

# Tarefa

## Objetivo

O objetivo desta tarefa é reforçar as suas habilidades de orientação a objetos em Objective-C.

## Descrição do Problema e Restrições

### Requisitos

Foi solicitada a implementação de uma aplicação em Objective-C para Controle de Reservas de Passagens Aéreas (CRPA). O CRPA permite que um usuário possa criar aeroportos, companhias aéreas e voos. Cada companhia aérea possui um conjunto de voos os quais ela opera. Um voo possui um local de aeroporto de partida e um aeroporto de destino. Não é possível ter o mesmo aeroporto como partida e destino. Cada voo possui um conjunto de categorias (como primeira classe, executiva ou classe econômica). Cada categoria possui assentos divididos em linhas e colunas. A aplicação possui um SystemManager, que é um ponto único de acesso para todas as funcionalidades providas.

Nesta tarefa, você deve desenvolver uma aplicação que forneça as seguintes funcionalidades:

1. **Criar um aeroporto.** Um aeroporto deve possuir uma sigla de três caracteres do alfabeto maiúsculos. Não é possível haver dois aeroportos com o mesmo nome.
2. **Criar uma companhia aérea.** Uma companhia aérea deve possuir um nome com, no máximo, 6 dígitos. Não é possível haver duas companhias aéreas com o mesmo nome.
3. **Criar um voo em uma companhia aérea, com um aeroporto de partida e um aeroporto de destino, um número de voo e uma data de chegada:** Um voo possui um identificador único, que é uma sequência de caracteres alfanuméricos.

4. **Criar uma categoria para um voo.** O número de colunas e linhas de assentos devem ser informados quando uma categoria for criada.

5. **Encontrar voos disponíveis.** Encontrar todos os voos partindo de um determinado aeroporto com destino a outro determinado aeroporto, nos quais haja assentos disponíveis para uma data escolhida.

6. **Reservar um assento.** Reservar um assento disponível em um voo específico, partindo de um aeroporto especificado e tendo outro aeroporto de destino escolhido, em uma data determinada.

7. **Imprimir detalhes do sistema.** Informar todos os valores dos atributos de todos os objetos (aeroportos, companhias, etc.) do sistema.

## Classes obrigatórias

Sua aplicação deve conter as seguintes classes:

### SystemManager

Essa classe disponibiliza uma interface para o sistema (fachada). Dito isso, os clientes interagem com a aplicação através de operações disponíveis nesta classe. O SystemManager está conectado com todos os aeroportos e objetos “aéreos” do sistema. Quando criado, o SystemManager não possui aeroportos, companhias ou voos. Para criá-los, os métodos createAirport() e createAirline() devem ser chamados. A classe também possui operações para criação de categorias de voos e voos, encontrar voos disponíveis e reservar assentos em algum voo. Um método de descrição imprime na tela informações sobre todos os aeroportos, companhias aéreas, voos e assentos, chamado displaySystemDetails().

**createAirport(String n):** Cria um objeto do tipo aeroporto e anexa-o no SystemManager. O aeroporto terá um nome; esse nome deve ter exatamente 3 caracteres. Dois aeroportos não podem possuir o mesmo nome.

**createAirline(String n) :** Cria um objeto do tipo companhia aérea com um nome e anexa-o no SystemManager. Uma companhia

aérea deve ter um nome que não exceda 5 caracteres. Duas companhias aéreas não podem possuir o mesmo nome.

**createFlight(String aname, String orig, String dest, int year, int month, int day, String id):** Cria um voo para uma companhia aérea (aname), a partir de um aeroporto (orig), com destino a outro aeroporto (dest), em um dado ano (year), mês (month) e dia (day). O voo tem um identificador (id).

**createSection(String air, String flID, int rows, int cols, SeatClass s):**

Cria uma categoria de voo para um voo (flID), associado a uma companhia aérea (air). A categoria recebe o número de linhas e colunas de assentos.

**findAvailableFlights(String orig, String dest):** Encontra todos os voos de um aeroporto (orig) até o outro (dest), com assentos disponíveis.

**bookSeat(String air, String fl, SeatClass s, int row, char col):** Reserva um assento em uma linha (row) e coluna (col) específicos, na categoria especificada (s), em um voo (fl) de uma companhia aérea (air).

**displaySystemDetails():** Mostra os valores dos atributos de todos os objetos do sistema.

## Airport

Objetos dessa classe representam aeroportos. A única informação armazenada é o nome, que deve ter exatamente 3 caracteres.

## Airline

Essa classe mantém informações de companhias aéreas. Uma companhia aérea pode ter zero ou mais voos associados com ela. Quando se cria uma companhia aérea, ela não possui voos. Todos os voos de uma mesma companhia aérea devem ter identificadores únicos.

## Flight

Essa classe mantém informações sobre voos. Um voo pode ter 0 ou mais categorias. Só pode haver categorias únicas em um voo (somente uma Primeira Classe, por exemplo). As categorias são definidas por uma classe enumeradora chamada “SeatClass” que possui os valores Primeira, Executiva e Econômica.

## Seat

Essa classe mantém informações sobre assentos. Especificamente, cada assento possui um identificador (número da linha + letra da coluna), e um status para indicar se ele está reservado ou não.

## Example Client Class

Um exemplo simples está implementado abaixo, mostrando operações do SystemManager.

@implementation ClientProg

-(void)startApp

```
{
    SystemManager *res = [[SystemManager alloc] init];

    //Create airports
    [res createAirport:@"DEN"];
    [res createAirport:@"DFW"];
    [res createAirport:@"LON"];
    [res createAirport:@"JPN"];
    [res createAirport:@"DE"]; //invalid
    [res createAirport:@"DEH"];
    [res createAirport:@"DEN"];
    [res createAirport:@"NCE"];
    [res createAirport:@"TRIord9"]; //invalid
    [res createAirport:@"DEN"];
    //Create airlines
    [res createAirline:@"DELTA"];
    [res createAirline:@"AMER"];
    [res createAirline:@"JET"];
    [res createAirline:@"DELTA"];
    [res createAirline:@"SWEST"];
    [res createAirline:@"AMER"];
    [res createAirline:@"FRONT"];
    [res createAirline:@"FRONTIER"]; //invalid
    //Create flights
    [res createFlight:@"DELTA" :@"DEN" :@"LON" :2009:10:10 :@"123"];
    [res createFlight:@"DELTA" :@"DEN" :@"DEH" :2009:8:8 :@"567"];
```

```
[res createFlight:@"DELTA" :@"DEN" :@"NCE" :2010 :9 :8 :@"567"];
//invalid
[res createFlight:@"JET" :@"LON" :@"DEN" :2009 :5 :7 :@"123"];
[res createFlight:@"AMER" :@"DEN" :@"LON" :2010 :10 :1 :@"123"];
[res createFlight:@"JET" :@"DEN" :@"LON" :2010 :6 :10 :@"786"];
[res createFlight:@"JET" :@"DEN" :@"LON" :2009 :1 :12 :@"909"];
//Create sections
[res createCategory:@"JET" :@"123" :2 :2 :economy];
[res createCategory:@"JET" :@"123" :1 :3 :economy];
[res createCategory:@"JET" :@"123" :2 :3 :first];
[res createCategory:@"DELTA" :@"123" :1 :1 :business];
[res createCategory:@"DELTA" :@"123" :1 :2 :economy];
[res createCategory:@"SWSERTT" :@"123" :5 :5 :economy];
//invalid
[res displaySystemDetails];
[res findAvailableFlights: @"DEN" :@"LON"];
[res bookSeat:@"DELTA" :@"123" :business :1 :'A'];
[res bookSeat:@"DELTA" :@"123" :economy :1 :'A'];
[res bookSeat:@"DELTA" :@"123" :economy :1 :'B'];
[res bookSeat:@"DELTA" :@"123" :business :1 :'A'];
//already booked
[res displaySystemDetails];
[res findAvailableFlights:@"DEN" :@"LON"];
```

}

@end