Estudante: Filipe Luiz Orlamünder

Instituição: Católica SC - Centro Universitário

Curso: Engenharia de Software

Data de Entrega: 10/07/2025

Projeto: Check Milhas –

Gestão Inteligente de Emissões de Passagens Aéreas com Milhas



Resumo

Este projeto propõe o desenvolvimento do Check Milhas, um sistema digital robusto e intuitivo dedicado ao controle e à gestão da emissão de passagens aéreas utilizando milhas acumuladas nos programas de fidelidade Latam Pass, Smiles e Azul Fidelidade.

A crescente complexidade das regras específicas de cada programa, especialmente no que tange ao cadastro e gerenciamento de beneficiários, representa um desafio significativo para os usuários. O Check Milhas visa solucionar essa questão, permitindo o cadastro de múltiplos perfis de usuário e o controle individualizado de beneficiários para cada programa, respeitando seus limites e prazos. O sistema oferecerá funcionalidades como adição, edição, exclusão e verificação do status dos beneficiários, com uma exibição dinâmica e clara das informações.

O objetivo central é proporcionar maior eficiência no uso estratégico das milhas, prevenindo bloqueios ou restrições por descumprimento das normativas dos programas de fidelidade. Este documento detalha o contexto, os objetivos, a descrição funcional e técnica, as considerações de design e a stack tecnológica do projeto, bem como os próximos passos para sua implementação.

1. Introdução

Contexto

A utilização de milhas aéreas como forma de pagamento para emissão de passagens tornou-se uma prática disseminada entre os consumidores, impulsionada pela expansão dos programas de fidelidade oferecidos pelas companhias aéreas. Apesar da popularidade, essa modalidade de resgate é permeada por uma série de regras e restrições específicas impostas por cada programa em relação ao número de beneficiários permitidos para a emissão de passagens em nome de terceiros.

Para o usuário comum, acompanhar e gerenciar essas informações, que frequentemente variam entre os programas, pode se configurar como uma tarefa complexa. Essa complexidade é intensificada em situações onde um indivíduo administra as milhas de diversos membros da família, busca otimizar o resgate de passagens para grupos ou atua informalmente na intermediação de passagens aéreas utilizando seu saldo de milhas. Esta última prática, embora comum, exige um controle extremamente rigoroso para evitar a infração das regras dos programas, que podem resultar em penalidades como o bloqueio da conta.

Justificativa

A idealização do projeto Check Milhas emerge da necessidade de ferramentas que auxiliem no gerenciamento eficaz e seguro de milhas aéreas. Erros no controle de beneficiários ou o descumprimento das regras de emissão podem acarretar consequências severas, incluindo a perda temporária do acesso à conta ou, em casos mais graves, o bloqueio vitalício do usuário no programa de fidelidade. Diante desse cenário, um sistema centralizado, com interface intuitiva e que automatize parte desse controle, apresenta-se como uma solução valiosa para indivíduos que buscam maximizar o uso estratégico de seus programas de fidelidade com segurança.

Ademais, à medida que um número crescente de pessoas adota as milhas como um meio de economizar em viagens aéreas, a demanda por sistemas que simplifiquem esse processo também aumenta. Do ponto de vista da Engenharia de Software, o projeto oferece desafios interessantes no que tange à modelagem de regras de negócio variáveis e complexas, ao design de uma interface de usuário clara e eficiente para apresentação de dados dinâmicos e à garantia da integridade e segurança das informações gerenciadas pelo sistema.

Objetivos

Objetivo Geral:

* Analisar a complexidade crescente na gestão de milhas aéreas para a emissão de passagens, identificando os principais desafios enfrentados pelos usuários. Pretende-se aprofundar o entendimento sobre as diversas regras e restrições dos programas de fidelidade, visando elucidar o cenário atual e as dificuldades inerentes ao uso estratégico desses benefícios.

Objetivos Específicos:

* Expor as dificuldades na administração de múltiplos perfis e contas de milhas, destacando o empenho necessário para manter a organização e o controle eficaz.
* Propor uma forma de acompanhar o status de beneficiários em diferentes plataformas, que muitas vezes levam a equívocos e perdas de oportunidades.

2. Descrição do Projeto

Tema do Projeto

O Check Milhas será um sistema web projetado para centralizar e automatizar o controle de beneficiários na emissão de passagens aéreas com milhas. O foco estará nas regras específicas dos principais programas de fidelidade brasileiros: Latam Pass, Smiles e Azul Fidelidade.

Cada usuário do Check Milhas poderá cadastrar e gerenciar até 10 perfis distintos. Esses perfis podem representar diferentes pessoas físicas (o próprio usuário, familiares, amigos ou clientes para quem o usuário presta algum tipo de assessoria informal). Para cada um desses perfis, será possível vincular contas dos três programas de fidelidade mencionados, permitindo um controle individualizado das regras e limitações aplicáveis a cada conta de milhagem.

O sistema tem como diferencial a exibição clara e objetiva do status atualizado de cada beneficiário, diretamente na listagem do respectivo programa de fidelidade. Essa visualização permitirá que o usuário acompanhe prazos, limites e restrições com base nas informações de emissão de passagens que ele mesmo insere e atualiza no sistema.

Dessa forma, o Check Milhas auxiliará na prevenção de erros comuns, como a extrapolação do limite de beneficiários, contribuindo para o uso consciente e seguro das contas de milhas e evitando surpresas desagradáveis como suspensões ou bloqueios.

As regras específicas consideradas para cada programa são:

* Latam Pass:
  + O titular da conta pode emitir passagens para si de forma ilimitada.
  + Para terceiros, o programa permite a emissão para até 25 CPFs distintos em um período de 12 meses.
  + Não há exigência de cadastro prévio de beneficiários; o controle é feito pela Latam no momento da emissão, com base nos CPFs para os quais passagens foram emitidas no período.
* Smiles:
  + O titular da conta pode emitir passagens para si de forma ilimitada.
  + Para terceiros, o programa permite a emissão para até 25 CPFs distintos por ano civil (1º de janeiro a 31 de dezembro).
  + Similar ao Latam Pass, não há necessidade de cadastro prévio de beneficiários; o controle é realizado pela Smiles no momento da emissão.
* Azul Fidelidade:
  + O titular da conta pode emitir passagens para si de forma ilimitada.
  + É permitido cadastrar uma lista de até 5 beneficiários fixos vinculados à conta.
  + A substituição de um beneficiário nesta lista implica um período de carência de 60 dias, durante o qual nenhuma passagem pode ser emitida para o beneficiário que está sendo substituído nem para o novo beneficiário cadastrado.
  + Caso o usuário necessite emitir uma passagem para um beneficiário durante o período de carência de substituição, o processo de alteração vigente deverá ser cancelado, pois emissões são bloqueadas durante essa janela de troca.

A funcionalidade Validação Dinâmica será um botão que permitirá ao usuário verificar a quantidade de beneficiários que ainda podem ser adicionados em cada programa de fidelidade, como LATAM Pass, Smiles e Azul Fidelidade, listando essa informação de todos os perfis já cadastrados e possibilitando informar uma data futura para validação.

Problemas a Resolver

O projeto Check Milhas visa solucionar os seguintes problemas principais enfrentados pelos usuários de programas de milhagem:

* Complexidade no Gerenciamento Manual: Dificuldade em rastrear manualmente os limites de beneficiários, datas de emissão e períodos de carência em múltiplos programas de fidelidade, cada um com suas regras distintas.
* Risco de Penalidades: Desconhecimento, esquecimento ou interpretação equivocada das regras pode levar ao bloqueio de contas ou outras penalidades impostas pelas companhias aéreas.
* Falta de Visão Centralizada: Ausência de uma ferramenta que consolide as informações de diferentes programas e perfis, dificultando uma visão clara do status dos beneficiários e dos limites disponíveis.
* Gestão Ineficiente para Múltiplas Contas: Dificuldade em gerenciar milhas e beneficiários para diferentes pessoas (familiares, amigos, ou em cenários de venda informal de passagens), o que exige controle individualizado e rigoroso.

Limitações

É importante delimitar o escopo do projeto Check Milhas. O sistema não abordará:

* Integração Automática com APIs das Companhias Aéreas: O sistema não se conectará diretamente aos sistemas das companhias aéreas para buscar saldos de milhas, verificar status de beneficiários em tempo real.
* Compra e Venda de Milhas: O sistema é focado na gestão de beneficiários e não incluirá funcionalidades para transacionar milhas.
* Cobertura de Todos os Programas de Fidelidade: Inicialmente, o sistema se concentrará exclusivamente nos programas Latam Pass, Smiles e Azul Fidelidade.

3. Especificação Técnica

Requisitos de Software

Os requisitos foram definidos com base nas necessidades práticas dos usuários e nas restrições impostas pelos programas de fidelidade abordados.

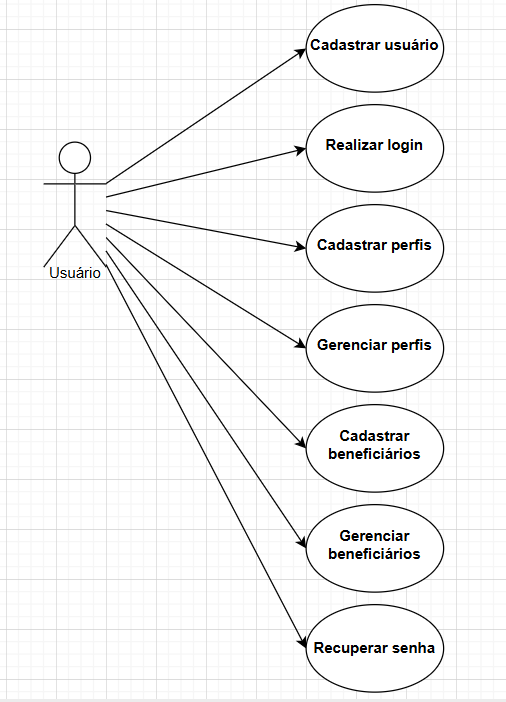
Requisitos Funcionais (RF):

* RF01: Permitir a criação de cadastro de um novo usuário, com login e senha.
* RF02: Permitir o login do usuário.
* RF03: Permitir o cadastro de até 10 perfis por usuário.
* RF04: Permitir que cada perfil tenha possa acessar os 3 programas de fidelidade (Latam Pass, Smiles, Azul Fidelidade).
* RF05: Permitir cadastrar beneficiários com nome completo, CPF e data de emissão/cadastro.
* RF06: Controlar status dos beneficiários conforme regras específicas de cada programa:
  + Latam Pass: Limite de 25 beneficiários a cada 12 meses.
  + Smiles: Limite de 25 beneficiários por ano civil.
  + Azul Fidelidade: Lista fixa de 5 beneficiários, com carência de 60 dias para substituição.
* RF07: Apresentar a lista de beneficiários de forma dinâmica, com os status atualizados.
* RF08: Permitir edição e exclusão de beneficiários com confirmação prévia.
* RF09: Permitir a recuperação de senha para usuários do sistema Check Milhas.
* RF10: Permitir validação dinâmica dos beneficiários que estarão disponíveis.

Requisitos Não-Funcionais (RNF):

* RNF01: A interface do sistema deve ser intuitiva e responsiva.
* RNF02: As informações dos usuários e beneficiários devem ser armazenadas de forma segura no banco de dados.
* RNF03: O sistema deve apresentar bom desempenho, com tempos de resposta rápidos para as interações do usuário.
* RNF04: O sistema deve ser desenvolvido utilizando tecnologias que garantam manutenibilidade e escalabilidade futuras.
* RNF05: O sistema deve ser compatível com os navegadores web modernos mais utilizados.

Representação dos Requisitos - Diagrama de Casos de Uso (UML):



Considerações de Design

Visão Inicial da Arquitetura: O sistema Check Milhas será desenvolvido seguindo uma arquitetura cliente-servidor.

* Cliente (Frontend): Uma aplicação web single-page application (SPA) desenvolvida em React, TypeScript, Vite e Tailwind CSS responsável pela interface do usuário e comunicação com o backend.
* Servidor (Backend): Uma API REST desenvolvida em Node.js, responsável pela lógica de negócios, processamento das regras dos programas e persistência no banco de dados.
* Banco de Dados: Um banco de dados relacional MySQL para armazenamento dos dados.

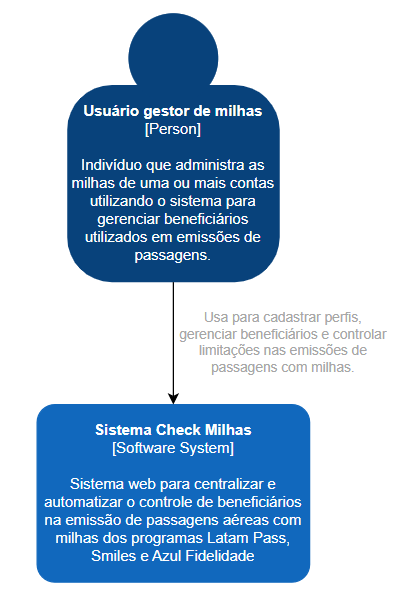
Padrões de Arquitetura:

* Frontend: Serão considerados padrões de arquitetura e design de componentes adequados ao React, TypeScript, Vite e Tailwind CSS visando a reutilização e manutenibilidade, além de um eficiente gerenciamento de estado.
* Backend: Será adotada uma arquitetura em camadas para promover a separação de responsabilidades, testabilidade e manutenibilidade do código da API.

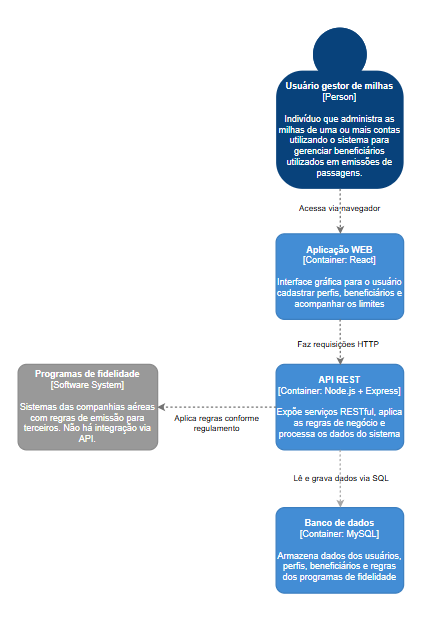
Modelos C4: Detalhamento da Arquitetura em Níveis

A arquitetura do sistema Check Milhas será documentada utilizando o Modelo C4, que oferece uma visão hierárquica em diferentes níveis de abstração para facilitar a compreensão.

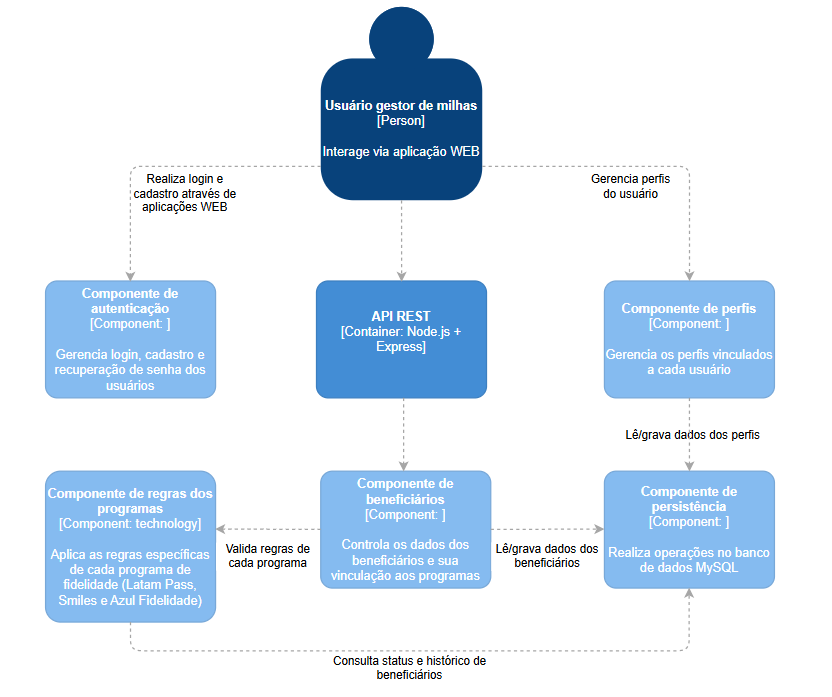
* Nível 1 - Contexto (System Context Diagram):
  + Este nível apresenta uma visão geral do sistema Check Milhas e suas interações externas. O sistema interage com usuários (indivíduos que gerenciam suas milhas e emissões de passagens) e é influenciado pelas regras dos Programas de Fidelidade (Latam Pass, Smiles, Azul Fidelidade).



* Nível 2 - Contêiner (Container Diagram):
  + O diagrama de contêineres detalha as principais aplicações e serviços que compõem o Check Milhas. São eles: a Aplicação Web (Frontend) em React, que é a interface do usuário no navegador ; a API REST (Backend) em Node.js, que gerencia a lógica de negócios e as regras dos programas ; e o Banco de Dados (MySQL), para persistência de todos os dados do sistema.



* Nível 3 - Componente (Component Diagram):
  + Neste nível, um contêiner específico (como a API REST) é detalhado em seus componentes internos. Exemplos de componentes para o Backend incluem: Gerenciamento de Usuários (autenticação e autorização) , Gerenciamento de Perfis e Beneficiários (cadastro, edição, exclusão e aplicação das regras específicas de cada programa como Latam Pass, Smiles e Azul Fidelidade), e um Componente de Persistência para interação com o banco de dados.



* Nível 4 - Código (Code Diagram):
  + O nível de Código é o mais granular, mostrando a estrutura interna de um componente em termos de classes, interfaces e métodos. Para o Check Milhas, isso envolveria a representação de classes como Beneficiario, que conteriam atributos como nome, CPF e data de emissão/cadastro, e a lógica para aplicar as regras de cada programa de fidelidade.

Stack Tecnológica

Linguagens de Programação:

* Frontend: Desenvolvimento com React, TypeScript Vite e Tailwind CSS.
* Backend: JavaScript (ou TypeScript) para desenvolvimento com Node.js.
* *Justificativa:* A escolha do JavaScript/TypeScript para ambas as camadas unifica a linguagem, facilitando o desenvolvimento. React é uma biblioteca consolidada para interfaces de usuário dinâmicas. Node.js é eficiente para APIs web e possui um vasto ecossistema.

Frameworks e Bibliotecas:

* Frontend (React):
  + Utilizará o ecossistema React, incluindo bibliotecas para gerenciamento de rotas.
  + Outras bibliotecas para componentes UI, gerenciamento de estado e requisições HTTP poderão ser avaliadas e incorporadas conforme a necessidade durante o desenvolvimento.
* Backend (Node.js):
  + Express.js como framework principal para a construção da API.
  + Outras bibliotecas para funcionalidades específicas, como autenticação e validação de dados, serão incorporadas conforme necessário.
* Banco de Dados:
  + MySQL: Um sistema de gerenciamento de banco de dados relacional.
  + *Justificativa:* MySQL é uma solução robusta, amplamente utilizada e adequada para armazenar dados estruturados como os do Check Milhas.
* Ferramentas de Desenvolvimento e Gestão de Projeto:
  + Controle de Versão: Git, com repositório hospedado no GitHub.
  + IDE: Visual Studio Code (VS Code).
  + Gestão de Projeto: Trello.
  + Gerenciador de Pacotes: npm ou Yarn.
  + Outras ferramentas de apoio ao desenvolvimento, teste e implantação serão definidas e adotadas conforme a necessidade ao longo do projeto.

Considerações de Segurança

Autenticação e Autorização:

* Implementação de um sistema de cadastro e login seguro, com senhas armazenadas utilizando hashing forte.
* Mecanismos de autorização para assegurar que um usuário só possa acessar e modificar seus próprios dados.

Proteção de Dados:

* Criptografia em Trânsito: Uso de HTTPS (TLS/SSL).
* Gerenciamento de Dependências: Monitoramento e atualização regular das dependências.
* Conformidade com a LGPD (Lei Geral de Proteção de Dados): O sistema será desenvolvido buscando aderência aos princípios da LGPD.

4. Próximos Passos

Após a aprovação e refinamento deste documento, os próximos passos para o desenvolvimento do Check Milhas serão:

1. Portfólio I - Planejamento e Prototipação:
   * Revisão Final da Documentação.
   * Design da Interface do Usuário (UI) e Experiência do Usuário (UX) (wireframes, mockups).
   * Modelagem do Banco de Dados (esquema final).
   * Configuração do Ambiente de Desenvolvimento (ferramentas, repositórios).
2. Portfólio II - Desenvolvimento e Validação:
   * Desenvolvimento do sistema.
   * Testes unitários, de integração e de usabilidade (conforme viabilidade).
   * Refinamento e Correção de Bugs.
   * Preparação para Implantação (ambiente de hospedagem).
   * Preparação para Apresentação Final.

5. Referências

* [Regulamento Latam Pass](https://latampass.latam.com/pt_br/descubra-latam-pass/termos-e-condicoes) - https://latampass.latam.com/pt\_br/descubra-latam-pass/termos-e-condicoes
* [Regulamento Smiles](https://www.smiles.com.br/regulamento-do-programa-smiles-novo) - https://www.smiles.com.br/regulamento-do-programa-smiles-novo
* [Regulamento Azul Fidelidade](https://www.voeazul.com.br/br/pt/programa-fidelidade/regulamento/novo-2024.html) - https://www.voeazul.com.br/br/pt/programa-fidelidade/regulamento/novo-2024.html
* [Revista Exame (Limite de Venda de Milhas](https://exame.com/invest/guia/qual-o-limite-de-vendas-de-milhas-por-cpf/)) - https://exame.com/invest/guia/qual-o-limite-de-vendas-de-milhas-por-cpf/

6. Avaliações de Professores

Assinatura do Professor(a):

Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_