

# Projeto Final da Disciplina POO 2020.2

## Objetivo

O trabalho tem por objetivo modelar e implementar um sistema utilizando o paradigma Orientado a Objetos aplicando os conceitos vistos no curso.

## Equipe

O trabalho deve ser feito prioritariamente em dupla.

## Requisitos

- O sistema deve ter uma classe *Aplicacao*, responsável pela interação com o usuário utilizando uma main iterativa, conforme temos visto nos exercícios.
- O sistema deve ter pelo menos uma classe *Visao*, responsável pela exibição (métodos show)
- O sistema deve ter pelo menos uma classe *Controle*, responsável por atender às solicitações do usuário
- Além das classes já citadas acima, o sistema deve conter pelo menos 3 classes concretas que modelam a aplicação.
- O sistema deve conter ao menos uma classe abstrata.
- As classes de modelo devem implementar a interface *Comparable*, que irá definir a forma de ordenação das instâncias.
- O sistema deve prover funcionalidades de um CRUD: criação, edição, exclusão e listagem ordenada
- O sistema deve implementar ao menos duas regras de negócio.
  - Considere uma regra de negócio uma funcionalidade além daquelas descritas no CRUD.
- O sistema deve utilizar tratamento de exceções.

## Cronograma: entregáveis e datas

### Até 03/03 - Envio da equipe e tema:

- Incluir na [planilha](#) o nome completo e matrícula da dupla e o tema escolhido para a aplicação.
- Os temas não podem se repetir, portanto será respeitada a ordem de inclusão do tema na planilha.

### 12/03 - Relatório versão 1:

- No relatório deve constar a descrição do sistema proposto, com detalhamento dos requisitos do sistema (funcionalidades) e a primeira versão do diagrama de classes UML.
- Envio dos entregáveis via moodle

**26/03 - Relatório versão 2 e código-fonte versão 1:**

- Código-fonte da versão 1 da aplicação
- A aplicação deve estar funcionando com interação com usuário com pelo menos 6 funcionalidades já implementadas
- No relatório deve constar quais funcionalidades já foram implementadas e quais atividades foram realizadas por cada um dos membros da equipe.
- O diagrama deve estar atualizado caso haja necessidade de mudanças na modelagem.
- Envio dos entregáveis via moodle

**07/04 - Apresentação, relatório final, código-fonte e aplicação:**

- Cada equipe deve preparar uma apresentação contemplando a execução de algumas funcionalidades da aplicação (no máximo 5 minutos)
- Arguição sobre o projeto
- Envio dos entregáveis via moodle

## **Sugestões de temas**

Esses temas são ideias as quais vocês podem se inspirar e trabalhar as possíveis funcionalidades. Os temas não são restritos à listagem abaixo.

- Adoção de pets
- Acompanhamento do seu pet
- Divulgação e contratação de serviços
- Planejamento financeiro
- Planejamento de estudos
- Alarme e controle de medicamentos
- Lista de compras
- Compartilhamento de caronas
- Denúncias COVID-19
- Tabela nutricional
- Guia turístico
- Plano de leitura pessoal
- Empréstimo/Troca de livros

Modelar um sistema de seu interesse que apresente as seguintes características:

- deve apresentar no mínimo cinco (5) classes;
- deve existir hierarquia de herança entre algumas classes, podendo haver classes abstratas;
- deve haver pelo menos uma classe de agregação que agrupará instâncias das outras classes (lembre-se do array de publicações da classe Biblioteca). Dá-se o nome de Agregação a esse processo de agregar objetos de outras classes. Por exemplo, Banco contém um array de n Contas, Veículo contém um array de 4 pneus, um motor, etc.;
- deve haver métodos para incluir, excluir e consultar elementos da classe de agregação;
- deve haver uma classe Aplicacao, além das outras já definidas, com método *main*, que criará e manipulará instâncias das classes de sua aplicação;
- deve apresentar pelo menos uma interface que deve ser implementada em alguma(s) classe(s);
- devem ser fornecidas as variáveis de instância e métodos para cada classe;
- as classes devem possuir métodos sobrecarregados (mesmo nome, diferentes parâmetros);
- as classes devem apresentar construtores sobrecarregados;
- as subclasses têm que possuir métodos redefinidos (não necessariamente todos);
- as classes devem ser encapsuladas;
- a aplicação deve trabalhar com exceções (padrões ou personalizadas) sempre que possível;
- será levado em consideração também o padrão de codificação (nomes de variáveis, classes, etc.) estudado ao longo do semestre;
- estruturar a aplicação em pacotes;
- apresentar o diagrama de classes UML completo de sua aplicação;
- procure utilizar em sua aplicação, sempre que possível, os conceitos de Orientação a Objetos abordados em sala de aula: herança, agregação, classes abstratas, polimorfismo, interfaces, etc. Quanto mais conceitos sua aplicação utilizar, mais bem avaliado será seu trabalho.
- Apresentar uma interface gráfica que facilite a utilização do sistema;
- inicialize o sistema com alguns dados iniciais, para facilitar a apresentação.

### **As Equipes**

Cada equipe deve possuir de 2 a 3 integrantes. Organizem-se e escolham uma aplicação p/ cada equipe.

### **Forma de Avaliação**

A nota do trabalho levará em consideração:

- diagrama de classes UML;
- classes codificadas (arquivos java) em formato digital;
- apresentação da aplicação rodando no computador do laboratório em horário de aula;
- aguçação individual sobre o trabalho.

### **Data das Apresentações dos Trabalhos**

Os trabalhos serão apresentados em laboratório, seguindo ordem de sorteio, nas datas abaixo:

11/12	Segunda-feira
-------	---------------

### Sugestões de Sistemas

Cada equipe poderá escolher entre as sugestões de sistema fornecidas abaixo ou outra aplicação de interesse da equipe, podendo estender ou melhorar as sugestões fornecidas. As aplicações sugeridas abaixo são, apenas, como já dito, sugestões. A equipe poderá usar a imaginação para inventar uma nova aplicação ou incrementar alguma sugerida.

1

<b>Aplicação</b>	<b>Sistema de Aluguel de Veículos</b>
<i>Classes</i>	Veículo, Aluguel, Cliente
<i>Sub-classes</i>	Veículo: carro de passeio, caminhão, ônibus, moto, etc
<i>Agregação</i>	Locadora de Veículos, que possuirá um conjunto de veículos disponíveis para aluguel.
<i>Interface</i>	"Multável" aluguel pode sofrer multa por atraso.

2

<b>Aplicação</b>	<b>Sistema de Controle de Alunos</b>
<i>Classes</i>	Aluno
<i>Sub-classes</i>	de Aluno: Aluno de graduação, Aluno de pós-graduação; de Aluno de graduação: Aluno de Iniciação científica (possui bolsa de estudo)
<i>Agregação</i>	Curso, que conterà um conjunto de alunos
<i>Interface</i>	As classes Aluno de IC e Aluno de Pós-graduação implementam uma interface chamada <i>Remunerável</i> que tem como propriedades os dias trabalhados e o valor da bolsa. Pensar nos métodos dessa interface.

3

<b>Aplicação</b>	<b>Sistema de Gestão de Companhias Aéreas</b>
<i>Super-classe</i>	Aeronave
<i>Sub-classes</i>	de Aeronave: Aeronave de Carga, Aeronave de Passageiros
<i>Agregação</i>	Companhia Aérea, que conterà um conjunto de aeronaves
<i>Interface</i>	Automática (que pode ser guiada via piloto automático)

4

<b>Aplicação</b>	<b>Sistema de Controle de Passagens Aéreas</b>
<i>Classes</i>	Passagem, Cliente
<i>Sub-classes</i>	de Passagem: Passagem de Primeira Classe, Passagem de Classe Econômica
<i>Agregação</i>	Vôo será uma classe que agregará diversas passagens.
<i>Interface</i>	"Transferível" (que pode ser transferida para outra pessoa).

5

<b>Aplicação</b>	<b>Sistema de Controle de Clubes de Futebol</b>
<i>Classes</i>	Clube, Jogadores, Técnico
<i>Sub-classes</i>	de Clube: clube de 1ª divisão (tem comissão de tv, por ex.), clube de 2ª divisão,...
<i>Agregação</i>	Confederação
<i>Interface</i>	"Comissionável" (aquele clube que pode receber comissão da TV).

6

<b>Aplicação</b>	<b>Sistema de Controle de Volumes dos Correios</b>
<i>Classes</i>	Correspondência, Remetente, Destinatário
<i>Sub-classes</i>	De Correspondência: Cartas, Encomendas (tem peso)
<i>Agregação</i>	Loja dos Correios
<i>Interface</i>	"Retornável" (aquele pacote que não pôde ser entregue).

7

<b>Aplicação</b>	<b>Imobiliária</b>
<i>Super-classe</i>	Imóvel, Cliente, Aluguel

UFC Quixadá - Programação Orientada a Objetos 2020.2 - Profa. Livia Almada

Sub-classes	de Imóvel: Casa, Apartamento, Sítio
Agregação	Imobiliária
Interface	Multável (aluguel pode sofrer multa por atraso)
8	
<b>Aplicação</b>	<b>Gestão de Planos de Saúde</b>
Classes	Plano de Saúde, Cliente, Contrato
Sub-classes	De Plano de Saúde: Plano Empresa, Plano Individual, Plano Enfermaria, Plano Apartamento
Agregação	Cooperativa
Interface	
9	
<b>Aplicação</b>	<b>Sistema de Hotelaria</b>
Classes	Meios de Hospedagem, Cliente, ReservaImpl
Sub-classes	Hotel, Pousada
Agregação	Complexo Turístico
Interface	Reserva (registrar reserva, cancelar, efetivar)
10	
<b>Aplicação</b>	<b>Gestão de Biblioteca</b>
Super-classe	Publicação
Sub-classes	Livros, Revistas
Agregação	Biblioteca
Interface	Gestão de Biblioteca