

## **Definição do projeto:**

O sistema de reserva de voos aéreos é uma aplicação projetada para facilitar a reserva de assentos em voos comerciais. Ele oferece aos usuários a capacidade de visualizar e reservar voos disponíveis, fornecendo uma experiência intuitiva e conveniente para os viajantes.

Tendo suas funcionalidades iniciais como a visualização de voos onde o sistema exibe uma lista de voos disponíveis com detalhes relevantes, como número do voo, destino, horário de partida, preço e assentos disponíveis, de reserva de assentos onde os usuários podem selecionar um voo e reservar assentos disponíveis para sua viagem. Outra funcionalidade que vale apenas citar é a de cancelamento de reserva, os usuários têm a opção de cancelar suas reservas e entre outras funcionalidades ou neste caso requisitos que viram a ser implementados, o desenvolvimento do sistema está a ser aplicado na linguagem de programação Java utilizando também para sua especificação formal o OpenJML.

# Documento de Requisitos: Sistema de reserva de voos junto de especificações formais:

## Requisitos Funcionais:

---

### [RF001] Visualizar Voos Disponíveis.

---

O sistema deve após sua execução deve exibir uma lista de voos disponíveis para o usuário, onde deve haver de forma visível as credenciais de cada voo, como o número de voos, o destino e a quantidade de assentos que estão disponíveis .

**PRIORIDADE:**      (X ) Essencial;              ( ) Importante;              ( ) Desejável.

```
Unset
/*
 * Requisito RF001 - Visualizar Voos Disponíveis
 *
 * displayFlights - visualizar voos
 * @ ensures (\forall int i; 0 <= i && i < flights.size();
 *      (\exists Flight f; flights.contains(f);
 *          f == flights.get(i) &&
 *          System.out.println((i + 1) + ". Voo " + f.getFlightNumber() + ":
 *Destino - " + f.getDestination() +
 *          ", Assentos disponíveis - " + f.availableSeats())));
 */
```

---

### [RF002] Selecionar Voo.

---

O sistema deve permitir que os usuários sejam capazes de selecionar um voo da lista dos voos que estão disponíveis.

**PRIORIDADE:**      (X) Essencial;      ( ) Importante;      ( ) Desejável.

```
Unset
/*
 * Requisito RF002 - Selecionar Voo
 *
 * getFlight - obter voo
 *
 * @ requires index >= 0 && index < flights.size();
 * @ ensures \result != null;
 * @ ensures \result.equals(flights.get(index));
```

---

### [RF003] Reservar Assentos.

---

O sistema deve permitir que os usuários possam reservar assentos para passageiros em um voo selecionado, onde deve haver uma disponibilidade de assentos para que seja possível reservá-lo, isso será possível a partir do momento em que ele seleciona o voo, e informa seu nome.

```
Unset
/*
 * Requisito RF003 - Reservar Assentos
 *
 * bookSeat - reservar assento
 *
 * @ requires isSeatAvailable();
 * @ requires passengers != null;
 * @ requires passengerName != null;
 * @ ensures passengers.contains(passengerName);
 * @ ensures \result ==> passengers.contains(passengerName);
 * @ ensures \result ==> passengers.size() == \old(passengers.size()) + 1;
 * @ ensures !\result ==> passengers.size() == \old(passengers.size());
 *     lableSeats() == \result[i].availableSeats());
 */
```

**PRIORIDADE:**      (X ) Essencial;              ( ) Importante;              ( ) Desejável.

---

#### [RF004] Cancelar Reserva.

---

O sistema deve permitir que o usuário a partir do momento que ele informa seu nome, ele tenha a opção de cancelar sua reserva.

**PRIORIDADE:**      (X ) Essencial;              ( ) Importante;              ( ) Desejável.

```
Unset
/*
 * Requisito RF004 - Cancelar Reserva
 *
 * cancelReservation - cancelar reserva
 *
 * @ requires flightIndex >= 0 && flightIndex < flights.size();
 * @ requires passengerName != null;
 * @ ensures (\forall int i; 0 <= i && i < \old(flights.size());
 * (\old(flights.get(i).getFlightNumber()).equals(flights.get(i).getFlightNumber()) &&
 * \old(flights.get(i).getDestination()).equals(flights.get(i).getDestination(
 * )) && \old(flights.get(i).getCapacity()) == flights.get(i).getCapacity() &&
 * (\old(flights.get(i).getPassengers()).contains(passengerName) ?
 * \old(flights.get(i).getPassengers()).size() ==
 * flights.get(i).getPassengers().size() + 1 :
 * \old(flights.get(i).getPassengers()).size() ==
 * flights.get(i).getPassengers().size())));
 */
```

---

#### [RF005] Visualizar detalhes do voo selecionado.

---

O sistema deve permitir que os usuários visualizem os detalhes de um voo específico, incluindo o número do voo, destino, assentos disponíveis e lista de passageiros. Isso fornece informações adicionais sobre o voo selecionado durante o processo de reserva, permitindo que os usuários façam escolhas informadas sobre a reserva de assentos.

**PRIORIDADE:**      (X ) Essencial;              ( ) Importante;              ( ) Desejável.

Unset

```
/*
 * Requisito RF005 - Visualizar detalhes de voo selecionado
 *
 * displayDetails - detalhes de exibição
 *
 * @ requires flights != null;
 * @ ensures (\forall int i; 0 <= i && i < flights.size();
 *           (\exists Flight flight; flights.contains(flight);
 *           flight.getFlightNumber().equals(\result[i].getFlightNumber()) &&
 *           flight.getDestination().equals(\result[i].getDestination()) &&
 *           flight.availableSeats() == \result[i].availableSeats()));
 */
```

Unset

## Requisitos não Funcionais:

---

### [RNF001] Ambiguidade de usuários.

---

O sistema não deve permitir que tenham usuários com o mesmo nome.

**PRIORIDADE:**      (X) Essencial;      ( ) Importante;      ( ) Desejável.

---

### [RNF002] Invalidade das opções.

---

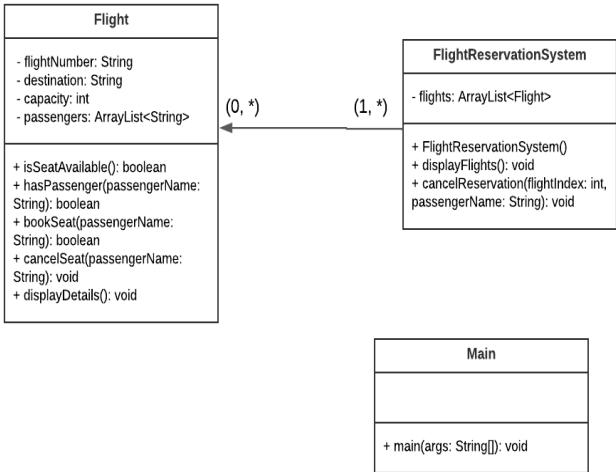
O sistema não deve aceitar opções que não estejam disponíveis para reserva.

**PRIORIDADE:**      (X) Essencial;      ( ) Importante;      ( ) Desejável.

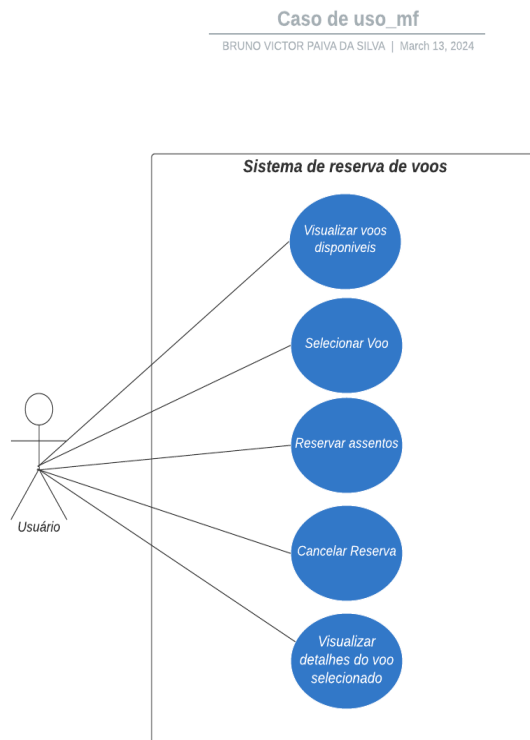
# Diagrama de classe:

Diagrama de classe

BRUNO VICTOR PAIVA DA SILVA | March 13, 2024



## Diagrama de caso de uso:



## Referências:

<https://www.openjml.org/documentation/>  
<https://github.com/OpenJML/OpenJML>  
<https://docs.oracle.com/en/java/>