

# **Avaliação P1 - ECM251 - Linguagens de Programação I**

Prof. Murilo Zanini de Carvalho

[murilo.carvalho@maua.br](mailto:murilo.carvalho@maua.br)

## **\_Instruções gerais para avaliação**

Instruções gerais para realização da avaliação:

- A avaliação deve ser realizada de forma individual;
- Qualquer tipo de consulta ao material desenvolvido pelo professor ou a outros livros texto pode ser realizado;
- Sobre nenhuma hipótese, seu código, parcialmente ou integralmente, deve ser compartilhado com outros alunos. Qualquer comportamento suspeito, que possa ser verificado com a análise dos códigos, será penalizado com os critérios previstos no regimento da instituição;
- Duas datas ficam reservadas para interações com o professor para tirar dúvidas sobre o desenvolvimento da atividade:
  - o 15/06/2020 - das 9h30 até às 13h00;
  - o 22/06/2020 - das 9h30 até às 13h00.
- Durante as interações com o professor, questões de implementação poderão ser discutidas, mas a forma como cada um dos alunos vai fazer a implementação do código também será avaliada;
- A participação dos alunos nas interações é opcional;
- Além da interação online síncrona nas datas apresentadas, os alunos podem enviar dúvidas para o docente por e-mail ([murilo.carvalho@maua.br](mailto:murilo.carvalho@maua.br)) ou no canal de comunicação utilizado na disciplina, o Slack (<https://3d-2020-maua.slack.com>). As perguntas serão respondidas em até 72 horas.
- Cada nova classe, interface, enumeração que for criada para solução do problema, deve ser registrado no serviço de controle de versão utilizado, no nosso caso, no Git. Trabalhos que não tiverem um histórico de desenvolvimento (as etapas para a construção da solução), serão desconsiderados e terão a nota zero atribuídos a ele.

## Descrição do problema

A Pizzaria “o Rato que Ri” está com sérios problemas. O seu antigo sistema de pedidos está uma completa zona. Quando você soube que a sua pizzaria favorita estava com problemas, afinal, a pizza de muçarela deles é imbatível, decidiu ajudar.

Ao conversar com o dono do negócio, ele abre o coração para você, contando todas as suas amarguras na vida, tudo que ele já se arrependeu de não ter feito, que ele gostaria de ter escolhido o Bulbasaur, mas por ter escolhido o Charmander, ele acredita que tudo na vida tomou o caminho atual. Neste momento, você avisa que estava se referindo ao sistema de pedido. Ele se recompõe e diz que o problema é que tudo que ele fazia era controlado por uma planilha do Excel, que a sua sobrinha de 4 anos acabou jogando fora de forma accidental.

Ao ouvir a descrição do problema, você percebeu que está seria uma ótima oportunidade de colocar em teste seus conhecimentos de orientação a objetos. Você começa a perguntar para o dono quais as funcionalidades que ele gostaria que o sistema possuísse. Vocês chegaram a conclusão de que parte do sistema seria implementada como teste para que o dono pudesse ver se você deveria terminar a implementação e ganhar pizza grátis para o restante de sua vida.

Vocês concordaram que o sistema deveria possuir apenas **um usuário** por enquanto. Esse usuário seria necessário para as operações que **geram ou alteram algum pedido**, como criar um pedido ou alterar um pedido existente. O usuário deve possuir um **nome, um e-mail e uma senha**, que neste momento, já devem estar cadastrados no sistema. A senha que o dono pediu, para este teste, é **“123456”**.

Além do usuário, seu sistema deve possuir um **modelo de pedido** que possibilite cadastrar os seguintes campos: **um id para o pedido, sua descrição, seu valor, uma forma de pagamento e um estado para ele**. O id do pedido deve ser um texto contendo **alguns números aleatórios gerados pelo sistema**. A descrição do pedido deve conter as informações do que consiste o pedido. A forma de pagamento deve ser uma das seguintes: **DINHEIRO, DEBITO, CREDITO, VALE\_ALIMENTACAO, VALE\_REFEICAO**. O estado de todo novo pedido deve ser **REALIZADO**, os demais estados que ele pode assumir são: **REALIZADO, PREPARACAO, SAIU\_PARA\_ENTREGA, ENTREGUE, DEVOLVIDO**.

Seu sistema deve apresentar uma interface similar a:

```
Pizzaria o Rato que Ri:
1 - Nova venda
2 - Verificar pedidos
3 - Alterar pedidos
0 - Sair
```

A lista com todos os pedidos, por hora, deve ficar no sistema principal. Qualquer funcionário pode verificar os pedidos que foram realizados, mas para realizar novos pedidos ou alterar os existentes, o funcionário deve informar sua senha.

Para modificar um pedido, alterar apenas o seu estado. Para separar os pedidos, utilizar o valor de seu id. Para gerar os id's, utilizar a seguinte função:

```
private String geradorId() {
    Random random = new Random();
    String idGerado = "";
    for(int i = 0; i < 3; i++)
        idGerado += random.nextInt(10);
    return idGerado;
}
```

Quando o usuário desejar sair do sistema, ele deve ser fechado. Assim você ficou feliz por poder praticar ao mesmo que ajuda sua pizzeria favorita.

De forma inocente e cheio de orgulho, você foi comentar com seu professor de programação a respeito do trabalho. Ao ouvir sua descrição, ele ficou super empolgado e pediu para você lembrar de utilizar ao menos uma interface para o sistema (parece que a autenticação do usuário um bom usuário), utilizar enumerações onde fizer sentido, lembrar dos 4 pilares da orientação a objetos (deixar apenas os setters e getters que forem necessários, sempre programando para interface onde for possível, apenas para citar alguns) e para você separar suas classes em pacotes que façam sentido.

Ao ver a mistura de confusão de medo e preocupação em seus olhos, ele entendeu que talvez você tivesse se esquecido quais seriam bons nomes de pacotes para utilizar, então ele te sugeriu:

Com medo das reações futuras, você decide sair. Quando, neste momento, ele anuncia para sala que realizar essa implementação seria a avaliação P1 de todos.

## **\_Pré-requisitos**

Alguns dos pré-requisitos para o desenvolvimento da sua solução:

- A versão do Java que será considerada para as soluções é o **Java JDK 8** (Amazon Corretto 8 - <https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html>).
- O aluno é livre para escolher a ferramenta de desenvolvimento que ele julgar mais conveniente para elaboração da solução. Recomenda-se o uso da IDE **IntelliJ** CE (<https://download.jetbrains.com/idea/ideaIC-2020.1.2.exe>) ou do editor de texto Visual Studio Code (<https://code.visualstudio.com/>) com os plug-ins para desenvolvimento em Java (<https://code.visualstudio.com/docs/languages/java>).
- Recomenda-se o uso da ferramenta **GitHub for Desktop** (<https://desktop.github.com/>), para fazer as atualizações no repositório remoto, GitHub.
- No repositório, colocar toda a documentação gerada pelo **Javadoc** do projeto.
- O repositório da avaliação pode ser o seu pessoal, apenas indicar onde está o diretório da solução.
- 

## **\_Forma de entrega**

A entrega deve acontecer de forma virtual, até o dia 26/06/2020, às 23h59. O aluno deve enviar uma mensagem privada no Slack para o professor com o link para o repositório. O professor vai responder essa mensagem confirmando que ele já clonou a solução apresentada pelo aluno.

**ATENÇÃO:** a data limite para realizar a entrega é dia 26/06/2020, não sendo possível realizar nenhum tipo de prorrogação nesse prazo. Qualquer problema que o aluno tiver, ele deve ser reportado antes ao professor e a entrega da solução deve ser antecipada. Repetindo, sobre nenhuma condição o trabalho será aceito fora do prazo.

## Pontuação escolhida por item

<b>Critério</b>	<b>Peso</b>
Pilar da abstração da Orientação a Objetos	2
Pilar do encapsulamento da Orientação a Objetos	2
Pilar da herança da Orientação a Objetos	2
Pilar do polimorfismo da Orientação a Objetos	2
Coesão de código	2
Utilização correta dos pacotes	1
Desenvolvimento da(s) interface(s)	2
Descrição dos modelos utilizados na aplicação	2
Codificação das enumerações do sistema	2
Realizar toda a comunicação com o usuário apenas na classe do sistema	2
Documentar todo o projeto (Javadoc)	8