

		<b>Primeira Avaliação Parcial Primeiro Semestre de 2023</b>	
Curso: <b>Engenharia de Software</b>			
Disciplina: <b>Projeto de Sistemas Orientados a Objetos</b>		Turma: <b>A</b>	Período: <b>3</b>
Professor: <b>Paulo Sergio da Conceição Moreira</b>		Data: <b>10/04/2023</b>	Nota:
Aluno (a):			
<b>Instruções:</b> <ul style="list-style-type: none"> <li>• Prova individual;</li> <li>• A prova tem valor máximo de 40 pontos;</li> <li>• A entrega deve ser realizar via ambiente virtual, com o envio dos respectivos arquivos com os códigos;</li> <li>• A prova cujo código não puder ser replicado, receberá nota zero;</li> <li>• A prova cujo status no ambiente virtual constar como rascunho, ao final do prazo de entrega, receberá nota zero.</li> </ul>			

## IMPLEMENTAÇÃO DE SISTEMA PARA CÁLCULO DE CORRIDAS DE APLICATIVOS

Como estudante do curso de Engenharia de Software, você foi contratado para desenvolver um sistema de cálculo de corridas para um motorista de aplicativo. Como requisito, o sistema deve conter classes relacionadas à viagem, à forma de pagamento, à corrida e ao motorista.

Para facilitar a sua implementação, o cliente forneceu as seguintes especificações:

- Viagem: deve ter como atributos a origem, o destino, a distância em km e o preço da viagem;
- Pagamento: deve ter como atributos o valor e a forma de pagamento (que pode ser "dinheiro", "cartao" ou "pix");

- c) Corrida: deve ter como atributos a distância em km, o tempo em minutos e a forma de pagamento. Como métodos, esta classe deve ter os métodos `calcular_preco_base`, `calcular_desconto` e `valor_total`. O método `calcular_preco_base` deve calcular o preço base da corrida, considerando duas vezes a distância percorrida mais metade do tempo da corrida. O método `calcular_desconto` deve calcular o desconto de acordo com a forma de pagamento, que é de 10% para pagamentos via pix e 5% para pagamentos via cartão. O método `valor_total` deve retornar o valor total da corrida, já com o desconto aplicado;
- d) A classe Motorista deve ter como atributo o nome do motorista e uma lista de corridas realizadas. Ela deve ter os métodos `adicionar_corrida` e `calcular_total_recebido`. O método `adicionar_corrida` deve adicionar uma nova corrida à lista de corridas realizadas pelo motorista. O método `calcular_total_recebido` deve calcular o total recebido pelo motorista em todas as corridas realizadas, levando em consideração os descontos aplicados de acordo com a forma de pagamento.

### **CONSIDERAÇÕES**

1. Pontos a serem avaliados: a) implementação correta das classes, tanto quanto aos atributos, quanto aos métodos; b) criação dos objetos, de forma a avaliar se a solução apresentada corresponde ao proposto na demanda; c) validações e tratamentos de possíveis erros, como valores negativos e nulos; d) legibilidade do código (nome de variáveis, métodos, comentários, descrição de funções etc.);
2. Todo código apresentado na solução será avaliado, e a nota será proporcional. Deste modo, mesmo que não apresente a solução “completa”, todo estudante receberá uma nota, de acordo com o avanço que apresentou para a demanda proposta;
3. O desenvolvimento de interfaces gráficas, integrações com banco de dados ou outros recursos que não fazem parte do escopo da disciplina não será avaliado. Todavia, caso a reprodução da solução apresentada seja comprometida por conta dessas soluções, seja por erro na implementação ou falta de arquivos e/ou bibliotecas, haverá redução de pontos.