

Programação Orientada a Objetos

Projeto Orientado a Objetos

Valor: 70 pts

Última atualização: 13/04/2023

1. Orientações Gerais

O trabalho deve ser realizado em grupos de até cinco alunos e deverá ser entregue no moodle de forma incremental a cada semana (sprint), sempre no domingo até às 23:59. É permitida a discussão de estratégias com os colegas, porém a modelagem e implementação somente deve ser feita pelos integrantes do grupo. Cópias de modelos e programas (ainda que parciais) não serão aceitas. Entregas com atraso poderão ser aceitas, porém serão penalizadas com perda de pontos pelo grupo. Deve ser postado APENAS um arquivo compactado por grupo contendo todos os entregáveis da semana. Apenas um aluno deve realizar a entrega informando o nome dos integrantes do grupo. Os grupos não devem ser alterados ao longo do semestre.

2. Objetivo

Neste trabalho prático desenvolveremos um modelo de classes e implementaremos um sistema de gestão para um Aeroporto. O projeto será desenvolvido em sprints semanais, que envolverão a modelagem das classes para atender ao escopo da sprint, e também a implementação dessas classes em PHP. O código-fonte das classes deverá ser testado através da chamada de métodos das mesmas.

A dinâmica de trabalho se dará da seguinte forma:

- A sprint se inicia em uma sexta-feira, na qual serão apresentados os requisitos a serem atendidos;
- A modelagem das classes em linguagem UML deverá ser entregue via moodle até o domingo subsequente em formato PDF;
- Na quinta-feira da semana seguinte, a implementação das classes será iniciada na aula na sala 312 do bloco 1 (CCE).
- O código-fonte gerado deverá ser enviado via moodle também até o domingo subsequente.

A pontuação se dará da seguinte forma:

- 6 pts para a entrega do modelo de classes a cada sprint, totalizando 42 pts ao final do semestre.
- O código-fonte, apesar de entregue a cada sprint, somente será avaliado e pontuado na entrega da versão final, totalizando 28 pts.

2.1. Sprint 1

Nesta sprint, vamos iniciar o desenvolvimento do sistema, que deverá atender aos seguintes requisitos:

- O sistema deve permitir o cadastro de companhias aéreas, contendo nome, código, razão social, CNPJ e sigla.
 - A sigla deve ser formada por duas letras.
- O sistema deve permitir o cadastro de aeronaves, contemplando os seguintes atributos:
 - Fabricante (Ex.: Boeing)
 - Modelo (Ex.: A-320)
 - Capacidade de passageiros
 - Capacidade de carga em kg
 - Registro da aeronave
 - Composto pelo prefixo, que contém duas letras
 - Um hífen
 - Seguido de três letras
 - (Ex.: PR-GUO)
 - No Brasil, somente são permitidos para voos comerciais os prefixos PT, PR, PP, PS, que devem ser validados.
- Toda aeronave pertence a uma companhia aérea.
- O sistema deve permitir o cadastro de voos, que consistem em um deslocamento entre dois aeroportos em um dado horário de partida, realizado por uma companhia aérea.
 - A frequência dos voos também deve ser modelada. Existem voos diários e voos em apenas alguns dias da semana.
 - Um voo costuma ser executado por uma mesma aeronave, que pode executar mais de um vôo.
 - A duração estimada do voo também deve ser armazenada.
- A execução de um voo, ou seja, sua ocorrência e os respectivos detalhes, também deve ser armazenada.
 - Por exemplo: Voo AD4127 da Azul, entre CNF e CGH, executado no dia 13/03/2023.
 - Os horários de partida e de chegada devem ser armazenados.
 - Ocasionalmente, uma aeronave diferente da aeronave cadastrada pode executar um voo.
- Os aeroportos também devem ser cadastrados no sistema com sua sigla, que possui três letras, e a cidade e estado do aeroporto.

2.2. Sprint 2

Nesta sprint devem ser modeladas e implementadas classes para atendimento dos seguintes requisitos:

- O sistema deve permitir a definição de um código para cada voo, caso ainda não tenha sido modelado, seguindo a seguinte regra de validação:
 - O código deve ser composto de duas letras seguidas de quatro números;
 - As duas letras iniciais devem coincidir com a sigla da companhia responsável por executar o voo.
- Em conversas com os usuários, convencionou-se que o termo *viagem* deve ser utilizado para representar uma execução de um voo em uma data específica. Esse termo passará a ser utilizado a partir desta sprint visando facilitar a comunicação entre desenvolvedores, PO e usuários.
- O sistema deve realizar a programação de viagens de voos ativos por um período de até 30 dias a partir da data atual, visando permitir a venda de passagens.
- O sistema deve permitir a compra antecipada de passagens por clientes, identificados pelo nome e sobrenome, além de um documento de identificação válido (RG ou Passaporte).
 - Toda passagem deve ter um aeroporto de origem e um aeroporto de destino;
 - O valor da tarifa deve ser também modelado, assim como a ocupação e disponibilidade de assentos em cada viagem;
 - Também deve ser permitida a escolha de um assento, opcional e sem custo para o passageiro;
 - O cliente pode optar por adquirir até no máximo 3 franquias de bagagens de até 23kg, cujo valor unitário deve ser definido por cada companhia aérea.
- Caso seja solicitada uma passagem entre dois aeroportos que não possuem voos diretos, o sistema deve compor a passagem utilizando dois. No momento vamos permitir apenas uma conexão.
 - Nesse caso as informações de ambas as viagens devem ser armazenadas;
 - O valor total da passagem deve ser a soma dos voos.

2.3. Sprint 3

Nesta sprint devem ser modeladas e implementadas classes para atendimento dos seguintes requisitos:

- O sistema deve permitir o cadastro de passageiros, que serão de dois tipos:
 - Passageiro Comum – que possui os campos abaixo e nenhum benefício especial:
 - Nome e sobrenome
 - Documento de identificação válido (RG ou Passaporte)
 - CPF (se brasileiro)
 - Nacionalidade
 - Data de Nascimento
 - E-mail
 - Passageiro Vip – que, além das informações do Passageiro Comum, possui os seguintes benefícios:
 - Alteração e cancelamento de voo sem custo
 - 1 franquia de passagem gratuita por viagem
 - Franquias adicionais com desconto de 50%
- Os campos CPF, e-mail e Data de Nascimento dos passageiros devem ser validados.
- Passageiros devem estar associados às suas passagens.
- Deve ser possível acessar o histórico de vôos de um passageiro em ordem cronológica.
- O sistema deve permitir o check-in de passagens já adquiridas em um período compreendido entre 48h e 30 minutos do horário de partida do primeiro voo. Em caso de não realização do check-in o sistema deve registrar o NO SHOW, que indica o não comparecimento do passageiro.
- Além do check-in o sistema deve permitir o registro de embarque (ou não) do passageiro. Caso este não embarque, também deve ser registrado NO SHOW.
- Todos os status do passageiro em um voo deve ser passível de registro. São eles:
 - Passagem adquirida
 - Passagem cancelada
 - Check-in realizado
 - Embarque realizado
 - NO SHOW

Bom trabalho!