de um array na classe Estoque com todos os itens disponíveis dentro do próprio Estoque.

3.1.3 Cardapio

A classe Cardapio tem o objetivo de organizar, naturalmente, o próprio cardápio do restaurante, contendo os possíveis itens que podem ser pedidos pelos clientes, além de permitir a entradas e saída dinâmica

dos dados, de acordo com a necessidade do usuário. A classe interage diretamente com a classe Estoque, podendo consultar a possibilidade da fabricação dos pratos a partir da disponibilidade dos ingredientes contidos no Estoque.

3.1.4 ItemCardapio

A classe Cardapio traz a necessidade da existência da classe ItemCardapio, que agrupa as principais informações de um item que deve constar no cardapio e que pode ser pedido ou não.

3.1.5 ListaComandas

Classe que contem como atributo uma lista do tipo Comanda além de métodos para a manipulação da lista que é utilizada na classe Comanda para o gerenciamento da comandas em aberto, com operações de inserir ou fechar uma comanda da lista.

3.1.6 Comanda

A classe Comanda tem o objetivo de armazenar informações e métodos de manipulação das comandas de cada cliente, listando um array do tipo ItemComanda que guarda a informação dos pedidos realizados. Atravéz dos métodos da classe, é possivel fazer a gestão dos pedidos além de computar o valor total que será passado para o caixa.

3.1.7 ItemComanda

Classe utilizada para a administração do contéudo das comandas criadas na classe Comanda.

3.1.8 Caixa

A classe Caixa deve cuidar da parte financeira do programa, cuidando das movimentações e do próprio balanço de caixa, sendo possível inserir tanto movimentações automáticas provenientes do fechamento de uma comanda quanto alterações manuais diretas.

3.1.9 Movimentacao

A classe Movimentacao deve definir o componente básico da classe Caixa, padronizando a estruturação dos dados.

3.1.10 Entrada

A classe Entrada herda a classe Movimentacao, permitindo apenas uma movimentação de entrada no caixa.

3.1.11 Saída

A classe Saida herda a classe Movimentacao, pemitindo apenas uma movimentação de saída no caixa.

3.2 Layout

O pacote layout concentra os elementos da interface gráfica do programa. O pacote Swing foi uitlizado como base e seus componentes foram utilizados para as criações dos painéis utilizados.

3.2.1 Header

Classe criada com o intuito de ser utilizada nas demais classes do pacote Layout. Ela é responsável por setar e organizar os *Headers* de cada painel de forma homogênea, para que fiquem modularizados de uma forma comum em todos. No construtor da classe é passado como parâmetro uma string com o título dos paineis que a utilizam.

3.2.2 InterfaceLayout

Interface criada para declarar a assinatura do médoto atualiza que foi implementado em todas as demais classes do pacote layout. Além de deixar declarado a constante dimensional das tabelas para que ambas compartilhem as mesmas dimensões independente dos paineis onde estão localizadas.

3.2.3 PainelCaixa, PainelCardapio, PainelComandas e PainelEstoque

Os painéis possuem estrutura parecida. Cada um deles implementa a interface gráfica relativa à classe representada no pacote Restaurante. Os componentes Swing utilizados dentro dos painéis foram, basicamente, botões, caixas de texto, tabelas, caixas de selecão, listas e caixas de diálogo. A disposição dos elementos gráficos foi feita manualmente através da utilização de *Layout Managers*.

3.2.4 FrameEvent

A classe FrameEvent é responsável por lidar com os eventos da janela. Ela usa o evento de abertura para carregar os dados do usuário armazenados em arquivos JSON. O evento de fechamento ativa o salvamento dos arquivos.

3.2.5 Contexto

O Contexto foi criado para armazenar as instâncias das classes do pacote Restaurante que são utilizadas ao longo do programa. Foi útil para que os paineis tivessem acesso de maneira prática a todos os objetos necessários.

3.2.6 Menu

Classe que faz a instanciação das demais classes do pacote Layout e onde chama os métodos principais para a execução do software.

3.3 Utils

Pacote que agrupa classes que foram utilizadas para a manipulação dos arquivos JSON utilizados no software.

3.3.1 Arquivo

A classe Arquivo cuida da manipulação dos arquivos jsons no computador, tanto em sua leitura como gravação dos seus dados.

3.3.2 Json

A classe Json recebe informações da classe arquivo e manipula suas strings para mapear o JSON para as variáveis do programa e carregar as informações.

4 Detalhes Complementares

O fluxo de execução do programa começa na classe Menu. Nela é instanciado um objeto da classe Contexto, que servirá de base para toda a aplicação. No Menu é criado o frame principal do programa. Nesse frame, adicionamos os painéis que serão utilizados, de forma que os botões laterais alternam a exibição de cada um dos painéis. Cada painel está intimamente relacionado com sua classe correspondente, instanciada no Contexto (Caixa, Estoque, ListaComandas, Cardápio), entretanto, também são realizadas alterações em outras classes, como em quando uma comanda é fechada e o valor da conta é automaticamente inserido como entrada no caixa. Ao longo de todo o desenvolvimento implementamos tratamentos de exceções em diversas casos, principalmente para que a entrada de dados feita pelo usuário fosse a esperada.

5 Conclusão

O desenvolvimento do software de gerenciamento de restaurante como trabalho requisitado pela disciplina de Orientação a Objetos se mostrou uma tarefa desafiadora, possibilitou um grande compartilhamento de conhecimento entre os integrantes do grupo e proporcionou o desenvolvimento tanto de habilidades técnicas quanto de habilidades de relacionamento interpessoal e de trabalho em equipe. O uso da ferramenta de gerenciamento de versão de códigos GIT e da plataforma de repositórios GitHub possibilitou fácil integração entre pequenas partes desenvolvidas por cada pessoa. O uso do GIT foi uma grande fonte de aprendizado, além da própria linguagem Java e, de maneira geral, do conceito de Orientação a Objetos. O software se apresenta funcional conceitualmente, entretanto reconhecemos que algumas melhorias poderiam ser realizadas, tanto na arquitetura e na organização do código quanto em funcionalidades.