



UNIVERSIDADE FEDERAL DO MARANHÃO  
CAMPUS SÃO LUÍS - CIDADE UNIVERSITÁ  
ENGENHARIA DA COMPUTAÇÃO

PROJETO E DESENVOLVIMENTO DE SOFTWARE

**SISTEMA DE COMPRA RÁPIDA DE PASSAGEM AÉREA INTEGRADO COM  
INTELIGÊNCIA ARTIFICIAL (PROJETO SKAI)**

PROFESSOR DR. THALES LEVI AZEVEDO VALENTE

KAUAN GUILHERME ALVES PINHEIRO SANTOS - 2021052316

LILIA ROSA COELHO MOURA - 2021024732

RAYLAN BRUNO SANTANA CARVALHO - 2021068826

VIRGINIA MARIA MONDEGO FERREIRA - 2021069968

YASMIN CANTANHEDE SANTOS - 2021066044

SÃO LUÍS - MA

JUNHO DE 2025

## **1. INTRODUÇÃO**

O presente Documento de Requisitos de Software (DRS) descreve de forma detalhada as funcionalidades e restrições do sistema de compra rápida de passagens aéreas com integração de Inteligência Artificial (apelidado de SkAI). Este sistema será uma aplicação web que visa agilizar o processo de busca e aquisição de passagens por meio de interfaces tradicionais e conversacionais. A documentação tem como objetivo alinhar as expectativas entre os envolvidos no projeto, incluindo desenvolvedores, stakeholders e usuários finais, estabelecendo uma base sólida para o desenvolvimento, testes e manutenção do sistema.

### **1.1 ESCOPO DO PROJETO**

Muitos usuários relatam dificuldades com plataformas de viagem tradicionais por serem confusas, poluídas visualmente e pouco intuitivas. Além disso, o processo de buscar, comparar e escolher passagens pode ser cansativo, exigindo que o usuário acesse vários sites e faça análises manuais. Nesse caso, o público-alvo são: pessoas que desejam comprar passagens aéreas de forma rápida, sem complicações, incluindo viajantes ocasionais e frequentes, que valorizam uma boa experiência de uso.

### **1.2 FINALIDADE**

A finalidade deste documento é definir de maneira clara e objetiva os requisitos funcionais e não funcionais do sistema de compra rápida de passagens aéreas. Ele servirá como guia para todas as fases do desenvolvimento do projeto, garantindo que o sistema atenda às necessidades dos usuários e funcione de forma eficiente, segura e intuitiva. Além disso, este documento contribuirá para a rastreabilidade dos requisitos ao longo do ciclo de vida do software.

## **2. DESCRIÇÃO GERAL**

O objetivo do sistema é proporcionar aos usuários uma plataforma prática, intuitiva e inteligente para a compra de passagens aéreas, eliminando a complexidade e o cansaço geralmente associados ao planejamento de viagens. Por meio de uma interface amigável e recursos como busca personalizada, recomendações com base em preferências, e suporte via chatbot, o sistema busca tornar a experiência de encontrar e adquirir passagens mais rápida, eficiente e agradável, atendendo especialmente aqueles que se sentem desconfortáveis com plataformas confusas e mal organizadas.

### **2.1 PERSPECTIVA DO PRODUTO**

O sistema de compra rápida de passagens aéreas é um software independente, porém com integração a serviços externos, como APIs de companhias aéreas e plataformas de pagamento para simulação de compra. Ele se encaixa no modelo de arquitetura cliente-

servidor, onde a aplicação web (cliente) se comunica com o backend e com serviços de terceiros para buscar dados de voos, processar pagamentos e responder a comandos do chatbot.

O sistema se destina a usuários finais que desejam buscar, comparar e adquirir passagens aéreas com rapidez, sem a necessidade de utilizar múltiplos sites ou plataformas. Diferencia-se de sistemas tradicionais por incorporar um assistente virtual baseado em Inteligência Artificial, que torna a interação mais fluida por meio de linguagem natural.

A aplicação será projetada com foco em escalabilidade, segurança e boa experiência do usuário. A interface gráfica será desenvolvida com atenção à acessibilidade, responsividade e simplicidade, permitindo que usuários de diferentes perfis consigam utilizar a ferramenta sem dificuldade.

## **2.2 FUNCIONALIDADES DO PRODUTO**

- Cadastro e login de usuários, convencional e pelo google.
- Busca de passagens aéreas via formulário.
- Interação com chatbot para busca de passagens personalizada.
- Simulação de compra de passagens pelo cartão e pix.
- Acesso ao histórico de compras.

## **2.3 RESTRIÇÕES**

- A aplicação será inicialmente desenvolvida para desktop (web responsivo).
- O armazenamento será feito em um banco de dados não relacional.
- O chatbot utilizará uma API de processamento de linguagem natural (da OpenAI ou google).
- A simulação de compra será feita por uma API de pagamentos.

## **3. REQUISITOS**

Requisitos são descrições das funcionalidades, comportamentos e restrições que um sistema deve atender para satisfazer as necessidades dos usuários e dos stakeholders. Eles servem como base para o desenvolvimento, testes e validação do software.

Nas tabelas a seguir, cada requisito foi associado a uma prioridade. A atribuição de níveis de prioridade aos requisitos — como alta, média e baixa — é fundamental para a gestão eficiente do desenvolvimento do sistema. Essa classificação permite que a equipe concentre seus esforços inicialmente nas funcionalidades essenciais (prioridade alta), que têm impacto direto na experiência do usuário ou são indispensáveis para o funcionamento mínimo do sistema.

Os requisitos de prioridade média geralmente envolvem funcionalidades importantes, mas que podem ser implementadas em uma segunda etapa, após o núcleo do sistema estar estável. Já os de baixa prioridade incluem melhorias ou recursos complementares, que podem ser planejados para versões futuras, conforme o tempo, orçamento e feedback dos usuários.

Essa organização facilita o planejamento de entregas (releases), gerenciamento de riscos e o uso mais eficiente dos recursos da equipe, além de permitir que um produto viável (MVP) seja entregue mais rapidamente, validando a solução junto ao público antes de evoluções mais complexas.

### 3.1 REQUISITOS FUNCIONAIS (RF)

Tabela 1 – Requisitos Funcionais

Código	Descrição	Prioridade
RF01	O sistema deve permitir pesquisa de voos por origem, destino, datas e filtros.	Alta
RF02	O sistema deve oferecer recomendações personalizadas de voos e destinos com base no histórico.	Baixa
RF03	O sistema deve permitir a compra rápida com preenchimento inteligente de dados.	Média
RF04	O sistema deve disponibilizar chat inteligente para dúvidas e sugestões com atendimento automatizado.	Alta
RF05	O sistema deve ser integrado com apis de companhias aéreas para busca de voos e políticas de alteração/cancelamento.	Alta
RF06	O sistema deve permitir realizar pagamentos com cartão de crédito/débito, pix (simulação com api).	Alta
RF07	O sistema deve exigir idade mínima de 18 anos para compra.	Baixa
RF08	O sistema deve suportar modo noturno na interface.	Média
RF09	O sistema deve exibir sugestões baseadas em clima e feriados.	Alta
RF10	O sistema deve exibir histórico de passagens/compras.	Alta
RF11	O sistema deve permitir autenticação do usuário via google.	Alta
RF12	O sistema deve descobrir as preferências do usuário para recomendação personalizada.	Alta

Fonte: Autoria própria, 2025

### 3.2 REQUISITOS NÃO FUNCIONAIS (RNF)

Tabela 2 – Requisitos não Funcionais

Código	Descrição	Prioridade
RNF01	Interface web e responsiva, compatível com navegadores e dispositivos móveis.	Alta
RNF03	Interface intuitiva, limpa e moderna.	Alta
RNF04	Sistema deve usar criptografia e tokenização para segurança de dados sensíveis.	Alta

Fonte: Autoria própria, 2025

## 4. DIAGRAMAS

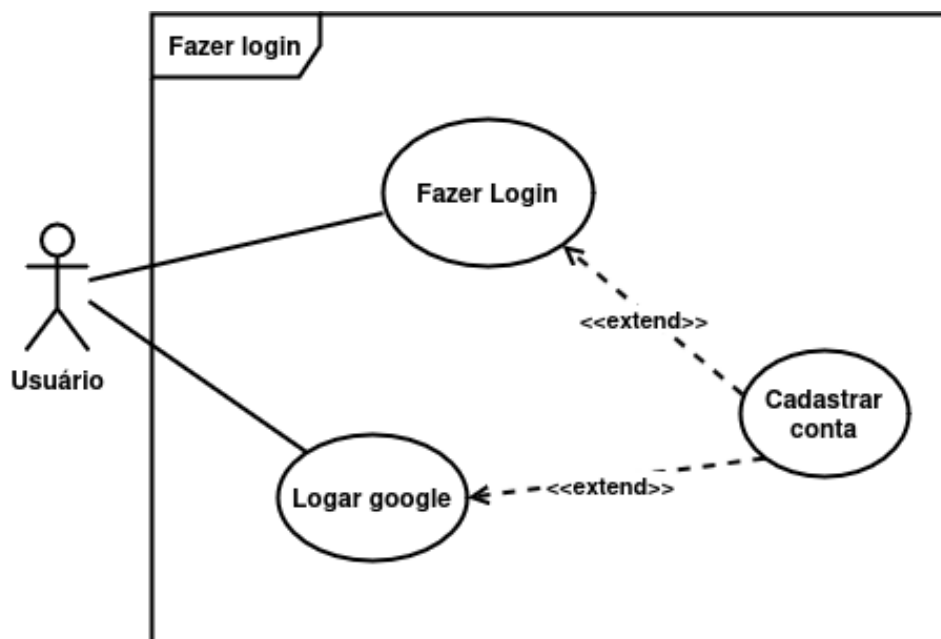
Um diagrama na UML (Unified Modeling Language) é uma representação visual que descreve aspectos estruturais ou comportamentais de um sistema de software. A UML é uma linguagem padronizada para modelagem, e seus diagramas ajudam a comunicar, entender e

documentar a arquitetura, o funcionamento e os relacionamentos entre os componentes do sistema. Esses diagramas são essenciais durante as fases de análise, design e documentação, pois facilitam o entendimento entre desenvolvedores, analistas e clientes, além de servirem como base para a implementação do software.

#### 4.1 DIAGRAMA DE CASOS DE USO

Esse diagrama mostra as funcionalidades principais do sistema do ponto de vista do usuário. Representa o que o sistema deve fazer, incluindo os atores (usuários ou sistemas externos) e os casos de uso (ações que eles podem realizar).

Imagem 1 – Diagrama de casos de uso de Fazer Login



Fonte: Autoria própria, 2025.

Descrição:

O caso de uso "Fazer Login" permite que o usuário acesse o sistema por meio de três opções: inserindo manualmente suas credenciais (e-mail e senha), optando por realizar um novo cadastro, ou utilizando a autenticação via conta Google (funcionalidade especial). Ao efetuar o login com sucesso, o sistema valida as informações e concede acesso às funcionalidades personalizadas da plataforma.

Atores: Usuário, Sistema de Autenticação Google (ator externo)

Fluxo Principal:

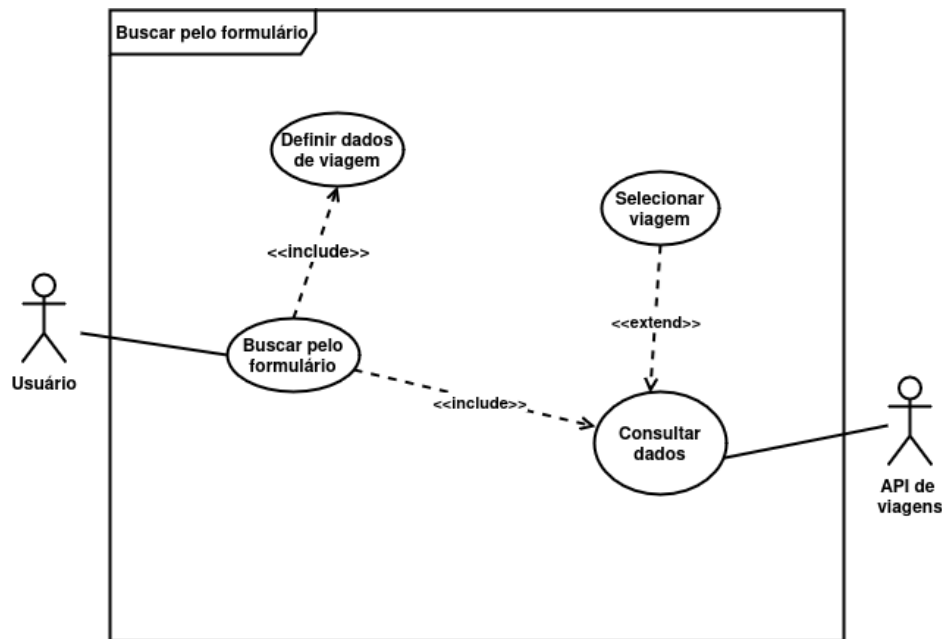
1. O usuário acessa a tela de login.
2. O sistema oferece três opções:
  - a. Inserir e-mail e senha (login tradicional)
  - b. Acessar tela de cadastro para criar conta
  - c. Fazer login com Google

3. O usuário escolhe uma das opções.
4. O sistema valida os dados ou redireciona para o serviço de autenticação Google.
5. Se os dados forem válidos, o usuário é autenticado e redirecionado à tela inicial.

Fluxos Alternativos:

- a. Se o login tradicional falhar, o sistema exibe uma mensagem de erro.
- b. Se for um novo usuário, ele pode ser redirecionado para a tela de cadastro.
- c. Se o login via Google for cancelado ou falhar, o sistema retorna à tela de login.

Imagem 2 – Diagrama de casos de uso de Buscar pelo Formulário.



Fonte: Autoria própria, 2025.

Descrição:

O caso de uso "Buscar pelo Formulário" permite que o usuário informe os dados da viagem desejada (como origem, destino, datas, número de passageiros e filtros opcionais) por meio de um formulário. Após o preenchimento e envio, o sistema valida os dados fornecidos, consulta uma API externa de companhias aéreas para buscar as opções disponíveis e exibe os resultados. O usuário pode, então, visualizar as opções e selecionar uma viagem para dar continuidade ao processo de compra.

Atores: Usuário, Sistema, API Externa de Companhias Aéreas (ator secundário)

Fluxo Principal:

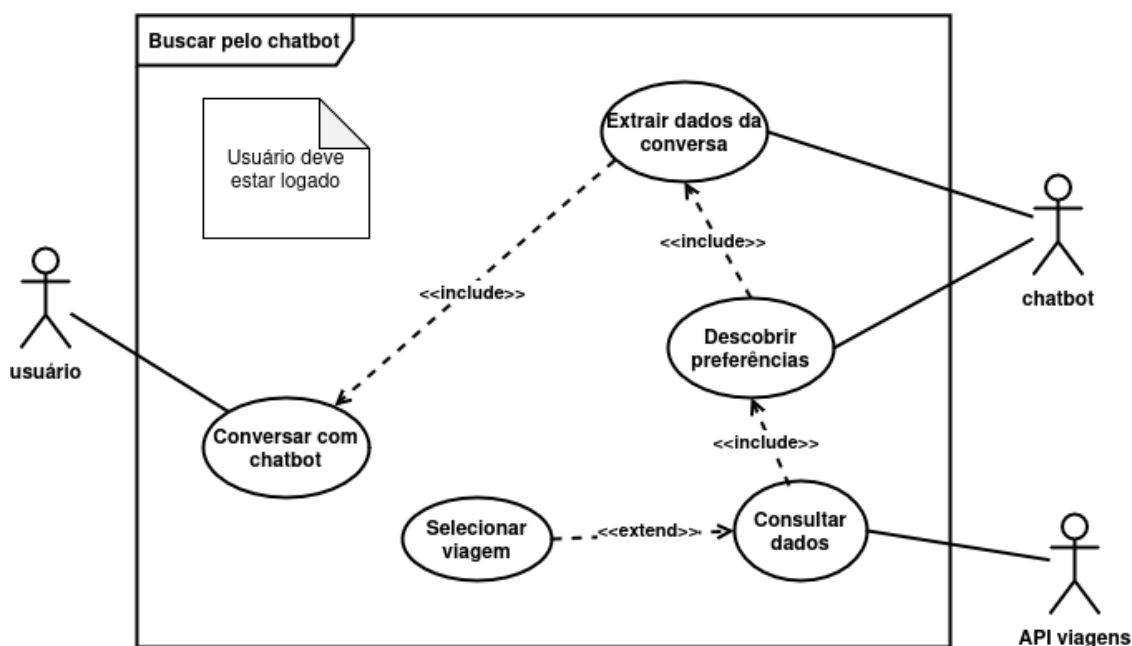
1. O usuário acessa o formulário de busca.
2. O usuário preenche os dados de viagem: origem, destino, datas.
3. O sistema valida os dados inseridos (ex: datas válidas, campos obrigatórios).
4. O sistema envia uma solicitação para a API externa com os parâmetros informados.
5. A API retorna as opções de voos disponíveis.
6. O sistema exibe os resultados ao usuário.

7. O usuário pode visualizar detalhes e selecionar uma opção de voo.

Fluxos Alternativos:

- Se os dados forem inválidos, o sistema exibe mensagens de erro e solicita correção.
- Se a API externa não retornar resultados (por erro ou falta de opções), o sistema informa ao usuário que nenhuma viagem foi encontrada.

Imagem 3 – Diagrama de casos de uso de Buscar pelo Chatbot.



Fonte: Autoria própria, 2025.

Descrição:

O caso de uso "Buscar pelo Chatbot" permite que o usuário interaja com um assistente virtual (chatbot) para encontrar passagens aéreas. Durante a conversa, o chatbot interpreta as mensagens do usuário, extrai os dados relevantes (como origem, destino, datas, orçamento e preferências), e complementa essas informações com o histórico do usuário, se disponível. Em seguida, o sistema consulta uma API externa de companhias aéreas para obter opções de viagens compatíveis. Os resultados são apresentados no próprio chat, permitindo que o usuário visualize e selecione uma viagem diretamente pela conversa.

Atores: Usuário, Chatbot, API Externa de Companhias Aéreas

Fluxo Principal:

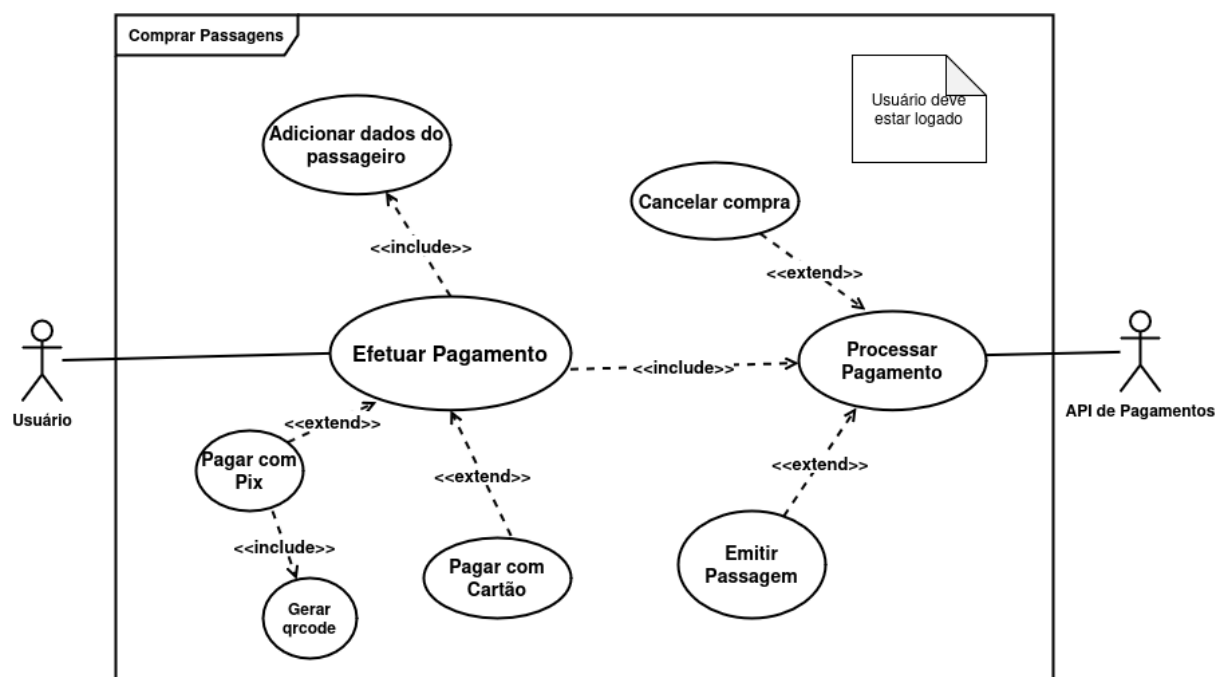
- O usuário inicia uma conversa com o chatbot.
- O chatbot faz perguntas ou interpreta mensagens livres para identificar os dados da viagem (ex: "Quero ir para o Rio de Janeiro no próximo feriado").
- O sistema analisa a conversa e extrai os dados necessários.
- O sistema complementa as informações com preferências do usuário (tipo de viagem, orçamento).
- O sistema envia a consulta para a API externa de voos.

6. A API retorna uma lista de viagens disponíveis.
7. O chatbot apresenta as opções de forma clara e interativa no chat.
8. O usuário pode solicitar detalhes ou selecionar uma das opções.

Fluxos Alternativos:

- a. Se a mensagem for ambígua ou incompleta, o chatbot solicitará mais informações.
- b. Se não houver resultados da API, o sistema mostrará um erro na procura de viagens.
- c. O usuário pode encerrar a conversa a qualquer momento.

Imagem 4 – Diagrama de casos de uso de Comprar Passagem.



Fonte: Autoria própria, 2025.

Descrição:

O caso de uso "Comprar Passagem" permite que um usuário autenticado realize a compra de uma passagem aérea após selecionar um voo. Durante o processo, o usuário deve informar os dados do passageiro, escolher a forma de pagamento (Pix ou cartão de crédito/débito), ou optar por cancelar a compra. O sistema deve validar as informações fornecidas, processar o pagamento por meio de uma API de pagamentos e, em caso de sucesso, emitir a confirmação da compra. Quando a opção for Pix, o sistema deve gerar um QR Code para facilitar o pagamento.

Atores: Usuário (logado), Sistema, API de Pagamentos

Pré-condição:

- a. O usuário deve estar autenticado no sistema.
- b. Deve haver um voo selecionado para compra.

Fluxo Principal:



1. O usuário acessa a tela de finalização de compra.
2. Informa os dados do(s) passageiro(s): nome, documento, data de nascimento etc.
3. Escolhe a forma de pagamento:
  - a. Pix: o sistema gera um QR Code.
  - b. Cartão: o sistema solicita os dados do cartão.
4. O sistema envia os dados para a API de pagamento.
5. A API responde com sucesso ou falha.
6. Se o pagamento for aprovado, o sistema emite a confirmação da compra e disponibiliza os dados do bilhete.
7. O usuário visualiza o comprovante e os detalhes da passagem.

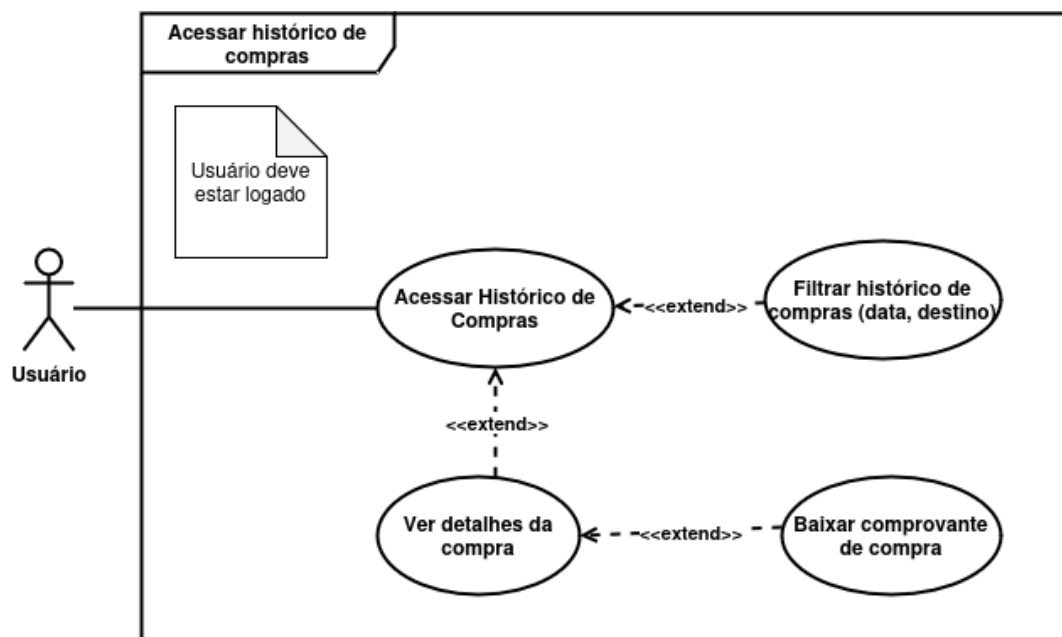
#### Fluxos Alternativos:

- a. O usuário pode cancelar a operação a qualquer momento antes do pagamento.
- b. Se a API de pagamento retornar falha (ex: cartão recusado, QR Code não pago no tempo limite), o sistema exibe uma mensagem de erro e permite tentar novamente.
- c. Caso o pagamento via Pix não seja confirmado, a compra não será finalizada.

#### Pós-condição:

- Se bem-sucedida, a compra é registrada no sistema e vinculada ao histórico do usuário.

Imagem 5 – Diagrama de casos de uso de Acessar Histórico de compras.



Fonte: Autoria própria, 2025.

#### Descrição:

O caso de uso "Acessar Histórico de Compras" permite que o usuário, após estar autenticado no sistema, visualize todas as suas compras de passagens realizadas. O usuário

pode filtrar as compras por critérios como data, destino ou status, visualizar os detalhes de cada compra e baixar o comprovante correspondente para fins de registro ou apresentação.

Atores: Usuário (logado), Sistema.

Pré-condição:

- O usuário deve estar autenticado no sistema.

Fluxo Principal:

1. O usuário acessa a seção de histórico de compras.
2. O sistema exibe a lista de compras realizadas pelo usuário.
3. O usuário pode aplicar filtros para facilitar a busca (ex: período, destino, status).
4. O usuário seleciona uma compra para ver seus detalhes completos.
5. O usuário pode optar por baixar o comprovante de compra em formato PDF ou similar.

Fluxos Alternativos:

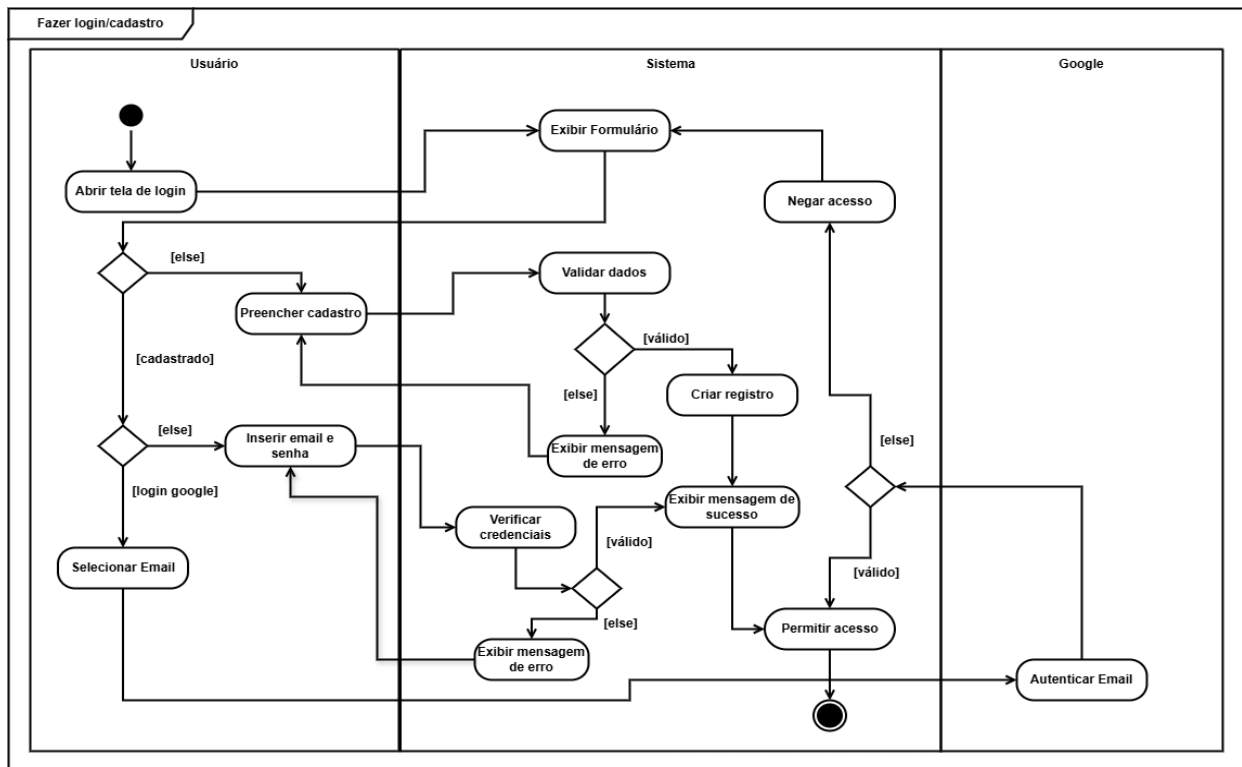
- a. Se não houver compras registradas, o sistema informa que o histórico está vazio.
- b. Caso haja erro na geração do comprovante, o sistema apresenta uma mensagem de falha e sugere tentar novamente

## **4.2 DIAGRAMA DE ATIVIDADES**

Esse diagrama representa o fluxo de trabalho ou processos dentro do sistema. Mostra como uma atividade começa, se ramifica, se repete e termina. É útil para descrever lógicas de negócio e processos com decisões.

O primeiro diagrama de atividades é o de login/cadastro, o qual descreve o fluxo de ações que um usuário realiza para acessar ou criar uma conta no sistema. O processo inicia, segundo a imagem 06 abaixo, com a abertura da interface de login, onde o usuário escolhe entre realizar o login ou efetuar um cadastro.

Imagem 6: Diagrama de Atividades de Login/Cadastro

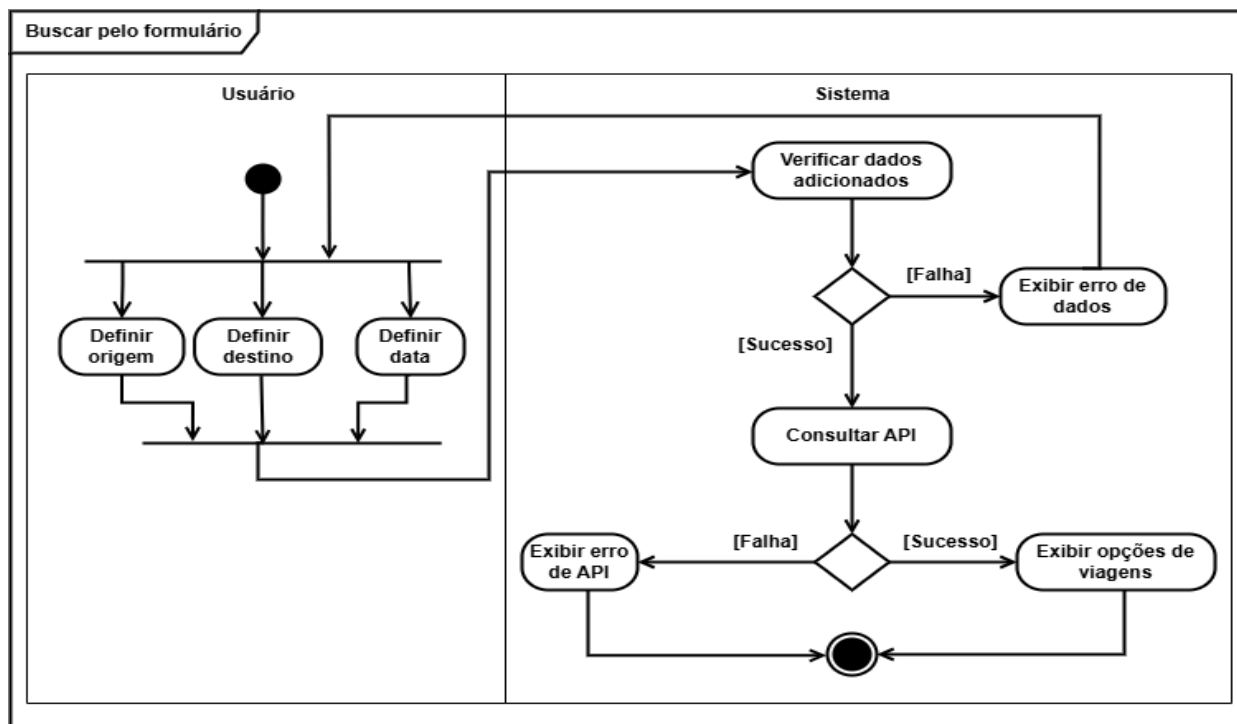


Fonte: Autoria própria, 2025

Se o usuário optar pelo login, o sistema solicita as credenciais (usuário e senha) e verifica-as, caso esteja validado, recebe mensagem de sucesso, permitindo o acesso como atividade final. Outro ponto de decisão é selecionar login com o google, tendo seu e-mail autenticado, permite acesso, caso não, ele retorna ao formulário. Se optar por preencher o cadastro, não possuindo nenhum, leva o sistema a validar os dados e, se válidos, cria um novo registro e exibe uma mensagem de sucesso. Caso contrário, uma mensagem de erro é exibida e o usuário retorna para preencher o cadastro novamente.

O diagrama abaixo, imagem 07, ilustra o processo de “Buscar pelo formulário”. O usuário inicia definindo a origem, destino e a data da viagem, O sistema verifica os dados adicionados, e se houver falha na verificação, um erro de dados é exibido, caso contrário, o sistema consulta uma API externa. Em caso de falha na consulta da API, um erro é exibido. No entanto, se a consulta for completada, é mostrado as opções de viagens, finalizando assim a busca.

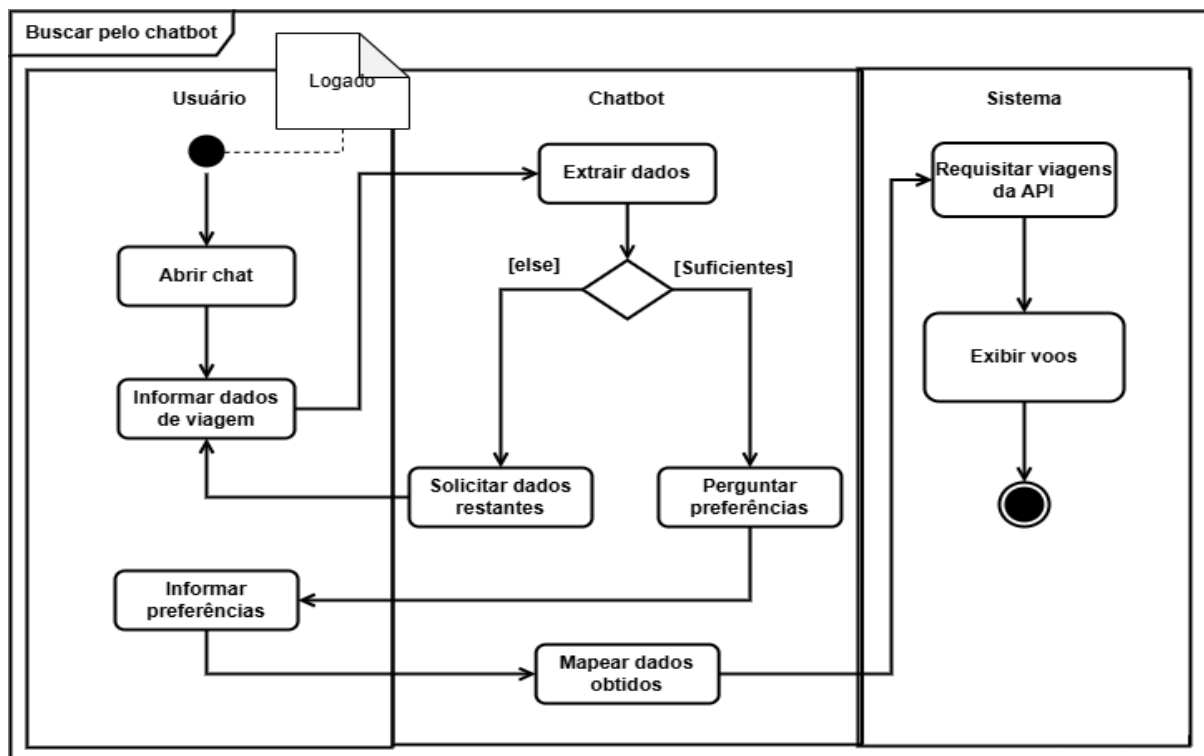
Imagem 7: Diagrama de Atividades de Buscar pelo formulário



Fonte: Autoria própria, 2025

A interação para “Buscar pelo chatbot” é descrita segundo a imagem 08. O usuário logado inicia abrindo o chat. Em seguida, informa os dados da viagem ao chatbot, que tenta extrair as informações. Se os dados não forem suficientes, o chatbot solicita os dados restantes ao usuário. Se forem suficientes, ele pergunta as preferências. Após coletar todas as informações, o chatbot mapeia os dados obtidos e o sistema requisita as viagens da API para então exibir os voos disponíveis ao usuário, encerrando as atividades da interação.

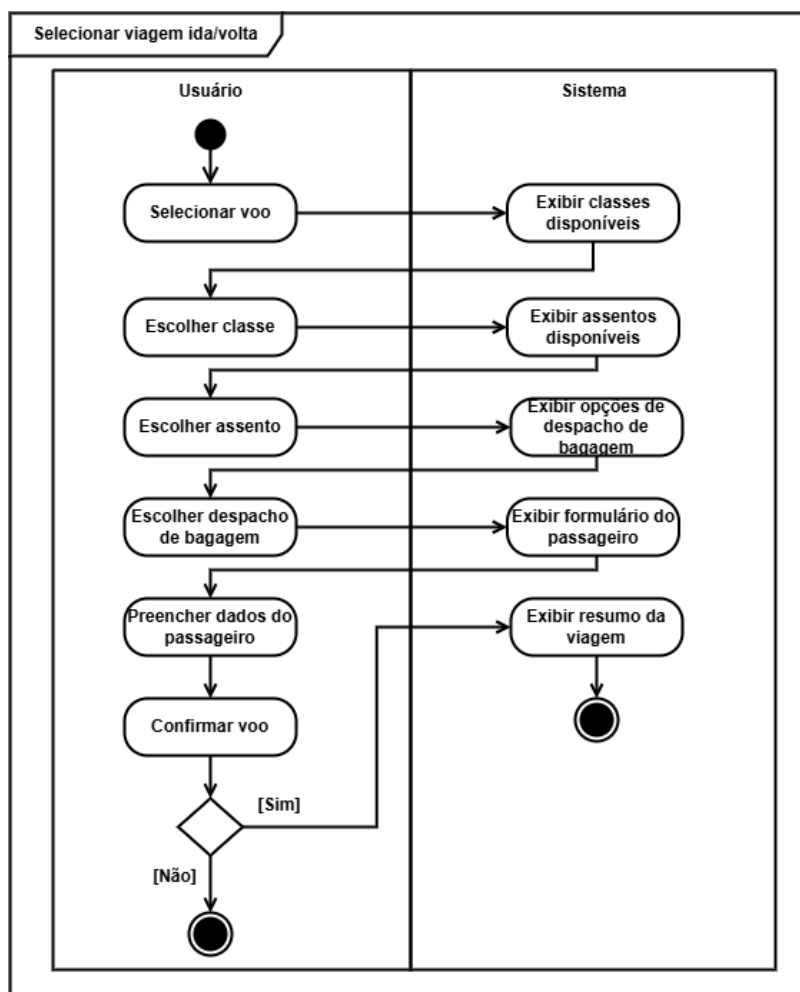
Imagem 8: Diagrama de Atividades de Buscar pelo chatbot



Fonte: Autoria própria, 2025

Para “Selecionar Viagem ida/volta”, o usuário escolhe o voo, o sistema responde exibindo as classes disponíveis e o usuário seleciona uma classe. Essa atividade leva à exibição dos assentos livres, e, após a escolha, são apresentadas as opções de despacho de bagagem. Uma vez selecionada a opção de bagagem, o sistema exibe o formulário do passageiro para preenchimento. Com os dados preenchidos, o usuário decide se confirma ou não o voo. Com a tomada de decisão, o usuário finaliza com “não” ou o sistema mostra um resumo da viagem caso “sim”, finalizando o fluxo.

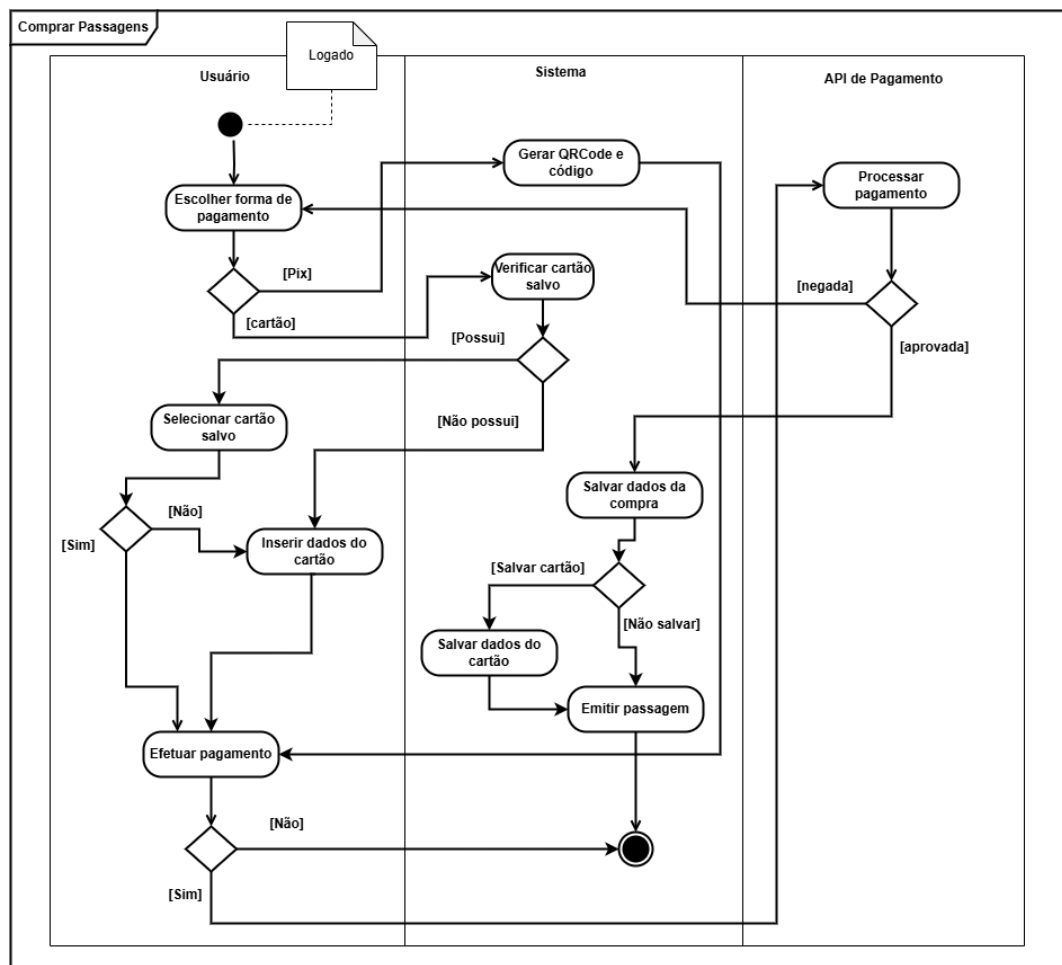
Imagem 9: Diagrama de Atividades de Selecionar Viagem



Fonte: Autoria própria, 2025

O processo para “Compra de passagem” começa com o usuário logado escolhendo a forma de pagamento, de acordo com a imagem 10, optando por pix ou cartão. Se for pix, o sistema gera um QR code, pede tomada de decisão para efetuar pagamento, processa e finaliza caso válido, ou retorna ao início. Se for cartão, o sistema verifica se há um cartão salvo, se houver, pergunta se deseja usar e faz o ciclo de pagamento, se não houver, o usuário insere novos dados. Quando o pagamento é aprovado, o sistema salva os dados da compra e pergunta se o usuário deseja salvar os dados do cartão, se sim ou se não, o sistema finaliza a compra.

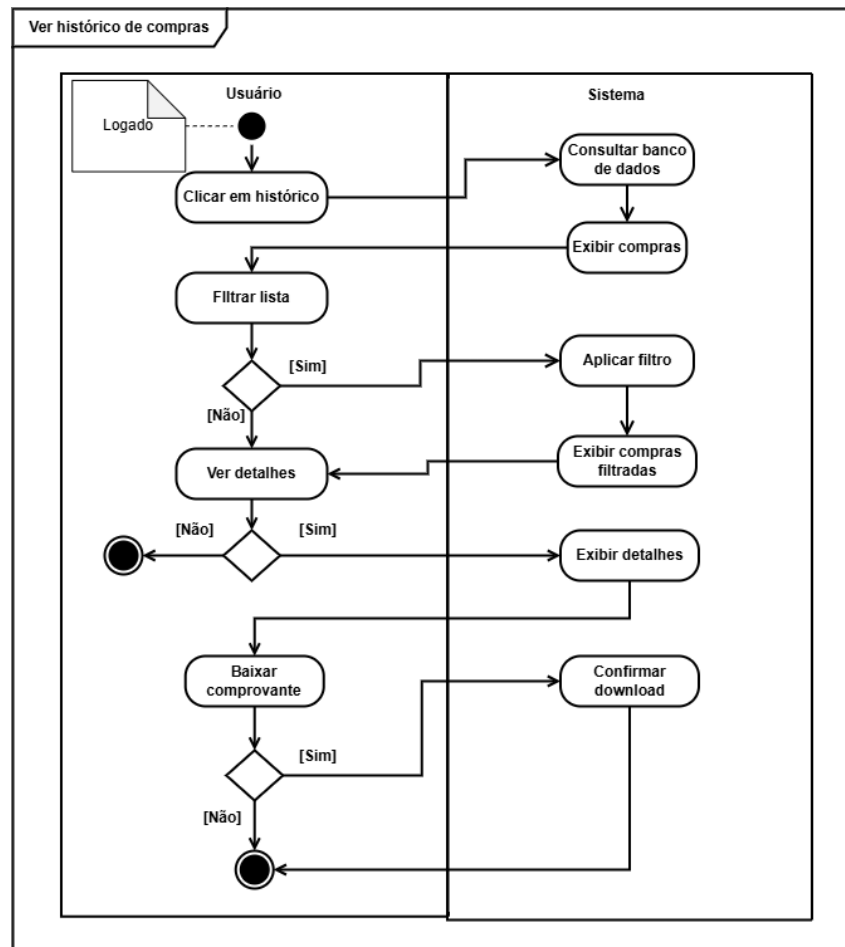
Imagem 10: Diagrama de Atividades de Compra de Passagens



Fonte: Autoria própria, 2025

Segundo a imagem 11 anexada abaixo, o diagrama de atividades de histórico inicia com o usuário, logado, clicando na opção de ver o histórico. O sistema consulta o banco de dados e exibe as compras realizadas. O usuário possui a opção de filtrar a lista, levando o sistema a aplicar e exibir as compras. Em seguida, pode-se escolher ver os detalhes de uma compra específica, resultando na exibição dos detalhes pelo sistema, com a decisão de baixar o comprovante, o sistema confirma o download caso seja afirmado e conclui a atividade, caso contrário, só finaliza diretamente.

Imagem 11: Diagrama de Atividades Histórico de Compras



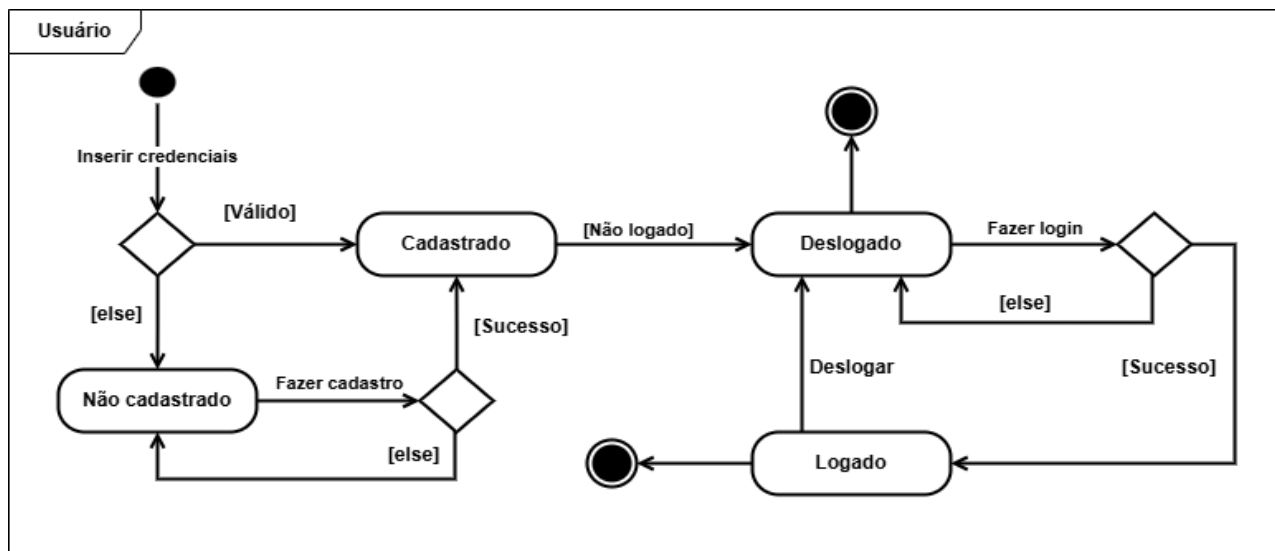
Fonte: Autoria própria, 2025

### 4.3 DIAGRAMA DE ESTADOS

Esse diagrama mostra os estados possíveis de um objeto ao longo do tempo e as transições entre esses estados. É útil para modelar o comportamento de elementos que mudam de estado com base em eventos.

Imagem 12: Diagrama de Estados de Usuário

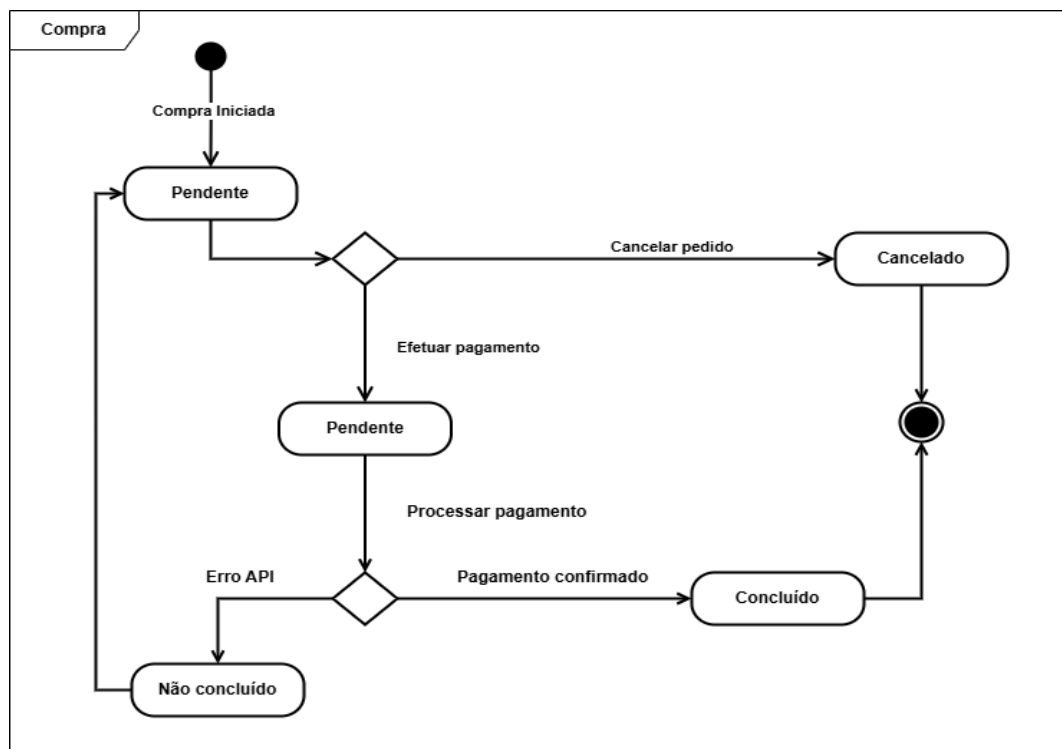




Fonte: Autoria própria, 2025

Neste diagrama, conforme imagem 12, ilustra o ciclo de vida do objeto “Usuário” dentro de um sistema, com foco na autenticação e no registro. Seu processo começa com “Inserir credenciais” seguido de um ponto de decisão: se são válidas ou não. Se for válida, o objeto “Usuário” entra no estado de “Cadastrado” e depois passa para o estado “Deslogado” para realizar o login. Caso o login seja bem-sucedido, o usuário vai para o estado “Logado” e seu estado finaliza ou volta para “Deslogado” se não for. Seu estado finaliza em “Deslogado” se o usuário não optar por fazer login novamente. Se as credenciais iniciais forem inválidas, seu estado transita para “Não cadastrado”. Desse último estado, uma ação “Fazer cadastro” pode levar ao estado “Cadastrado” após ter sucesso, ou permanecer em “Não Cadastrado” se falhar.

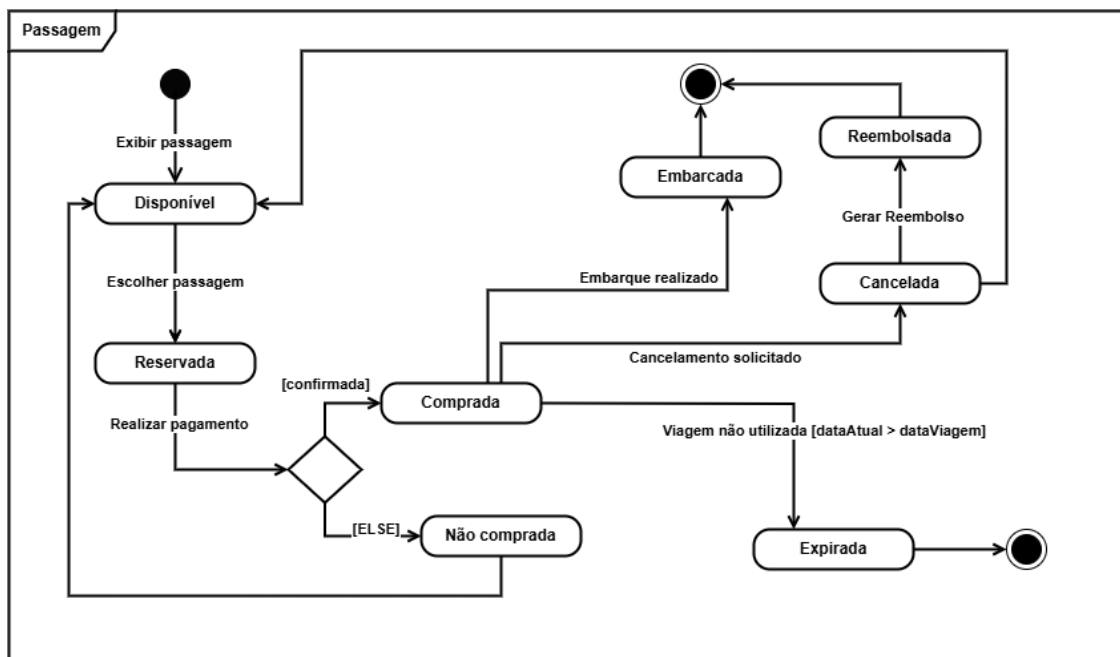
Imagem 13: Diagrama de Estados de Compra



Fonte: Autoria própria, 2025

Na imagem 13, modela-se o processo do objeto “Compra”, transitando do pseudo-estado inicial para “Pendente” com a ação de “Compra Iniciada”. Do estado “Pendente”, o usuário pode optar por “Cancelar pedido”, movendo a compra para um estado final “Cancelado”. Alternativamente, o usuário pode “Efetuar pagamento”, o que mantém a compra em um estado “Pendente” enquanto o pagamento está sendo processado. A ação “Processar pagamento” leva a um ponto de decisão: se “Pagamento confirmado”, a compra atinge o estado final “Concluído”. Se ocorrer um “Erro de API” durante o processamento, a compra retorna ao estado “Não concluído”, de onde pode potencialmente retornar a “Pendente” para tentar o pagamento novamente.

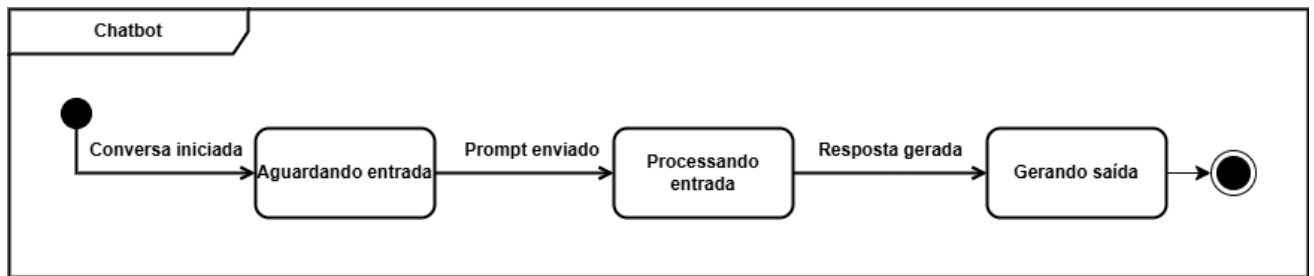
Imagem 14: Diagrama de Estados de Passagem



Fonte: Autoria própria, 2025

O ciclo de vida do objeto “Passagem” começa em um estado inicial, de acordo com a imagem 14, passando para “Disponível” onde pode ser visualizada ou selecionada. Uma vez selecionada, ela transita para “Reservada”. Segue-se uma tentativa de pagamento, que leva ao estado “Comprada” se for confirmada, ou a “Não comprada” caso contrário, retornando ao estado “Disponível”. Uma passagem comprada pode ser “Embarcada”, após o evento de embarque bem-sucedido, ou pode ser “Cancelada” se um cancelamento for solicitado. Uma passagem cancelada pode então ser “Reembolsada” e finalizar. Além disso, se uma passagem comprada não for utilizada até a data da viagem, ela transita para o estado final “Expirada”.

Imagem 15: Diagrama de Estado de Chatbot



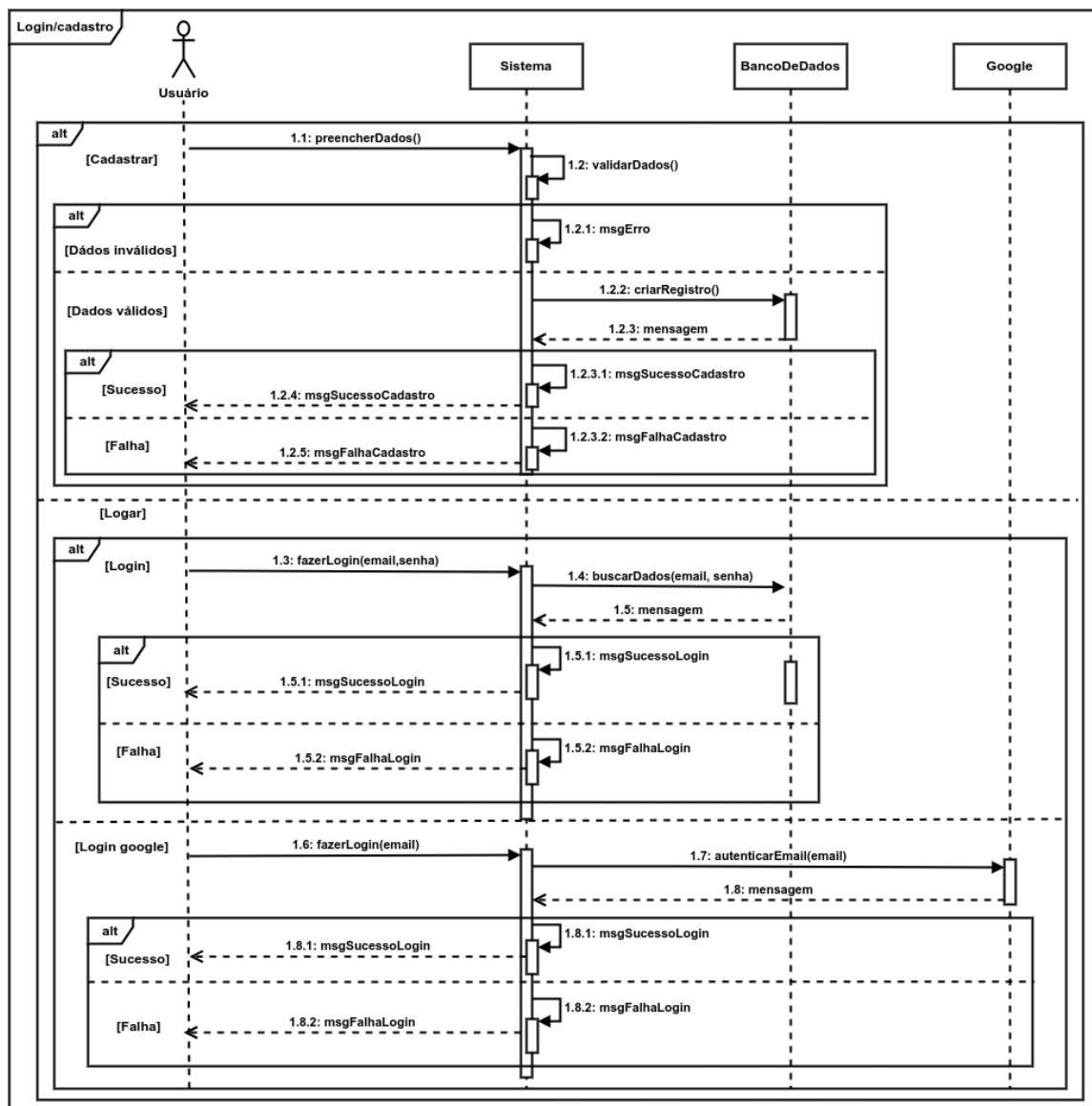
Fonte: Autoria própria, 2025

Demonstra os estados operacionais de chatbot, conforme imagem 15, iniciando com o evento “Conversa Iniciada”. O chatbot então entra no estado “Aguardando entrada”. Após esse estado, transita com o evento “Prompt enviado” para “Processando entrada”. Assim qu a entrada é processada, o estado do chatbot se move com “Resposta gerada” para “Gerando saída”, no qual seu ciclo de interação atual é concluído.

#### 4.4 DIAGRAMA DE SEQUÊNCIA

Esse diagrama mostra a interação entre objetos ou componentes em ordem cronológica. Indica quem envia mensagens para quem, em qual ordem, e como os elementos colaboram para realizar uma função.

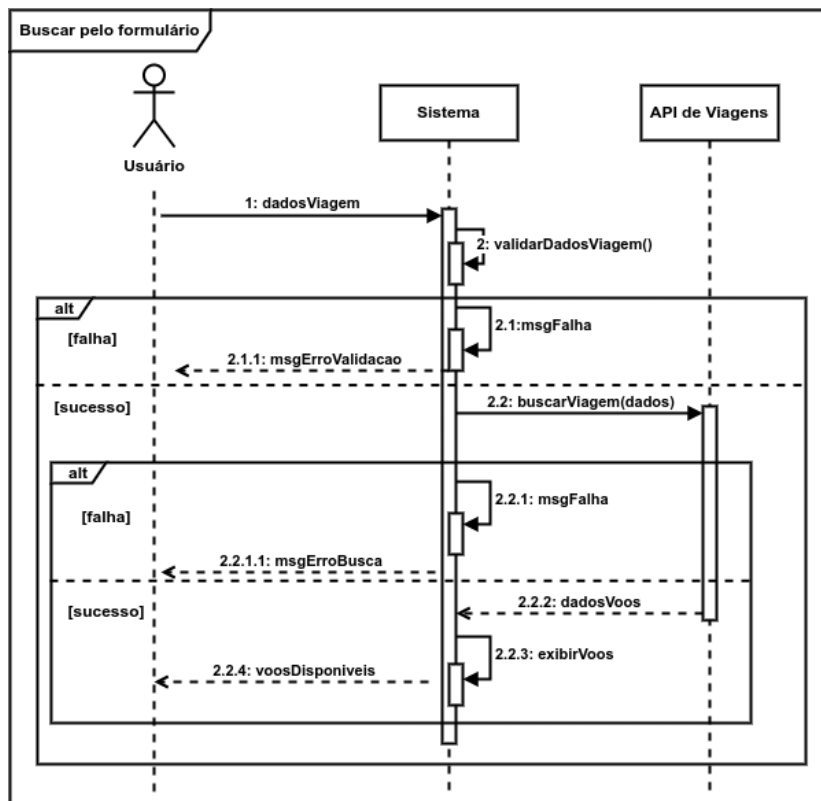
Imagem 16: Diagrama de Sequencia Login/Cadastro



Fonte: Autoria Própria, 2025

Este diagrama, representa o ciclo de vida das interações do objeto "Usuário" com o sistema, com foco nos processos de cadastro, autenticação tradicional (com e-mail e senha) e autenticação via Google. O fluxo inicia-se quando o usuário preenche o formulário de entrada. Em seguida, o sistema realiza a validação dos dados informados. Caso os dados sejam inválidos, o sistema retorna uma mensagem de erro ao usuário, interrompendo o processo de cadastro. Se os dados forem considerados válidos, o sistema executa o procedimento de registro junto ao banco de dados e, posteriormente, retorna uma mensagem ao usuário informando o sucesso ou falha da operação de cadastro. Após o cadastro, o usuário pode realizar o login tradicional informando e-mail e senha. O sistema busca os dados no banco e, conforme o resultado da verificação, informa ao usuário se a autenticação foi bem-sucedida ou não. O diagrama contempla ainda a alternativa de login por meio da conta Google. Nesse caso, os dados do usuário são encaminhados para autenticação externa, e o sistema, ao receber a resposta da API, comunica ao usuário o sucesso ou a falha na tentativa de login.

Imagem 17: Diagrama de Sequência de Buscar pelo Formulário



Fonte: Autoria Própria, 2025

Este diagrama, descreve o fluxo de interação entre o usuário, o sistema e a API de viagens durante o processo de busca de voos por meio de um formulário. O processo tem início quando o usuário envia os dados da viagem desejada. O sistema, então, realiza a validação dessas informações. Caso os dados informados estejam incorretos ou incompletos, o sistema interrompe o fluxo e envia ao usuário uma mensagem de erro de validação. Por outro lado, se os dados forem válidos, o sistema prossegue com a solicitação de busca junto à API de viagens. Se a resposta da API indicar falha na busca, o sistema comunica ao usuário o erro ocorrido. Em caso de sucesso, os dados dos voos são retornados ao sistema, que, por sua vez, realiza a exibição das opções disponíveis ao usuário. Ao longo de todo o processo, são previstas mensagens de retorno explícitas em cada ponto de decisão, assegurando ao usuário um feedback claro e imediato sobre o resultado de suas ações.

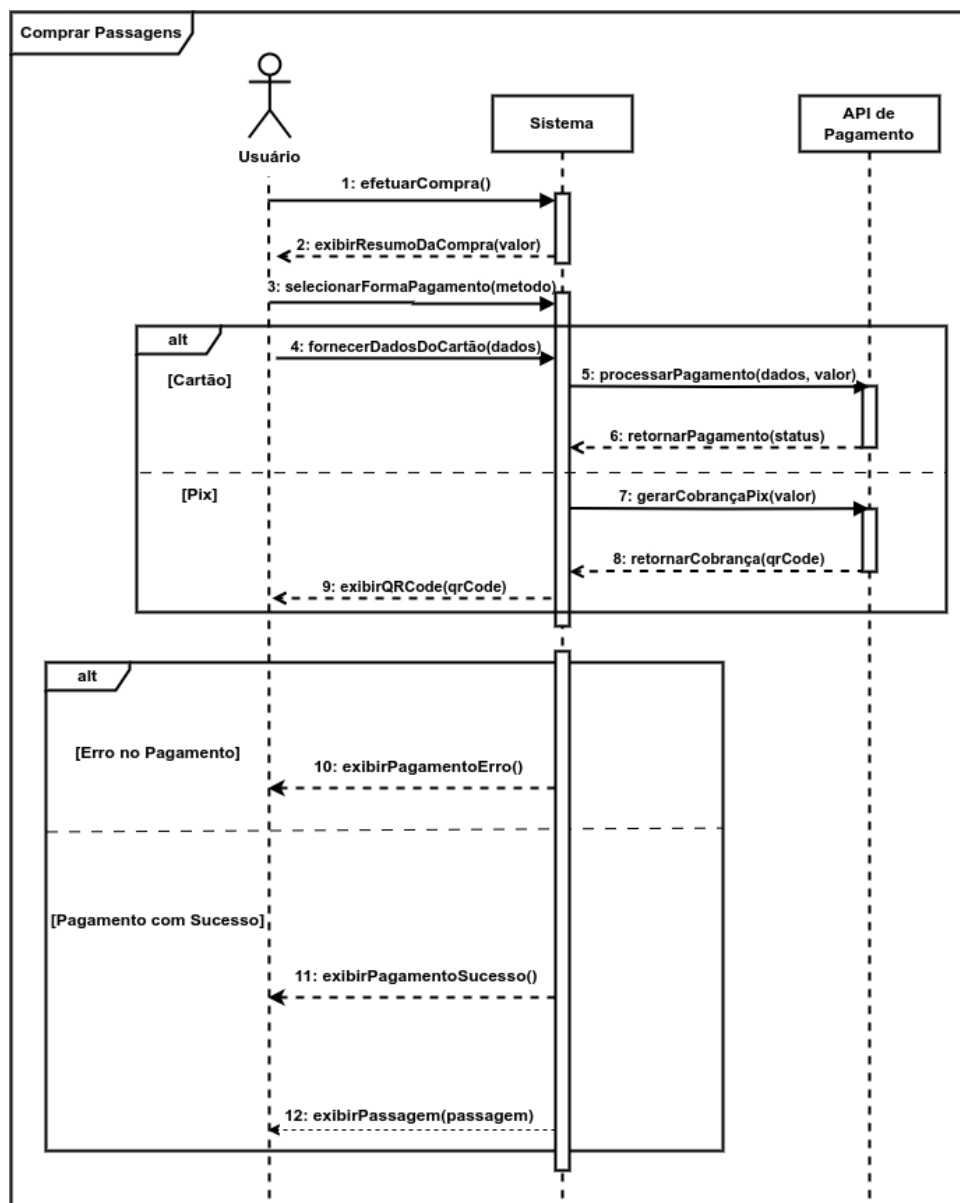
Imagem 18: Diagrama de Sequência Buscar pelo Chatbot

Fonte: Autoria Própria

Este diagrama representa o fluxo de interação entre o usuário, o chatbot e a API de viagens durante o processo de busca de viagens via assistente virtual. O processo é iterativo e começa quando o usuário envia uma mensagem com dados ainda incompletos sobre a viagem desejada.

O chatbot processa a mensagem inicial e retorna uma resposta parcial, solicitando mais informações. Este ciclo se repete até que os dados da viagem estejam completos. Em seguida, o chatbot constrói o contexto da viagem a partir das informações fornecidas. Caso o contexto ainda seja insuficiente, um novo ciclo de interação com o usuário é iniciado até que todos os parâmetros estejam adequadamente definidos. Com os dados e o contexto prontos, o chatbot realiza uma consulta à API de viagens. A resposta recebida pode conter uma lista de resultados ou indicar uma falha. Se a lista estiver vazia ou ocorrer algum erro, o sistema informa o usuário sobre a ausência de resultados. Em casos bem-sucedidos, o chatbot exibe os resultados disponíveis.

Imagem 19: Diagrama de Sequência de Comprar Passagem

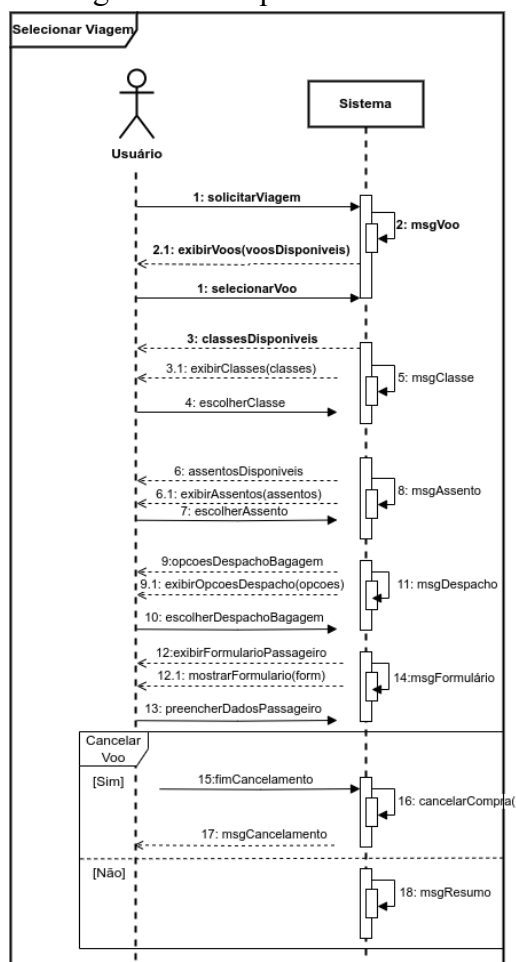


Fonte: Autoria Própria, 2025

Este diagrama representa o fluxo de interação entre o usuário, o chatbot e a API de viagens durante o processo de busca de viagens via assistente virtual. O processo é iterativo e começa quando o usuário envia uma mensagem com dados ainda incompletos sobre a viagem desejada.

O chatbot processa a mensagem inicial e retorna uma resposta parcial, solicitando mais informações. Este ciclo se repete até que os dados da viagem estejam completos. Em seguida, o chatbot constrói o contexto da viagem a partir das informações fornecidas. Caso o contexto ainda seja insuficiente, um novo ciclo de interação com o usuário é iniciado até que todos os parâmetros estejam adequadamente definidos. Com os dados e o contexto prontos, o chatbot realiza uma consulta à API de viagens. A resposta recebida pode conter uma lista de resultados ou indicar uma falha. Se a lista estiver vazia ou ocorrer algum erro, o sistema informa o usuário sobre a ausência de resultados. Em casos bem-sucedidos, o chatbot exibe os resultados disponíveis.

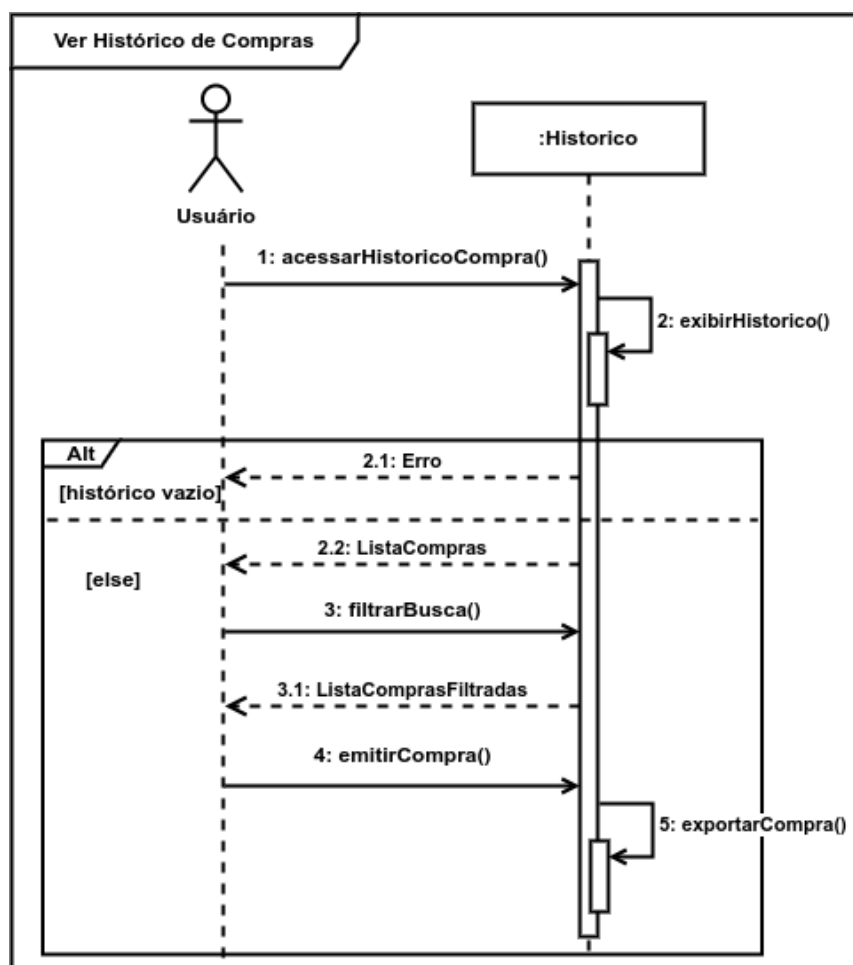
Imagem 20: Diagrama de Sequência de Selecionar Viagem



Fonte: Autoria Própria, 2025

Este diagrama representa o processo de seleção de uma viagem por parte do usuário. O fluxo tem início com a solicitação de viagem, momento em que o sistema retorna a lista de voos disponíveis. Após a escolha de um voo, o sistema fornece as classes disponíveis, seguidas da exibição de assentos, permitindo ao usuário escolher sua preferência. Em seguida, o sistema apresenta as opções de despacho de bagagem, possibilitando ao usuário selecionar a que melhor lhe convém. Após essas definições, é exibido o formulário para preenchimento dos dados do passageiro, encerrando assim a preparação da viagem. Há ainda uma verificação sobre a possibilidade de cancelamento do voo. Caso o usuário opte por cancelar, o sistema finaliza a operação e retorna uma confirmação. Se o cancelamento não for solicitado, o sistema prossegue com a exibição do resumo da viagem.

Imagem 21: Diagrama de Sequência de Histórico de Compras



Fonte: Autoria Própria, 2025

Este diagrama representa o processo de visualização do histórico de compras por parte do usuário. O fluxo tem início quando o usuário solicita o acesso ao histórico, ação que aciona o sistema para exibir os dados disponíveis. Em seguida, o sistema realiza uma verificação condicional. Caso o histórico esteja vazio, é retornada uma mensagem de erro informando a ausência de registros. Caso contrário, o sistema apresenta uma lista com as compras realizadas. Com a lista em mãos, o usuário pode aplicar filtros de busca para localizar compras específicas.



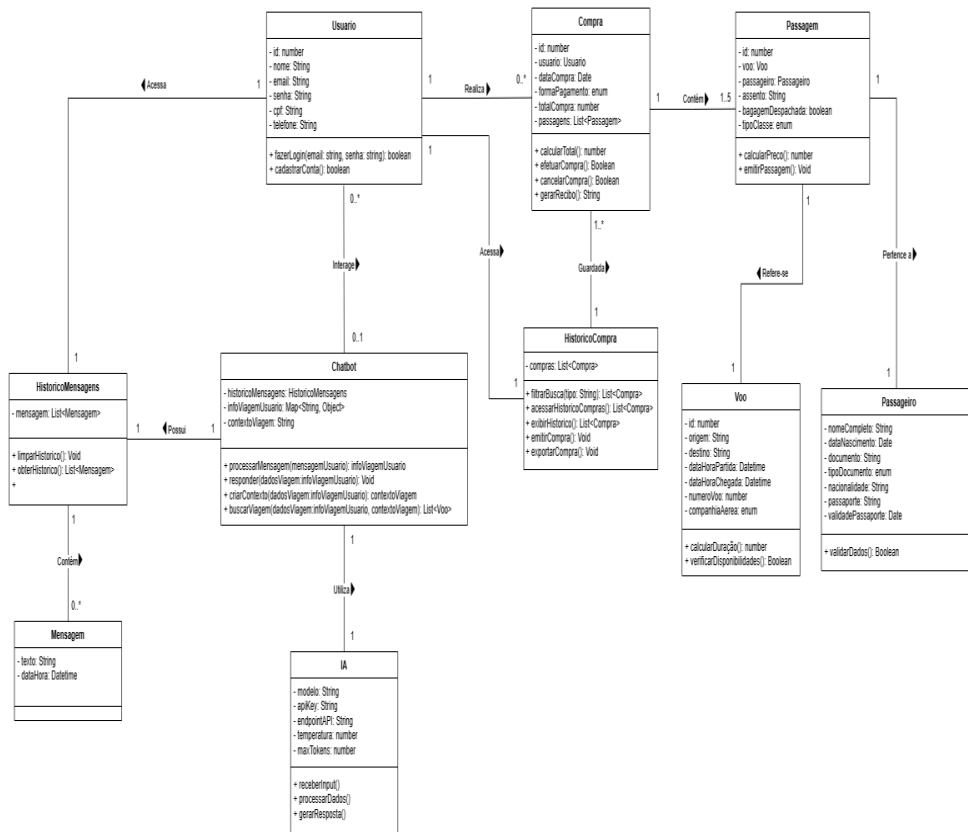
O sistema, então, retorna uma nova lista contendo apenas os resultados que atendem aos critérios definidos

## 4.5 DIAGRAMA DE CLASSES

Esse diagrama representa as estruturas estáticas do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas (como herança, associação etc.). No contexto do desenvolvimento de software, serve como uma espécie de “mapa” conceitual que ilustra como os diferentes componentes de um sistema de relacionam.

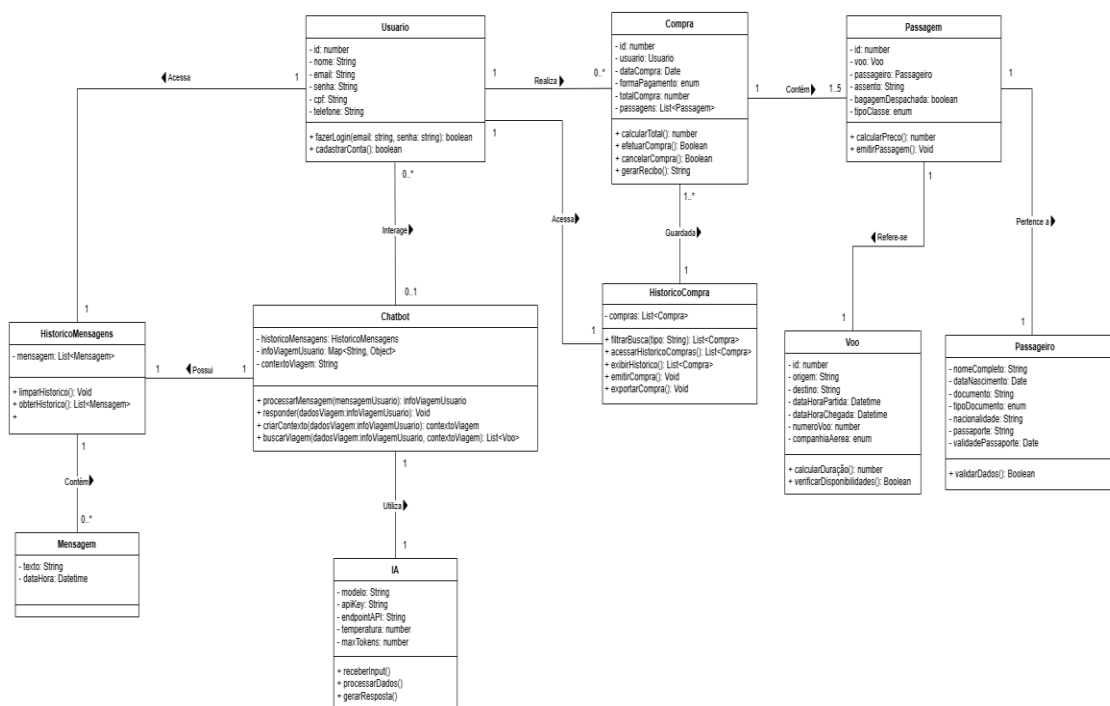
A estrutura principal do SkAI reflete a arquitetura de um assistente virtual de inteligente que tem o objetivo de auxiliar os usuários a encontrarem as melhores opções de voos, utilizando um chatbot integrado com inteligência artificial (IA).

Imagem 22 – Diagrama de classes (a)



Fonte: Autoria própria, 2025.

Imagem 23 – Diagrama de classes (b)



Fonte: Autoria própria, 2025.

#### 4.5.1 USUÁRIO

A classe "Usuario" representa o componente central de interação do sistema. Ela é responsável por modelar informações básicas dos usuários que utilizam a plataforma.

Tabela 3 – Atributos do Usuário

Atributo	Significado
id	Identificador único do usuário
nome	Nome do usuário
email	Endereço de e-mail para login
senha	Senha de acesso ao sistema
cpf	Número de identificação pessoal
telefone	Número para contato

Fonte: Autoria própria, 2025.

Entre suas responsabilidades, têm-se: armazenar dos dados cadastrais do usuário, garantir a autenticação no sistema, realizar compra de passagens, acessar o histórico de compras, interagir com o chatbot e acessar o histórico de conversas.

Tabela 4 – Métodos do Usuário

Método	Significado
fazerLogin(email, senha)	Verifica as credenciais e permite o acesso ao sistema

cadastrarConta( )	Realiza o registro de um novo usuário no banco de dados
-------------------	---

Fonte: Autoria própria, 2025.

#### 4.5.2 CHATBOT

A classe “Chatbot” representa o assistente virtual do sistema, responsável por interagir diretamente com os usuários. Sua principal função é compreender as mensagens recebidas, extrair informações relevantes e responder de forma contextualizada, com o apoio da inteligência artificial integrada.

Tabela 5 – Atributos do Chatbot

Atributo	Significado
historicoMensagens	Lista de mensagens trocados com o usuário
infoViagemUsuario	Estrutura de dados que armazena informações extraídas das mensagens.
contextoViagem	Estrutura de dados que armazena informações extraídas das mensagens.

Fonte: Autoria própria, 2025.

Esta classe é responsável por interpretar comandos dos usuários, construir o contexto da conversa e fornecer respostas inteligentes. Ele também realiza buscas por voos e interage com a IA para gerar respostas personalizadas.

Tabela 6 – Métodos do Chatbot

Método	Significado
processarMensagem(mensagemUsuario)	Analisa a mensagem recebida e extrai dados relevantes.
responder(dadosViagem)	Gera e envia uma resposta ao usuário.
criarContexto(dadosViagem)	Mantém a continuidade do diálogo com base em interações anteriores.
buscarViagem(dadosViagem, contexto)	Realiza a busca por voos conforme os dados fornecidos.

Fonte: Autoria própria, 2025.

#### 4.5.3 IA

A classe “IA” representa o módulo de inteligência artificial do sistema, utilizado para interpretar mensagens, gerar respostas, criar contextos e auxiliar na tomada de decisão automatizada do sistema.

Tabela 7 – Atributos da IA

Atributo	Significado
modelo	Nome do modelo de linguagem utilizado.

apikey	Chave de acesso à API da IA.
endpointAPI	URL da API utilizada para comunicação com o modelo.
temperatura	Grau de aleatoriedade na geração das respostas.
maxTokens	Limite máximo de tokens na resposta da IA.

Fonte: Autoria própria, 2025.

Esta classe é responsável por processar as entradas de texto enviadas pelo Chatbot, gerar respostas coerentes, sumarizar informações, interpretar comandos e fornecer sugestões contextuais aos usuários com base no histórico de interações.

Tabela 8 – Métodos da IA

Método	Significado
receberInput( )	Recebe o texto de entrada enviado pelo chatbot.
processarDados( )	Processa o conteúdo recebido e interpreta a intenção do usuário.
GerarResposta( )	Retorna uma resposta textual baseada no contexto e dados processados.

Fonte: Autoria própria, 2025.

#### 4.5.4 HISTÓRICO DE MENSAGENS

A classe “HistoricoMensagens” representa o conjunto de interações registradas entre o usuário e o chatbot do sistema. Sua função é armazenar, organizar e fornecer acesso ao histórico completo de mensagens trocadas, possibilitando que o sistema mantenha o contexto das conversas.

Tabela 9 – Atributos do Histórico de Compras

Atributo	Significado
mensagens	Lista de mensagens trocadas durante as interações com o chatbot.

Fonte: Autoria própria, 2025.

Essa classe é responsável por registrar todas as mensagens enviadas e recebidas, garantindo que o chatbot consiga manter uma conversação contínua e contextualizada. Além disso, possibilita consultas e análises posteriores das interações do usuário com o sistema.

Tabela 10 – Métodos do Histórico de Compras

Método	Significado
limparHistorico( )	Remove todas as mensagens registradas.
obterHistorico( )	Retorna a lista completa de mensagens trocadas.

Fonte: Autoria própria, 2025.

#### 4.5.5 MENSAGEM

A classe “Mensagem” representa uma única mensagem individual trocada entre o usuário e o chatbot, podendo ser tanto uma entrada do usuário quanto uma resposta do sistema.

Tabela 11 – Atributos da Mensagem

Atributo	Significado
texto	Conteúdo textual da mensagem.
dataHora	Data e hora exata em que a mensagem foi enviada ou recebida.

Fonte: Autoria própria, 2025.

Esta classe tem a responsabilidade de armazenar o conteúdo e o momento exato de envio de cada mensagem, possibilitando o rastreamento cronológico das interações.

#### 4.5.6 COMPRA

A classe “Compra” representa o processo de aquisição de passagens por parte do usuário no sistema. Ela centraliza os dados relacionados a uma transação, como data, forma de pagamento, valor total e as passagens adquiridas.

Tabela 12 – Atributos de Compra

Atributo	Significado
id	Identificador único da compra.
usuario	Referência ao usuário que realizou a compra.
dataCompra	Data e hora em que a compra foi efetuada.
formaPagamento	Forma de pagamento utilizada (ex: cartão, pix).
totalCompra	Valor total da compra.
passagens	Lista de passagens adquiridas na transação.

Fonte: Autoria própria, 2025.

Esta classe é responsável por registrar e gerenciar todas as informações de uma transação realizada pelo usuário. Ela calcula o valor total com base nas passagens escolhidas, permite o cancelamento, gera recibos e mantém o vínculo entre o usuário e suas passagens.

Tabela 13 – Métodos de Compra

Método	Significado
calcularTotal	Calcula e retorna o valor total da compra com base nas passagens.
efetuarCompra	Processa e confirma a realização da compra.
cancelarCompra	Realiza o cancelamento da compra, se possível.
gerarRecibo	Gera um comprovante da transação.

Fonte: Autoria própria, 2025.

#### 4.5.7 HISTÓRICO DE COMPRAS

A classe “HistoricoCompra” representa o conjunto de todas as compras realizadas por um usuário dentro do sistema. Sua função é armazenar, organizar e permitir o acesso ao histórico completo de transações, facilitando a visualização, reemissão e controle das passagens adquiridas.

Tabela 14 – Atributos do Histórico de Compras

Atributo	Significado
compras	Lista contendo todas as transações efetuadas pelo usuário.

Fonte: Autoria própria, 2025.

Esta classe é responsável por manter um registro completo das compras feitas ao longo do tempo, permitindo que o usuário consulte transações anteriores, filtre informações específicas e exporte dados.

Tabela 15 – Métodos de Compra

Método	Significado
filtrarBusca(tipo: String)	Permite filtrar o histórico com base em critérios como data, destino ou forma de pagamento.
acessarHistoricoCompras( )	Retorna a lista completa de compras registradas.
exibirHistorico( )	Exibe as compras de forma organizada.
emitirCompra( )	Gera uma nova emissão de comprovante para uma compra específica.
exportarCompra( )	Exporta os dados das compras para um formato externo (ex: PDF).

Fonte: Autoria própria, 2025.

#### 4.5.8 PASSAGEM

A classe “Passagem” representa uma passagem aérea individual adquirida por um usuário no sistema. Ela contém as informações essenciais da reserva, como o voo associado, os dados do passageiro, o assento escolhido, a classe da passagem e se há bagagem despachada.

Tabela 16 – Atributos da Passagem

Atributo	Significado
id	Identificador único da passagem.
voo	Voo ao qual a passagem está vinculada.
passageiro	Dados da pessoa que utilizará a passagem.

assento	Código do assento reservado (ex: 12A, 15C).
bagagemDespachada	Indica se a passagem inclui bagagem despachada.
tipoClasse	Define a classe da passagem (Econômica, Executiva, Primeira Classe).

Fonte: Autoria própria, 2025.

Esta classe é responsável por consolidar todos os dados referentes ao bilhete aéreo de um passageiro. Ela garante que as informações estejam corretamente vinculadas ao voo e ao usuário, além de permitir o cálculo de valores e a emissão da passagem.

Tabela 17 – Métodos da Passagem

Método	Significado
calcularPreco( )	Calcula o valor da passagem com base na classe, bagagem e voo.
emitirPassagem( )	Gera e disponibiliza a passagem para o passageiro, com todos os dados necessários.

Fonte: Autoria própria, 2025.

#### 4.5.9 PASSAGEIRO

A classe “Passageiro” representa a pessoa que irá utilizar uma determinada passagem aérea adquirida por meio do sistema. Essa classe armazena os dados pessoais e documentais do indivíduo, necessários para a emissão da passagem.

Tabela 18 – Atributos do Passageiro

Atributo	Significado
nomeCompleto	Nome completo do passageiro.
dataNascimento	Data de nascimento do passageiro.
documento	Documento de identificação (RG, passaporte, etc.).
tipoDocumento	Tipo do documento informado (RG, CPF, Passaporte, etc.).
nacionalidade	Nacionalidade do passageiro.
passaporte	Número do passaporte (obrigatório para voos internacionais).
validadePassaporte	Data de expiração do passaporte.

Fonte: Autoria própria, 2025.

Esta classe é responsável por fornecer todos os dados do indivíduo que utilizará a passagem aérea. Esses dados são essenciais para o preenchimento do bilhete, validação junto à companhia aérea.

Tabela 19 – Métodos do Passageiro

Método	Significado
--------	-------------

validarDados( )	Verifica se todos os campos obrigatórios estão preenchidos corretamente e se os documentos estão válidos.
-----------------	---

Fonte: Autoria própria, 2025.

#### 4.5.10 VOO

A classe “Voo” representa uma operação aérea disponível no sistema. Ela contém todas as informações necessárias sobre a rota, horários, companhia aérea e identificação do voo, sendo essencial para o planejamento e emissão de passagens.

Tabela 20 – Atributos do Voo

Atributo	Significado
id	Identificador único do voo.
origem	Cidade ou aeroporto de origem do voo.
destino	Cidade ou aeroporto de destino do voo.
dataHoraPartida	Data e hora da partida do voo.
dataHoraChegada	Data e hora previstas de chegada.
numeroVoo	Código do voo (ex: G3 1234).
companhiaAerea	Nome da companhia aérea responsável (ex: LATAM, Azul, Gol).

Fonte: Autoria própria, 2025.

Esta classe é responsável por fornecer todas os dados referentes ao deslocamento aéreo associado a passagem. Ela permite o controle e a validação de horários, rotas e disponibilidade de lugares no momento da emissão da passagem.

Tabela 21 – Métodos do Voo

Método	Significado
calcularDuracao( )	Calcula a duração do voo em horas e minutos.
verificarDisponibilidade( )	Verifica se ainda há assentos disponíveis para venda.

Fonte: Autoria própria, 2025.

## 5. CONSIDERAÇÕES FINAIS

Este documento serve como base para o desenvolvimento do sistema, sendo passível de revisões durante o processo. A colaboração entre os desenvolvedores e usuários será essencial para garantir que os requisitos estejam completos e corretos.



## REFERÊNCIAS

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML: um guia prático para modelagem de sistemas orientados a objetos através da UML. 2. ed. Rio de Janeiro: Elsevier, 2008.

IATA. Key Trends in Passenger Behavior: Insights from the IATA Global Passenger Survey 2024 [site]. 5 de novembro de 2024. Disponível em: <https://www.iata.org/en/publications/newsletters/iata-knowledge-hub/key-trends-in-passenger-behavior/>. Acesso em: 02 de junho de 2025.

FOWLER, Martin. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005.

OBJECT MANAGEMENT GROUP (OMG). Unified Modeling Language (UML) Version 2.5.1. [S. l.]: OMG, 2017. Disponível em: <https://www.omg.org/spec/UML/2.5.1/PDF>.

## RECONHECIMENTOS E DIREITOS AUTORAIS

@autor: Kauan Guilherme Alves Pinheiro Santos, Lilia Rosa Coelho Moura, Raylan Bruno Santana Carvalho, Virginia Maria Mondego Ferreira, Yasmin Cantanhede Santos.

@contato: kauan.santos@discente.ufma.br, liliarosaestudos@gmail.com, raylan.bruno@discente.ufma.br, virginiamondego@gmail.com, yasmin.cantanhede@discente.ufma.br

@data última versão: 13/06/25

@versão: 1.0

@outros repositórios: kauanguilhermesantos, LiliaRosa19, Raylan-BR, VirginiaMond, yasmincantanhede

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

## Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina PROJETO E DESENVOLVIMENTO DE SOFTWARE, sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2025.1, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da Licença MIT, conforme descrito abaixo, e, portanto, é compatível com a GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:

Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>