



UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA - SEMESTRE SELETIVO 2025/1
PROGRAMAÇÃO ORIENTADA A OBJETOS - DOCENTE ELIAS BATISTA
FERREIRA

| | |
|---------------------|------------------------------|
| | Grupo |
| Integrante 1 | JÚLIA DE SOUZA NASCIMENTO |
| Integrante 2 | ANA LUÍSA PEREIRA DOS SANTOS |
| Integrante 3 | GABRIEL RODRIGUES DA SILVA |

PRIMEIRA ENTREGA:

a) Definição do Problema e do Escopo do Trabalho.

Definição do Problema:

Atualmente, o setor de bagagens de uma companhia aérea realiza a gestão dos processos relacionados à *danificação, extravio e recuperação de bagagens* de forma manual, utilizando papéis que posteriormente são escaneados e armazenados na nuvem. Esse processo é moroso, suscetível a erros, demanda grande esforço humano e dificulta a organização e o rastreamento eficiente das informações.

Proposta de Solução:

Desenvolver um sistema automatizado, preferencialmente em formato de aplicativo móvel, que permita o registro e organização digital desses processos, integrando o uso da câmera do dispositivo para captura de imagens e facilitando, assim, o armazenamento e classificação automática dos documentos.

Escopo do Trabalho:

O sistema irá:

- Permitir a captura de imagens dos documentos via câmera ou carregamento de arquivos de imagens dos documentos;
- Classificar automaticamente os processos nas categorias adequadas: *Danificação de Bagagem, Extravio de Bagagem, Itens Esquecidos em Avião,*

Recibo de Conserto de Bagagem, Recibo de Entrega de Bagagem Extraviada, Recibo de Indenização em Milhas e Recibo de Item Esquecido em Avião;

- Armazenar os documentos, organizados de forma estruturada para posterior consulta e rastreamento.

b) Lista de requisitos (o que o sistema deve fazer)

Requisitos Funcionais (RF):

O sistema deve:

- **RF1:** Permitir o cadastro de processos de bagagem (Danificação, Extravio, Itens Esquecidos).
- **RF2:** Armazenar informações básicas do processo (base, número do processo, data de abertura).
- **RF3:** Permitir a edição das informações do processo (exceto a Base e Número do Processo).
- **RF4:** Permitir a remoção de um processo iniciado.
- **RF5:** Exibir informações completas do processo, incluindo imagens.
- **RF6:** O sistema deve gerar automaticamente um identificador único e sequencial para cada novo processo e recibo cadastrado.
- **RF7:** Aceitar upload de imagens já existentes (JPG, PNG, PDF).
- **RF8:** Exibir miniatura da imagem antes do armazenamento para pré-visualização.
- **RF9:** Renomear arquivos de imagem para padronização.
- **RF10:** Armazenar documentos em um repositório local com metadados associados
- **RF11:** Identificar e classificar automaticamente os processos nas categorias: Danificação de Bagagem. Extravio de Bagagem, Itens Esquecidos em Avião, Recibos associados (Conserto, Indenização, Entrega, etc.).
- **RF12:** Permitir a associação de recibos a processos existentes.
- **RF13:** Armazenar informações específicas de cada tipo de recibo (ex.: quantidade de milhas).
- **RF14:** Permitir a edição dos dados do recibo.

- **RF15:** Listar todos os processos cadastrados, com possibilidade de filtros por Base e Número de Processo
- **RF16:** Buscar um processo específico por base + número do processo.
- **RF17:** Garantir que cada processo possa estar associado a no máximo um recibo.
- **RF18:** Garantir que cada recibo esteja vinculado a exatamente um processo.

Requisitos Não Funcionais (RNF):

- **RNF1:** Interface intuitiva para facilitar o registro e consulta de processos.
- **RNF2:** Interface responsiva..
- **RNF3:** Tempo de resposta ≤ 3 segundos para consultas.
- **RNF4:** Armazenamento seguro das imagens e metadados.
- **RNF5:** Armazenar até 20.000 processos sem degradação de performance.
- **RNF6:** Desenvolvimento em Java para desktop.
- **RNF7:** Funcionar em sistemas operacionais Windows e Linux.

c) Casos de uso com a representação dos atores e requisitos

Com o objetivo de modelar graficamente o comportamento funcional do sistema automatizado proposto, foi feito um diagrama de casos de uso que descreve o que o sistema faz, quem interage com ele e como essas interações acontecem.

O modelo contém um único ator, o funcionário do setor de bagagens, que vai ser o responsável por todas as interações com o sistema. Para facilitar a visualização e o entendimento, os casos de uso foram divididos em três grandes categorias funcionais: Gerenciar Processos, Imagens e Documentos e Classificação e Recibos.

1. Gerenciar Processos

Essas funcionalidades estão relacionadas à criação e manutenção dos processos de bagagem no sistema:

- **Cadastrar novo processo**
Permite ao funcionário registrar um novo caso de bagagem (danificada, extraviada ou esquecida). Este caso será posteriormente associado a

documentos e recibos.

- **Editar processo**
Permite alterar informações básicas do processo (como base ou número), conforme definido nos requisitos RF3.
- **Remover processo**
Possibilita excluir um processo iniciado incorretamente (RF4).
- **Visualizar processo**
Exibe todas as informações do processo, incluindo os documentos e recibos vinculados (RF5).
- **Listar processos com filtros**
Permite a navegação por todos os processos, aplicando filtros como tipo, base ou data (RF15).
- **Buscar processo por base e número**
Busca direta por processos através da combinação única base + número do processo (RF16).

2. Imagens e Documentos

Esses casos de uso tratam da captura, upload e gerenciamento das imagens/documentos associados a cada processo:

- **Capturar imagem via câmera**
Permite tirar uma foto do documento diretamente pelo dispositivo (RF6).
- **Fazer upload de imagem**
Aceita imagens existentes nos formatos JPG, PNG e PDF (RF7).
- **Visualizar miniatura**
Mostra uma pré-visualização da imagem antes de armazenar (RF8).
- **Renomear arquivo**
Permite padronizar os nomes dos arquivos para manter organização (RF9).
- **Armazenar documento**
Salva os documentos e metadados no repositório (local ou nuvem) (RF10).

3. Classificação e Recibos

Funcionalidades que classificam o tipo do processo e vinculam recibos:

- **Classificar automaticamente**

O sistema identifica o tipo de processo com base no conteúdo/documento inserido (RF11).

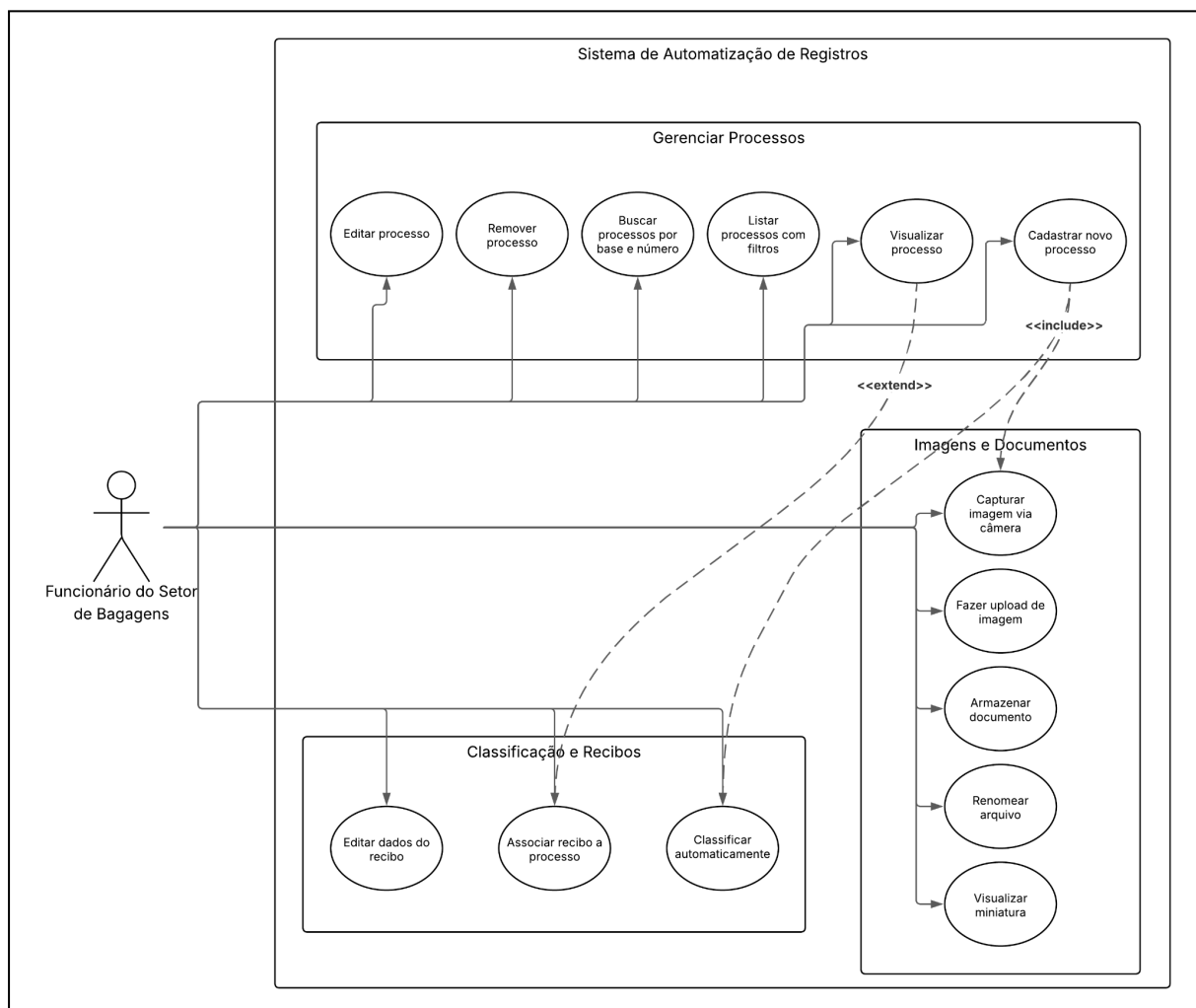
- **Associar recibo a processo**

Permite vincular recibos aos respectivos processos (danificação, extravio ou item esquecido) (RF12, RF17, RF18).

- **Editar dados do recibo**

Permite ajustar os dados do recibo, como data de assinatura ou valores (RF14).

Segue-se abaixo o diagrama de casos de uso que modela todos os itens mencionados:



d) Classes conceituais, representando o problema, com relacionamentos entre elas (pelo menos 4 classes conceituais).

Classe abstrata Processo

Propósito:

Representa de forma genérica qualquer processo de bagagem que será escaneado, organizado e armazenado no sistema.

Atributos (privados):

- **base: String**
Sigla do aeroporto em que o processo foi iniciado (por exemplo, "GYN" ou "GRU").
- **numeroProcesso: String**
Número de identificação do processo dentro daquela base. Como a combinação "base + numeroProcesso" é única, o mesmo número em bases diferentes pode existir (por exemplo, "GYN12345" e "GRU12345" são processos distintos).
- **dataAbertura: Date**
Data em que a imagem do documento foi capturada e cadastrado o processo no sistema.

Métodos públicos:

1. **capturarImagem(): String**
 - Aciona a câmera do dispositivo e retorna o caminho/URL da foto capturada.
2. **renomearDocumento(novoNome: String): void**
 - Renomeia o arquivo de imagem (por exemplo, para padronizar o nome no repositório).
3. **editarInformacoes(novosDados: Map<String, Object>): void**
 - Permite atualizar qualquer atributo do processo (até **base** e **numeroProcesso**, que são a chave de identificação caso tenham sido inseridas incorretamente).
4. **armazenarDocumento(): void**
 - Salva no repositório (nuvem ou local) o arquivo de imagem e os metadados mínimos (**base**, **numeroProcesso** e **dataAbertura**).

5. `listarDocumentos(): List<Processo>`

- Retorna todos os processos armazenados, podendo aceitar filtros simples (por exemplo, listar apenas processos de “Extravio” ou de uma determinada “base”).

6. `buscarDocumento(base: String, numeroProcesso: String): Processo`

- Retorna o processo específico — com seus atributos e caminho de imagem — identificado pela combinação de “base + numeroProcesso”.

Relacionamentos:

- **Herança (generalização):**
 - `DanificacaoBagagem` estende **Processo**
 - `ExtravioBagagem` estende **Processo**
 - `ItemEsquecidoAviao` estende **Processo**
- **Associação (1 → 1) com Recibo:**
 - Cada instância de **Processo** pode estar referenciada por exatamente um objeto **Recibo** (conforme modelado no diagrama).

Subclasses de Processo

Cada uma representa um tipo específico de processo de bagagem. Todas herdam os atributos e métodos de `Processo`.

DanificacaoBagagem

- **Descrição:**

Representa o processo em que a bagagem foi danificada.
- **Atributo específico (público):**
 - `etiquetaBagagemDanificada: String`

Número da etiqueta física da mala que sofreu dano (por exemplo, “BG-00012345”).
- **Relacionamentos:**
 - Herda todos os atributos e métodos de **Processo**.
 - Pode gerar um **ReciboConsertoBagagem** ou um **ReciboIndenizacaoMilhas** (cada um deles referencia, via atributos `base` e

`numeroProcesso`, esta instância de `DanificacaoBagagem`).

ExtravioBagagem

- **Descrição:**
Representa o processo em que a bagagem foi identificada como extraviada.
- **Atributo específico (público):**
 - `etiquetaBagagemExtraviada: String`
Número da etiqueta da mala extraviada (por exemplo, "BG-00098765").
- **Relacionamentos:**
 - Herda todos os atributos e métodos de **Processo**.
 - Pode gerar um único **ReciboEntregaBagagemExtraviada**, que referencia este processo de extravio.

ItemEsquecidoAviao

- **Descrição:**
Representa o processo em que um objeto foi esquecido dentro da aeronave e precisa ser devolvido ao passageiro.
- **Atributo específico (público):**
 - `numeroVoo: String`
Código do voo no qual o item foi deixado (por exemplo, "G3 1230").
- **Relacionamentos:**
 - Herda todos os atributos e métodos de **Processo**.
 - Pode gerar um único **ReciboItemEsquecidoAviao**, que referencia este processo de item esquecido.

Classe abstrata Recibo

Propósito:

Representa, de maneira genérica, qualquer tipo de comprovante (recibo) vinculado a um processo de bagagem.

Atributos (privados):

- **base: String**
Sigla do aeroporto, igual ao **base** do processo associado.
- **numeroProcesso: String**
Mesmo número que identifica o processo ao qual este recibo pertence.
- **dataAssinatura: Date**
Data em que o recibo foi assinado ou registrado no sistema (pode coincidir com o momento do upload).
- **processoAssociado: Processo**
Referência ao objeto **Processo** (DanificacaoBagagem, ExtravioBagagem ou ItemEsquecidoAviao) a que este recibo se refere.

Métodos públicos:

1. **associarAoProcesso(p: Processo): void**
 - Estabelece a ligação deste recibo a um objeto **p** de tipo **Processo**, copiando **p.base** e **p.numeroProcesso** para os atributos de Recibo.
2. **editarDadosRecibo(novosDados: Map<String, Object>): void**
 - Permite alterar qualquer atributo do recibo (por exemplo, corrigir a data de assinatura, se necessário).

Relacionamentos:

- **Herança (generalização):**
 - **ReciboConsertoBagagem** estende **Recibo**
 - **ReciboIndenizacaoMilhas** estende **Recibo**
 - **ReciboEntregaBagagemExtraviada** estende **Recibo**
 - **ReciboItemEsquecidoAviao** estende **Recibo**
- **Associação (1 → 1) com Processo:**
 - Cada instância de **Recibo** mantém um ponteiro (**processoAssociado**) para exatamente um **Processo**.

Subclasses de Recibo

Cada classe representa um tipo específico de comprovante gerado pelo processo correspondente. Todas herdam atributos e métodos de **Recibo**, mas adicionam

apenas o atributo mínimo necessário para distinguir o tipo — o restante dos detalhes fica contido na própria imagem/PDF que será armazenado.

ReciboConsertoBagagem

- **Descrição:**
Comprovante de que a bagagem danificada foi levada à oficina ou reparada.
- **Atributo específico (público):**
 - **entregaOuRetiradaEmAeroporto: String**
Indica se o conserto foi entregue ou retirado diretamente no aeroporto (por exemplo, “Retirada em GRU” ou “Entregue na oficina central”).
- **Relacionamento específico:**
 - Herda **base** e **numeroProcesso** do processo associável (que deve ser instância de **DanificacaoBagagem**).

ReciboIndenizacaoMilhas

- **Descrição:**
Comprovante de indenização concedida em milhas ao passageiro, por danificação irreparável.
- **Atributo específico (público):**
 - **quantidadeMilhas: int**
Número de milhas concedidas como forma de indenização.
- **Relacionamento específico:**
 - Herda **base** e **numeroProcesso** do processo associável (instância de **DanificacaoBagagem**).

ReciboEntregaBagagemExtraviada

- **Descrição:**
Comprovante que atesta a devolução da bagagem extraviada ao passageiro.
- **Atributo específico (público):**

- **entregaOuRetiradaEmAeroporto: String**
Indica onde ocorreu a entrega ou retirada da mala (por exemplo, “Entregue no balcão GYN” ou “Retirada no setor de achados de GRU”).
- **Relacionamento específico:**
 - Herda **base** e **numeroProcesso** do processo associável (instância de **ExtravioBagagem**).

ReciboItemEsquecidoAviao

- **Descrição:**
Comprovante de que um item esquecido na aeronave foi devolvido ao passageiro.
- **Atributo específico (público):**
 - **documentoIdentificacaoClienteRetirada: String**
Número do documento de identificação apresentado pelo passageiro no momento da retirada (por exemplo, RG ou CPF).
- **Relacionamento específico:**
 - Herda **base** e **numeroProcesso** do processo associável (instância de **ItemEsquecidoAviao**).

Mapa de Relacionamentos entre as Classes

Herança:

- **Processo** (abstrata)
 - ↓ DanificacaoBagagem
 - ↓ ExtravioBagagem
 - ↓ ItemEsquecidoAviao
- **Recibo** (abstrata)
 - ↓ ReciboConsertoBagagem
 - ↓ ReciboIndenizacaoMilhas
 - ↓ ReciboEntregaBagagemExtraviada
 - ↓ ReciboItemEsquecidoAviao

Associações principais:

- Cada **Processo** (seja de danificação, extravio ou item esquecido) — 1 → 1 —→ **Recibo** (cada recibo refere exatamente um processo).
- Na prática, isso significa:
 - Uma instância de **DanificacaoBagagem** pode ter, por exemplo, um **ReciboConsertoBagagem** ou um **ReciboIndenizacaoMilhas**.
 - Uma instância de **ExtravioBagagem** pode ter um **ReciboEntregaBagagemExtraviada**.
 - Uma instância de **ItemEsquecidoAviao** pode ter um **ReciboItemEsquecidoAviao**.

e) Diagrama de classes mostrando as principais classes, atributos e métodos e seus relacionamentos.

Diagrama:

Código Plantuml:

[PlantUML Editor](#)

