Automação comercial: Terceira etapa do sistema
Sistema para gerenciamento de vendas para bares e restaurantes.

Com o desenvolvimento da segunda versão do sistema foi possível perceber que um sistema cresce rapidamente com o acréscimo de algumas funcionalidades, consequentemente, a manutenção, correção e extensão deste código torna-se cada vez mais uma tarefa complexa. Diante deste contexto, a aplicação de Padrões de Projeto (*Design Patterns*) é uma importante estratégia na Engenharia de Software. Um Padrão de Projeto é uma espécie de proposta de solução para problemas em comum ou, como define Erich Gamma e colegas, padrões de projeto são "descrições de objetos e classes comunicantes que precisam ser personalizadas para resolver um problema geral de projeto num contexto particular". Existem inúmeros padrões de projeto, tais como *Abstract Factory, Singleton, Observer, Facade, Strategy, entre tantos outros.*

Além disso, a interface com o usuário é outro importante requisito para qualquer sistema, ou seja, quanto mais fácil e amigável for a interação do usuário com o sistema, melhor será o nível de satisfação dele. Logo, o uso do terminal até este momento serviu apenas para que as funcionalidades fossem testadas pelos desenvolvedores. Sendo assim, é hora de dar um "tapa" no visual deste sistema!

ESPECIFICAÇÕES

Nesta etapa do projeto, deverá ser utilizado o uso do padrão Modelo-Visão-Controle (*Model View Controler* - MVC) para garantir e facilitar a manutenção do código, portanto, este deverá ser dividido entre *models*, *views* e *controllers*.

Devem ser implementados um dos seguintes padrões GoF: Facade ou Abstract Factory.

Uma nova interface (gráfica!) com o usuário deverá ser implementada utilizando JavaFX, além das seguintes funcionalidades:

- Criação de uma nova entidade: Cliente (nome, cpf, email, telefone);
- Cadastrar cliente;
- Registrar a venda de cada cliente, constando: cliente, pratos e valor final;
- Gerar relatório/nota da compra do cliente.

Observação: todas as entradas/alterações dos dados devem ser realizadas através da interface.

PRODUTO

O produto poderá ser realizado **individualmente** ou em **dupla**.

Os artefatos gerados deverão ser entregues pelo *Classroom* em arquivo único compactado, até às 23:59 do dia **28/06/22 (terça-feira)**. São eles: **código-fonte, documentação (javadoc), testes de unidade (com suíte de testes) e diagrama de classes (somente entidades)**.

O código fonte deverá ser implementado na última versão estável do Java (utilize a versão opensource do JDK, OpenJDK) e deverá estar todo documentado seguindo o padrão Javadoc.

O código fonte deverá ser implementado na IDE **Eclipse**[1]. Para os testes, deve-se utilizar o framework JUnit 5. Portanto, tudo que você desenvolveu precisa atender aos seus respectivos testes.

OBSERVAÇÕES

A entrega após a data e horário definidos implicará em um desconto de 2 pontos na nota do produto. Após 24 horas de atraso, será descontado mais 1 ponto na nota. O problema não será mais recebido após 48hs de atraso. Antes de enviar o produto, certifique-se de ter lido o documento “Evitando cópias indevidas em trabalhos acadêmicos”, disponibilizado no *Google Classroom* e de ter adicionado a declaração de autoria do código nas classes desenvolvidas.

Todas as classes devem estar compilando e implementando as funcionalidades adequadamente. Todos os testes devem estar rodando e passando. Todas as classes, atributos e métodos que você criou devem estar documentados utilizando o padrão Javadoc. A correção será feita a partir da análise do código, execução dos testes de unidade e análise da documentação Javadoc. O produto entregue corresponde a 70% da nota do problema e o desempenho nos tutoriais corresponderá a 30% da nota.

CRONOGRAMA

AULA	DIA	ASSUNTO
1	17/mai	Apresentação Problema 3
2	24/mai	Problema 3
3	31/mai	Problema 3
4	07/jun	Problema 3
5	14/jun	Problema 3
6	21/jun	Problema 3

[1] https://www.eclipse.org/