



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TRABALHO COMPUTACIONAL

Disciplina: Algoritmos e Programação 2

Professores: Diego Américo Guedes e Ricardo Augusto Pereira Franco

Curso: Engenharia de Computação

Data da entrega: **08/12/2020**

Data da apresentação: **11/12/2020**

Diretrizes:

O **trabalho final** está dividido em **três partes**:

- 1ª** Entrega do programa desenvolvido (programa aplicação);
- 2ª** Entrega do documento explicativo;
- 3ª** Apresentação do programa entregue.

O programa deverá ser entregue através da submissão de arquivos contendo o código Java (.java) da aplicação na Plataforma Turing. O código **deverá ser comentado** para auxiliar sua compreensão e avaliação.

O documento explicativo deverá conter as informações e considerações realizadas pelo grupo para a implementação do código, além de conter o diagrama de classes UML desenvolvido para a aplicação.

O programa aplicação e o documento explicativo deverão ser entregues juntos (via Plataforma Turing) **até o dia 08/12/2020 às 23:59. Não serão aceitos trabalhos entregues após o limite definido.**

Para os grupos que não entregarem o trabalho ou entregarem o trabalho após o prazo definido, será atribuída nota 0,0 (nota zero).

A **apresentação do programa** deverá ser feita por **todos** os integrantes do grupo, demonstrando seus conhecimentos a respeito da Programação Orientada a Objetos e da linguagem Java, bem como sua contribuição no desenvolvimento da aplicação. A nota de apresentação **será única** para todos os integrantes do grupo.

A apresentação do programa será realizada no dia 11/12/2020 das 7:10 até às 8:50, em sua respectiva turma (aula prática).

A não participação do(a) aluno(a) na apresentação do trabalho resultará em atribuição de nota 0,0 (zero) para o(a) mesmo(a).

O trabalho final deverá ser feito em grupo de 5 (cinco) pessoas da mesma turma.

Para trabalhos entregues definidos como plágios, cópias idênticas e cópias disfarçadas, será atribuída nota 0,0 (zero) para todos os grupos envolvidos. Plágios, cópias idênticas e cópias disfarçadas serão definidas após análises do documento explicativo e programa aplicação entregues pelos grupos.

Dúvidas que não tenham sido respondidas por este documento, serão respondidas pelo professor de sua turma ou por ambos.



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Definição do projeto de software:

Baseado nas descrições (requisitos) dos clientes fornecidas abaixo, implemente um sistema de passagens aéreas utilizando os conceitos de Programação Orientada a Objetos (POO) e na Linguagem Java.

SISTEMA DE PASSAGENS

Descrição Geral: "Um sistema automático de emissão de passagens vende passagens de avião. A partir de uma lista de possíveis destinos, os usuários escolhem seu destino e apresentam um cartão de crédito ou de débito e um número de identificação pessoal exclusivo. Os destinos possíveis devem ser organizados de modo a facilitar a escolha. Após a escolha do destino, o sistema deve responder prontamente se há espaço disponível no voo. A passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito ou debitado de sua conta (caso utilize o cartão de débito). Quando o usuário inicia o programa, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino. Uma vez selecionado um destino e a quantidade de adultos e crianças, pede-se que os usuários insiram o número de seu cartão de crédito/débito. O saldo da conta ou a validade do cartão são checados e, então, o programa deverá fornecer um número de identificação pessoal (código da compra-*voucher*). Quando a transação de crédito é validada ou o valor é debitado da conta, a passagem aérea é emitida. O formato do bilhete de passagem deve ser definido pelos programadores."

Descrição Específica: O sistema deve oferecer a busca e compra de passagens aéreas.

Informações: Local de origem, local de destino, data de partida, data de volta, quantidade de adultos, quantidade de crianças, idade das crianças.

O sistema deve permitir que o cliente do sistema possa executar a busca fornecendo as informações obrigatórias e, caso haja resultado para a consulta, o programa deve disponibilizar as passagens aéreas com seus preços, origem, destino e horário. Caso não haja disponibilidade, deve ser informado ao cliente que não houve resultado para a consulta.

O programa deve validar se as informações sobre local de origem, local de destino, data de partida, data da volta e número de pessoas estão preenchidos.

A data de partida deve ser maior ou igual à data corrente.

A data de volta deve ser maior ou igual à data de partida.

O número de adultos ou o número de crianças deve ser maior que zero.

Caso o número de crianças seja maior que 0 (zero), a idade da criança deve ser preenchida.

O valor das passagens para crianças pode ser diferente do valor das passagens para adultos. Assim, o valor deve ser apresentado baseado em alguma regra, por exemplo: "Crianças acima de 6 (seis) anos devem pagar valor integral".



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Caso alguma das informações acima não cumpra essas regras, o programa deve informar o usuário a respeito do erro de preenchimento.

Observações:

- Em um primeiro momento, o trabalho pode parecer extenso e complexo. Contudo, com uma boa divisão de tarefas será possível alcançar excelentes resultados e uma boa distribuição de carga do trabalho. Reúna seu grupo, discuta suas ideias e divida o programa em módulos e entre os integrantes, assim, cada um contribuirá com o desenvolvimento do trabalho.
- Desenvolvam o diagrama de classes UML do programa aplicação antes de iniciar a implementação do código.
- Utilizem os conceitos estudados em POO (ex.: classes, classes abstratas, interfaces, etc.) para dividir a implementação do trabalho.
- A avaliação do trabalho levará em consideração os conceitos de POO utilizados no desenvolvimento do programa, além de avaliar os requisitos atendidos do programa aplicação (apresentados nas descrições).
- Desenvolvam o máximo de requisitos do programa possível e informem suas considerações detalhadas para justificar as decisões de implementação tomadas.
- Implemente as instâncias de teste na sua classe principal (Main).
- Não há a necessidade de desenvolver um sistema de banco de dados.