

Histórias de usuário:

Must have

Should have

Could have

Won't have

- Como usuário eu quero poder me cadastrar informando minhas informações pessoais para criar minha conta na aplicação.
 - Justificativa INVEST:
 - **Independent:** Pode ser desenvolvido separadamente das demais funcionalidades.
 - **Negotiable:** Os campos obrigatórios podem ser ajustados conforme decisão do time.
 - **Valuable:** Permite identificar o usuário e associar compras ao seu perfil.
 - **Estimable:** É possível estimar esforço técnico com base em criação de formulário, validação e persistência.
 - **Small:** Envolve apenas o registro do usuário.
 - **Testable:** Pode ser testado validando criação com dados corretos e bloqueio com dados inválidos.
 - Justificativa Moscow:
 - O cadastro é essencial para identificar o usuário, associar compras ao seu perfil e permitir acesso à área do cliente. Sem essa funcionalidade, não é possível vincular passagens compradas a um usuário específico, comprometendo o funcionamento do sistema.
- Como usuário eu quero poder logar na aplicação utilizando as informações cadastradas para acessar minha conta.
 - Justificativa INVEST:
 - **Independent:** Não depende da busca ou pagamento para ser implementado.
 - **Negotiable:** Pode incluir melhorias futuras (ex: autenticação em dois fatores).
 - **Valuable:** Garante acesso seguro às funcionalidades restritas.
 - **Estimable:** Implementação clara (validação dos dados do usuário e autenticação).
 - **Small:** Abrange apenas a autenticação do usuário.
 - **Testable:** Pode ser testado com credenciais válidas e inválidas.
 - Justificativa Moscow:
 - O login permite autenticação e acesso seguro às funcionalidades restritas (como carrinho e histórico). Sem ele, não há controle de acesso nem segurança mínima. É essencial para garantir uso individualizado da plataforma.

- Como usuário eu quero poder buscar passagens informando origem e destino para visualizar viagens disponíveis.

- Justificativa INVEST:

- **Independent:** Pode ser implementada antes do carrinho ou pagamento devido à componentização das telas.
- **Negotiable:** Forma de busca pode evoluir (ex: autocomplete).
- **Valuable:** Permite encontrar viagens disponíveis.
- **Estimable:** Escopo claro (consulta com filtros básicos).
- **Small:** Foco apenas na busca por origem/destino.
- **Testable:** Pode-se validar se retorna resultados corretos.

- Justificativa Moscow:

- Sem a funcionalidade de pesquisa, o usuário não consegue encontrar passagens para compra. O sistema perderia seu propósito principal, que é permitir a escolha de viagens.

- Como usuário eu quero poder filtrar passagens por data para escolher quando viajar.

- Justificativa INVEST:

- **Independent:** Complementa a busca, mas pode ser entregue separadamente.
- **Negotiable:** Pode incluir filtros avançados no futuro.
- **Valuable:** Facilita encontrar viagens na data desejada.
- **Estimable:** Implementação simples como filtro adicional.
- **Small:** Atua apenas sobre o resultado da busca.
- **Testable:** Pode ser testado validando exibição correta por data.

- Justificativa Moscow:

- Melhora significativamente a experiência do usuário, pois viagens dependem de datas. Porém, em um cenário mínimo, seria possível exibir todas as passagens disponíveis e permitir escolha manual. Não bloqueia o funcionamento do sistema.

- Como usuário eu quero poder filtrar passagens por tipo (aérea ou marítima) para encontrar a opção desejada.

- Justificativa INVEST:

- **Independent:** Não depende de outras funcionalidades, ex: pagamento ou cadastro.
- **Negotiable:** Pode incluir mais categorias futuramente.
- **Valuable:** Ajuda o usuário a encontrar o tipo desejado.
- **Estimable:** Filtro simples baseado em categoria.
- **Small:** Apenas segmentação dos resultados.
- **Testable:** Testável verificando se retorna apenas o tipo selecionado.

- Justificativa Moscow:

- É uma melhoria de usabilidade que facilita a navegação e segmentação. Porém, o sistema poderia funcionar exibindo todas as opções juntas inicialmente.

- Como usuário eu quero poder visualizar os detalhes da passagem (preço, data, horário e tipo) para avaliar se atende às minhas necessidades.
 - Justificativa INVEST:
 - **Independent:** Pode ser desenvolvida após a busca.
 - **Negotiable:** Informações exibidas podem ser ajustadas.
 - **Valuable:** Permite decisão informada antes da compra.
 - **Estimable:** Escopo claro (exibição de dados).
 - **Small:** Apenas visualização de detalhes.
 - **Testable:** Pode-se validar se os dados exibidos estão corretos.
 - Justificativa Moscow:
 - O usuário precisa visualizar informações completas antes de comprar. Sem detalhes (preço, horário, tipo), não é possível tomar decisões sobre a passagem. Isso impacta diretamente na confiabilidade do sistema.
- Como usuário eu quero poder adicionar passagens ao carrinho para revisar antes de finalizar a compra.
 - Justificativa INVEST:
 - **Independent:** Funcionalidade isolada dentro do fluxo de compra.
 - **Negotiable:** Pode evoluir para múltiplas regras futuras.
 - **Valuable:** Permite revisão antes do pagamento.
 - **Estimable:** Lógica clara de inclusão no carrinho.
 - **Small:** Apenas ação de adicionar item.
 - **Testable:** Verificar se o item aparece no carrinho após inclusão.
 - Justificativa Moscow:
 - O carrinho é parte do fluxo de compra. Sem ele, não há etapa intermediária entre seleção e pagamento. Ele permite revisão antes da finalização.
- Como usuário eu quero poder visualizar as passagens adicionadas ao carrinho para conferir minhas escolhas.
 - Justificativa INVEST:
 - **Independent:** Pode ser implementada separadamente.
 - **Negotiable:** Pode incluir melhorias futuras (ex: cálculo automático).
 - **Valuable:** Permite conferência antes da compra.
 - **Estimable:** Exibição simples de itens selecionados.
 - **Small:** Apenas visualização.
 - **Testable:** Validar se exibe corretamente os itens adicionados.
 - Justificativa Moscow:
 - Sem visualizar o carrinho, o usuário não consegue confirmar o que está comprando. Isso compromete a sua decisão e pode gerar erros de compra.
- Como usuário eu quero poder remover passagens do carrinho para ajustar minha compra antes do pagamento.
 - Justificativa INVEST:
 - **Independent:** Funciona isoladamente dentro do carrinho.

- **Negotiable:** Pode incluir confirmação de remoção.
 - **Valuable:** Permite correção antes do pagamento.
 - **Estimable:** Implementação simples de exclusão.
 - **Small:** Ação específica de remoção.
 - **Testable:** Validar se o item é removido corretamente.
- Justificativa Moscow:
 - Por mais que seja importante para experiência do usuário, não impede a compra.
- Como usuário eu quero poder pagar minhas passagens com cartão de crédito ou Pix para escolher o método de pagamento mais conveniente.
 - Justificativa INVEST:
 - **Independent:** Pode ser implementado após carrinho.
 - **Negotiable:** Pode incluir novos métodos no futuro.
 - **Valuable:** Concretiza a compra e gera receita.
 - **Estimable:** Escopo definido (integração com meio de pagamento).
 - **Small:** Abrange apenas finalização da transação.
 - **Testable:** Testável com pagamento aprovado e recusado.
 - Justificativa Moscow:
 - A funcionalidade de pagamento é essencial, pois é nela que ocorre a conversão e a geração de receita. Sem pagamento, o fluxo de compra não é concluído e o sistema não cumpre seu objetivo principal.
- Como usuário eu quero receber uma confirmação após o pagamento aprovado para garantir que a compra foi concluída com sucesso.
 - Justificativa INVEST:
 - **Independent:** Pode ser desenvolvida após pagamento.
 - **Negotiable:** A forma da confirmação pode variar (tela, e-mail).
 - **Valuable:** Gera confiança e segurança ao usuário.
 - **Estimable:** Implementação simples de retorno visual.
 - **Small:** Apenas confirmação da transação.
 - **Testable:** Validar exibição após pagamento aprovado.
 - Justificativa Moscow:
 - A confirmação garante segurança jurídica e confiança do usuário. Sem ela, o cliente não teria garantia de que a compra foi concluída.
- Como usuário eu quero ter uma área do cliente onde eu possa visualizar minhas passagens compradas para acompanhar minhas viagens.
 - Justificativa INVEST:
 - **Independent:** Pode ser entregue após MVP.
 - **Negotiable:** Pode evoluir com novas funcionalidades.
 - **Valuable:** Permite acompanhar passagens compradas.
 - **Estimable:** Escopo claro (listagem de compras).
 - **Small:** Apenas visualização do histórico.
 - **Testable:** Validar se exibe compras corretamente.
 - Justificativa Moscow:

- Agrega valor e melhora a experiência pós-compra, mas o sistema ainda poderia funcionar enviando confirmação por e-mail na versão inicial.
- Como usuário eu quero poder editar meus dados cadastrais para manter minhas informações atualizadas.
 - Justificativa INVEST:
 - **Independent:** Funcionalidade isolada.
 - **Negotiable:** Pode incluir novos campos futuramente.
 - **Valuable:** Permite atualização de informações.
 - **Estimable:** Implementação simples de edição.
 - **Small:** Apenas atualização de dados.
 - **Testable:** Validar se alterações são salvas corretamente.
 - Justificativa Moscow:
 - É uma funcionalidade importante para manutenção de dados, porém não impacta diretamente no fluxo principal de busca e compra.

Requisitos não funcionais:

Performance:

- RNF01: O tempo de resposta para o cadastro do usuário deve ser de até 5 segundos
- RNF02: O tempo de resposta de busca de uma passagem deve ser menor que 5 segundos
- RNF03: O tempo de carregamento das páginas de cadastro, login e pagamento deve ser de até 2 segundos
- RNF04: O sistema deve ser capaz de lidar com 1000 usuários simultâneos realizando buscas ou transações sem comprometer a performance

Segurança:

- RNF05: Os usuários serão identificados com login e senha
- RNF06: As informações de pagamento, senha e cadastro serão criptografadas
- RNF07: O sistema deve ser protegido contra ataques externos

Disponibilidade:

- RNF08: O sistema deve estar disponível 99.9% do tempo

- RNF09: Se houver uma falha no sistema, esta deve ser resolvida em até 3 horas pelo time de desenvolvimento

Usabilidade:

- RNF10: O sistema deve apresentar uma interface 100% responsiva
- RNF11: O sistema deve apresentar feedbacks para pagamentos e cadastros finalizados

Operacional:

- RNF12: O sistema deve ser capaz de crescer facilmente em termos de capacidade de processamento e armazenamento.

Interoperabilidade:

- RNF13: O sistema deve ter integração com APIs externas para a parte de pagamentos.

Critérios de aceite:

Como usuário eu quero poder me cadastrar informando nome, e-mail e senha para criar minha conta na aplicação.

Funcionais:

Cenário: Cadastro com dados válidos

Dado que o usuário informa nome, e-mail válido e senha válida

Quando clicar em "Cadastrar"

Então o sistema deve criar a conta e exibir mensagem de sucesso

Cenário: Cadastro com e-mail já existente

Dado que o usuário informa um e-mail já cadastrado

Quando clicar em "Cadastrar"

Então o sistema deve exibir mensagem informando que o e-mail já está em uso

Não funcional:

Cenário: segurança no cadastro

Dado que o usuário realiza o cadastro

Quando submeter os dados

Então o sistema deve armazenar a senha de forma criptografada

Como usuário eu quero poder logar na aplicação utilizando e-mail e senha cadastrados para acessar minha conta.

Funcionais:

Cenário: Login com dados válidos

Dado que o usuário informa e-mail e senha corretos

Quando clicar em "Entrar"

Então o sistema deve permitir acesso à conta

Cenário: Login com dados inválidos

Dado que o usuário informa e-mail ou senha incorretos

Quando clicar em "Entrar"

Então o sistema deve exibir mensagem de erro

Não funcional:

Cenário: segurança no login

Dado que o usuário realiza login

Quando submeter os dados

Então o sistema deve trafegar as informações de forma criptografada

Como usuário eu quero poder buscar passagens informando origem e destino para visualizar viagens disponíveis.

Funcionais:

Cenário: Busca com resultados

Dado que o usuário informa origem e destino válidos

Quando clicar em "Buscar"

Então o sistema deve exibir as passagens disponíveis

Cenário: Busca sem resultados

Dado que não existem passagens para a origem e destino informados

Quando clicar em "Buscar"

Então o sistema deve exibir mensagem informando que não há viagens disponíveis

Não funcional:

Cenário: performance na busca

Dado que o usuário realiza uma busca

Quando clicar em "Buscar"

Então o sistema deve responder em até 5 segundos

Como usuário eu quero poder filtrar passagens por data para escolher quando viajar.

Funcionais:

Cenário: Filtro com resultados

Dado que o usuário seleciona uma data válida

Quando aplicar o filtro

Então o sistema deve exibir passagens disponíveis para a data selecionada

Cenário: Filtro sem resultados

Dado que não existem passagens para a data selecionada

Quando aplicar o filtro

Então o sistema deve informar que não há viagens disponíveis

Não funcional:

Cenário: performance no filtro por data

Dado que o usuário aplica o filtro

Quando o sistema processar a solicitação

Então o tempo de resposta deve ser menor que 5 segundos

Como usuário eu quero poder filtrar passagens por tipo (aérea ou marítima) para encontrar a opção desejada.

Funcionais:

Cenário: Filtro por tipo aéreo

Dado que o usuário seleciona o tipo "Aérea"

Quando aplicar o filtro

Então o sistema deve exibir apenas passagens aéreas

Cenário: Filtro por tipo marítimo

Dado que o usuário seleciona o tipo "Marítima"

Quando aplicar o filtro

Então o sistema deve exibir apenas passagens marítimas

Não funcional:

Cenário: performance no filtro por tipo

Dado que o usuário aplica o filtro

Quando o sistema processar a solicitação

Então o tempo de resposta deve ser menor que 5 segundos

Como usuário eu quero poder visualizar os detalhes da passagem (preço, data, horário e tipo) para avaliar se atende às minhas necessidades.

Funcionais:

Cenário: Visualizar detalhes da passagem

Dado que o usuário seleciona uma passagem

Quando clicar em "Ver detalhes"

Então o sistema deve exibir preço, data, horário e tipo da passagem

Cenário: Dados consistentes

Dado que o usuário visualiza os detalhes

Quando as informações forem exibidas

Então os dados devem corresponder à passagem selecionada

Não funcional:

Cenário: carregamento dos detalhes

Dado que o usuário solicita os detalhes

Quando a página carregar

Então o tempo de carregamento deve ser de até 2 segundos

Como usuário eu quero poder adicionar passagens ao carrinho para revisar antes de finalizar a compra.

Funcionais:

Cenário: Adicionar passagem ao carrinho

Dado que o usuário seleciona uma passagem

Quando clicar em "Adicionar ao carrinho"

Então o sistema deve incluir a passagem no carrinho

Cenário: Atualização do valor total

Dado que uma passagem foi adicionada

Quando o carrinho for atualizado

Então o sistema deve recalcular o valor total corretamente

Não funcional:

Cenário: performance ao adicionar ao carrinho

Dado que o usuário adiciona uma passagem

Quando confirmar a ação

Então o sistema deve responder em até 2 segundos

Como usuário eu quero poder visualizar as passagens adicionadas ao carrinho para conferir minhas escolhas.

Funcionais:

Cenário: Exibir itens do carrinho

Dado que o usuário possui passagens no carrinho

Quando clicar no ícone do "carrinho"

Então o sistema deve listar todas as passagens adicionadas

Cenário: Exibir valor total

Dado que o carrinho possui passagens

Quando o usuário visualizar o carrinho

Então o sistema deve exibir o valor total da compra

Não funcional:

Cenário: carregamento do carrinho

Dado que o usuário acessa o carrinho

Quando a página carregar

Então o tempo de carregamento deve ser de até 2 segundos

Como usuário eu quero poder remover passagens do carrinho para ajustar minha compra antes do pagamento.

Funcionais:

Cenário: Remover passagem do carrinho

Dado que o usuário possui passagens no carrinho

Quando clicar em "Remover"

Então o sistema deve excluir a passagem selecionada

Cenário: Atualização do valor total após remoção

Dado que uma passagem foi removida

Quando o carrinho for atualizado

Então o sistema deve recalcular o valor total corretamente

Não funcional:

Cenário: performance ao remover item

Dado que o usuário remove uma passagem

Quando confirmar a ação

Então o sistema deve responder em até 2 segundos

Como usuário eu quero poder pagar minhas passagens com cartão de crédito ou Pix para escolher o método de pagamento mais conveniente.

Funcionais:

Cenário: Pagamento com cartão de crédito

Dado que o usuário escolhe pagar com cartão

Quando informar os dados válidos e confirmar

Então o sistema deve processar o pagamento

Cenário: Pagamento com Pix

Dado que o usuário escolhe pagar com Pix

Quando confirmar o pagamento

Então o sistema deve gerar a chave ou QR Code para pagamento

Não funcional:

Cenário: segurança no pagamento

Dado que o usuário informa dados de pagamento

Quando submeter as informações

Então o sistema deve criptografar os dados e utilizar integração segura com API externa

Como usuário eu quero receber uma confirmação após o pagamento aprovado para garantir que a compra foi concluída com sucesso.

Funcionais:

Cenário: Pagamento aprovado

Dado que o pagamento foi aprovado

Quando o processamento for concluído

Então o sistema deve exibir mensagem de confirmação

Cenário: Registro da compra

Dado que o pagamento foi aprovado

Quando a transação for finalizada

Então o sistema deve registrar a passagem na área do cliente

Não funcional:

Cenário: tempo de confirmação

Dado que o pagamento foi aprovado

Quando o sistema retornar o status

Então a confirmação deve ser exibida em até 2 segundos

Como usuário eu quero ter uma área do cliente onde eu possa visualizar minhas passagens compradas para acompanhar minhas viagens.

Funcionais:

Cenário: Visualizar passagens compradas

Dado que o usuário realizou compras

Quando acessar a área do cliente

Então o sistema deve listar todas as passagens adquiridas

Cenário: Visualizar detalhes da compra

Dado que o usuário seleciona uma passagem comprada

Quando visualizar os detalhes

Então o sistema deve exibir informações completas da viagem

Não funcional:

Cenário: carregamento da área do cliente

Dado que o usuário acessa a área do cliente

Quando a página carregar

Então o tempo de carregamento deve ser de até 2 segundos

Como usuário eu quero poder editar meus dados cadastrais para manter minhas informações atualizadas.

Funcionais:

Cenário: Alterar dados com sucesso

Dado que o usuário está autenticado

Quando editar nome ou e-mail e salvar

Então o sistema deve atualizar as informações

Cenário: E-mail já utilizado

Dado que o usuário informa um e-mail já cadastrado

Quando tentar salvar a alteração

Então o sistema deve exibir mensagem de erro

Não funcional:

Cenário: segurança na edição de dados

Dado que o usuário altera seus dados

Quando salvar as informações

Então o sistema deve armazenar os dados de forma criptografada quando aplicável