

# Projeto

## Modelando um sistema Marketplace

*Projetos de software são oportunidades práticas para que o.a.s aluno.a.s da disciplina Princípios de Orientação a Objetos demonstrem os objetivos de aprendizagem da disciplina. Portanto, um dos requisitos é que o código elaborado e o sistema desenvolvido estejam de acordo com as boas práticas apresentadas e discutidas nesse período.*

### Descrição

Este é um projeto individual.

Nesse projeto será criado um programa com funcionalidade similar a um marketplace. O marketplace é um e-commerce, mediado por uma empresa, em que vários lojistas se inscrevem e vendem seus produtos. Essa loja virtual funciona de forma que o cliente pode acessar um site e comprar itens de diferentes varejistas. No marketplace um usuário pode ser tanto um vendedor quanto um comprador.

O aluno deverá escolher um tema de Marketplace especializado da lista a seguir para basear seu projeto:

- Locação de quartos, casas, apartamentos etc., por quantidades de dias. Ex. Airbnb, Spotahome etc.
- Refeições, produtos de mercado, assinaturas de refeições etc. Ex. Ifood, Uber Eats etc.
- Cursos por prazo indefinido ou por período, formação com vários cursos etc. Ex. Coursera, Udemy etc.
- Livros, revistas, assinaturas de revistas, coleções de livros etc. Ex. Estante Virtual, Saraiva, Amazon Livros etc.

A escolha do aluno será limitada e pela escolha de outros alunos, e deve ser confirmada pelo professor de forma que a quantidade de alunos com cada projeto seja equilibrada. É recomendado que o aluno faça um perfil no Marketplace escolhido para conhecer como funciona por dentro.

Esse projeto será desenvolvido em duas etapas, P1 e P2. Este documento descreve a etapa P1, que é individual, e nela os casos de uso e requisitos estão reduzidos para serem desenvolvidos individualmente.

## Casos de uso:

- O usuário pode listar todos os produtos anunciados no Marketplace. A lista de produtos exibe um código do produto que pode ser usado em outras funções. A lista de produtos/serviços exibe no máximo 10 produtos/serviços por vez e um comando pode ser dado para exibir mais 10.
- O usuário pode exibir os detalhes de um anúncio de produto/serviço e dos fretes disponíveis no Marketplace.
- O usuário pode cadastrar um tipo de frete com atributos como empresa, prazo máximo e valor mínimo. Ex. Frete por Loggi em até 3 dias úteis e valor mínimo de R\$30,00. Frete por correios em até 14 dias úteis com valor mínimo de R\$14,00. Frete por Ifood em até 30 minutos por no mínimo R\$5,00.
- O usuário pode listar todos os tipos de frete. A lista de fretes exibe um código do cadastro do frete que pode ser usado em outras funções. A lista exibe no máximo 10 fretes por vez e um comando pode ser dado para exibir mais 10.
- O usuário pode editar um tipo de frete.
- O usuário pode excluir um tipo de frete.
- O usuário pode cadastrar um anúncio de um produto/serviço de uma lista de pelo menos 3 categorias diferentes possíveis, descrevendo o anúncio com pelo menos 3 atributos além de atributos comuns como título do anúncio, preço, quantidade.
- O usuário pode editar um anúncio.
- O usuário pode excluir um anúncio.

# Requisitos

- Ao executar o programa, ele deve criar automaticamente pelo menos 21 produtos/serviços anunciados de tipos variados.
- A interface com o usuário deve ser textual no console com menus numéricos de 0 a 9.
- A classe principal de execução do projeto deve se chamar Principal.java, que deve apresentar a interface com o usuário.
- O projeto deve ter outra classe de execução, a TesteDeSistema.java, que deve automaticamente executar os casos de uso do sistema com mensagens no console informando o que irá fazer e qual o resultado.
- O programa deve tratar todas as exceções possíveis, portanto não pode encerrar sua execução devido a uma exceção.
- Deve ter comentários descrevendo o que o programa deve fazer, o que cada classe e cada método devem fazer e o que as principais linhas de códigos fazem. Os nomes dados às variáveis, classe e métodos devem ser auto descritivos.
- Não deve haver códigos repetidos, nem atributos ou métodos repetidos no programa, e para garantir isso o aluno deverá usar: Herança, Polimorfismo, Interfaces, Classes Abstratas etc.
- O paradigma de encapsulamento deve ser seguido, portanto todos os atributos de classes devem estar como "private" a não ser que seja justificado o uso de outro modificador.
- O aluno deverá usar Collections (pelo menos List) e Generics;

# Entregas e prazos

- 22/08
  - Documento com descrição geral do projeto e diagrama de classes UML. Entregue como documento do Google Docs.
  - Projeto Java em Eclipse ou compatível com Eclipse. Entregue em único arquivo compactado no formato ZIP
- 23/08 - Entrevista individual com sobre os detalhes de desenvolvimento do projeto (não pode ser postergada).

# ATENÇÃO:

- Similaridades frequentes, plágio e cópia eliminam/desclassificam a apresentação de todo.a.s o.a.s envolvido.a.s!
- Entregas deverão ser feitas pelo Google Sala de Aula a não ser que seja informado outro destino.
- Durante as aulas as equipes podem e devem fazer consultas ao professor sobre o projeto.
- O não cumprimento dos prazos levam a multa de 30% por dia de atraso na nota do item sendo entregue.