Grupo: Joao Paulo Fortes – **10419607** / Felipe Teixeira – **10417483** /

Gabriel Boock – **10175655** / Murillo Mattar **- 10418329** / Gustavo Verlante **- 10418329**

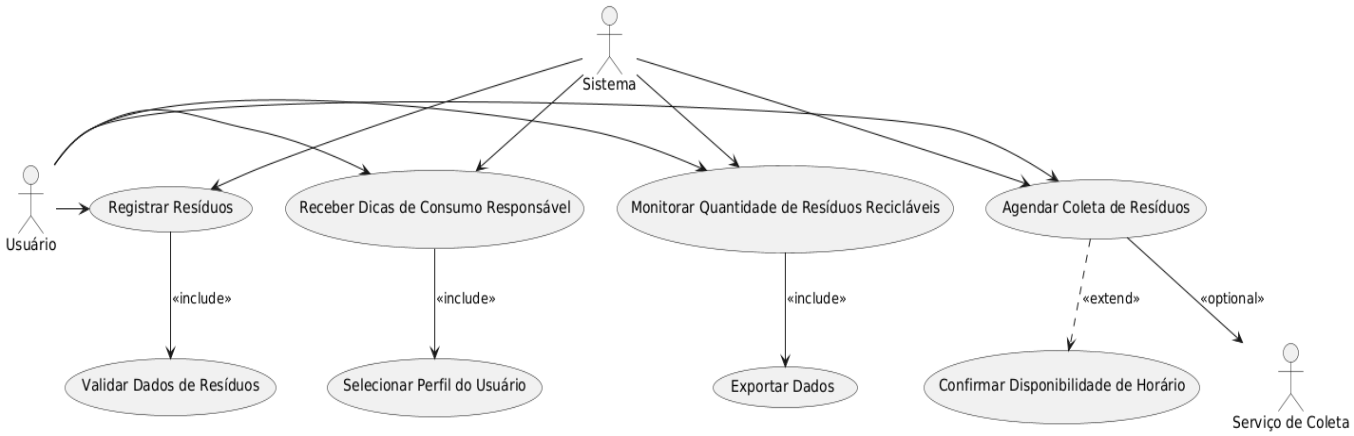
1. Tema do Projeto:

O sistema de Gerenciamento de Resíduos e Reciclagem Doméstica ajudará os usuários a registrar corretamente os tipos de resíduos gerados, incentivando o descarte adequado e consciente. Ele fornecerá dicas de consumo responsável para reduzir a geração de resíduos e exibirá relatórios sobre a quantidade reciclada, permitindo que os usuários acompanhem seu impacto ambiental. Além disso, o sistema facilitará o agendamento de coletas seletivas, promovendo práticas sustentáveis e apoiando a meta do ODS 12 de reduzir a geração de resíduos por meio de prevenção, reciclagem e reuso. O público-alvo são cidadãos preocupados com o meio ambiente que desejam monitorar e melhorar seus hábitos de descarte e reciclagem.

2. Justificativa:

O consumo responsável e a gestão adequada de resíduos são partes fundamentais para reduzir o impacto ambiental. Com este sistema, os usuários poderão organizar seu consumo de forma mais consciente, acompanhando o volume de resíduos gerados e orientando-se sobre práticas de reciclagem. Isso apoia diretamente a meta do ODS 12, que visa reduzir a geração de resíduos por meio de prevenção, redução, reciclagem e reuso.

3. Diagrama de Casos de Uso



4. Casos de Uso:

Caso de Uso 1: Registrar Resíduos

Objetivo: Permitir que o usuário insira os tipos de resíduos gerados para registro e monitoramento.

Fluxo Normal:

1. O usuário acessa a funcionalidade "Registrar Resíduos" no sistema.
2. O sistema solicita que o usuário informe o tipo de resíduo (plástico, papel, orgânico, etc.) e a quantidade (em kg ou unidades).
3. **(Include)** O sistema executa o fluxo "Validar Dados de Resíduos" para garantir que as informações estão corretas antes do registro.
4. O usuário preenche as informações e clica no botão "Registrar".
5. O sistema valida as informações (verifica se os campos obrigatórios foram preenchidos).
6. O sistema registra os dados de resíduos no banco de dados e confirma o registro ao usuário com uma mensagem de sucesso.
7. O usuário finaliza o processo ou insere novos registros.

Fluxo Alternativo:

- Se o tipo de resíduo informado pelo usuário não estiver na lista permitida, o sistema exibe uma mensagem de erro e apresenta uma lista de materiais aceitos para escolha.
- O usuário pode corrigir as informações e submeter novamente.

Fluxo de Exceção:

- Se houver uma falha de conexão ou erro no banco de dados durante o registro, o sistema exibe uma mensagem de erro ao usuário, sugerindo tentar novamente mais tarde.
- O sistema pode oferecer uma opção para salvar os dados localmente até que a conexão seja restabelecida.

Caso de Uso 2: Receber Dicas de Consumo Responsável

Objetivo: Fornecer dicas ao usuário para melhorar suas práticas de consumo e reduzir a geração de resíduos.

Fluxo Normal:

1. O usuário acessa a funcionalidade "Receber Dicas de Consumo Responsável".
2. **(Include)** O sistema executa o fluxo "Selecionar Perfil do Usuário" para definir dicas personalizadas com base no perfil do usuário.
3. O sistema exibe uma lista de dicas personalizadas com base no tipo de resíduo mais gerado pelo usuário.
4. O usuário pode clicar em uma dica para visualizar detalhes adicionais ou ignorá-las.
5. O sistema permite que o usuário marque as dicas como "Lidas" ou "Interessantes", para referenciá-las posteriormente.

Fluxo Alternativo:

- Se o sistema não encontrar dados suficientes sobre o perfil de consumo do usuário, ele pode exibir dicas genéricas de consumo consciente.

Fluxo de Exceção:

- Se houver um erro no carregamento das dicas (por exemplo, falha no serviço de recomendação), o sistema notifica o usuário e tenta carregar novamente após um intervalo de tempo.

Caso de Uso 3: Monitorar Quantidade de Resíduos Recicláveis

Objetivo: Exibir relatórios detalhados sobre os resíduos reciclados pelo usuário.

Fluxo Normal:

1. O usuário acessa a funcionalidade "Monitorar Quantidade de Resíduos Recicláveis".
2. O sistema exibe um painel com gráficos mostrando a quantidade de resíduos reciclados em um determinado período (diário, semanal, mensal).
3. O usuário pode selecionar diferentes intervalos de tempo e visualizar relatórios específicos.
4. O usuário finaliza a visualização e retorna ao menu principal.

Fluxo Alternativo:

- Se o usuário selecionar um intervalo de tempo para o qual não há dados, o sistema exibe uma mensagem informando que não foram gerados resíduos recicláveis no período.

Fluxo de Exceção:

- Caso ocorra uma falha no carregamento dos gráficos, o sistema exibe uma mensagem de erro e tenta recarregar os dados.

Caso de Uso 4: Agendar Coleta de Resíduos

Objetivo: Permitir que o usuário agende coletas de resíduos recicláveis com base na sua localização.

Fluxo Normal:

1. O usuário acessa a funcionalidade "Agendar Coleta de Resíduos".
2. O sistema solicita que o usuário insira a localização para a coleta e selecione o tipo de resíduo a ser coletado.

3. O usuário escolhe uma data e horário disponíveis para a coleta.
4. **(Extend)** Se o horário desejado não estiver disponível, o sistema executa o fluxo "Confirmar Disponibilidade de Horário" para sugerir datas alternativas.
5. O sistema valida as informações e confirma o agendamento.
6. O usuário recebe uma confirmação e pode visualizar ou editar o agendamento futuramente.

Fluxo Alternativo:

- Se não houver disponibilidade de data/horário, o sistema informa o usuário e sugere datas alternativas para a coleta.

Fluxo de Exceção:

Se houver uma falha de comunicação com o serviço de agendamento, o sistema exibe uma mensagem de erro e sugere que o usuário tente novamente mais tarde

5. Atores

- **Usuário**: Representa o cidadão que utiliza o sistema para registrar resíduos, receber dicas, monitorar a quantidade reciclada e agendar coletas de resíduos recicláveis.
- **Sistema**: Atua na interação com o usuário, processando dados, validando informações e fornecendo respostas às solicitações.
- **Serviço de Coleta** (Opcional): Pode ser incluído caso o sistema interaja diretamente com serviços de coleta externos.

6. Requisitos Iniciais:

Requisitos Funcionais:

1. **Registro de Resíduos**:
O sistema deve permitir o registro de resíduos pelos usuários, incluindo:

- Tipo de resíduo (plástico, papel, orgânico, etc.)
- Quantidade (em kg ou unidades)
- Data e horário do registro

- Validação dos dados antes do registro final, garantindo que todas as informações obrigatórias sejam preenchidas corretamente.

2. Relatórios de Resíduos Gerados:
O sistema deve fornecer relatórios detalhados sobre o volume de resíduos gerados, incluindo:

- Relatórios diários, semanais e mensais
- Gráficos e tabelas comparativas
- Visualização por tipo de resíduo, localização e histórico do usuário.

3. Notificações e Dicas de Consumo Responsável:

O sistema deve enviar notificações personalizadas com dicas sobre consumo responsável, baseadas no histórico de resíduos gerados, como:

- Dicas de redução de consumo
- Sugestões de reciclagem e reutilização
- Dicas personalizadas, dependendo do perfil de resíduos do usuário.

4. Agendamento de Coleta Seletiva:
O sistema deve permitir que o usuário agende coletas seletivas, incluindo:

- Escolha do tipo de resíduo a ser coletado
- Inserção da localização para a coleta
- Escolha da data e horário para a coleta (com validação de disponibilidade de horário).
- Caso o horário desejado não esteja disponível, o sistema deve sugerir horários alternativos.

5. Gráfico de Consumo Consciente:
O sistema deve gerar um gráfico do consumo consciente ao longo do tempo, baseado nos dados de resíduos reciclados, mostrando:

- A evolução da reciclagem de resíduos
- Comparações entre períodos (diário, semanal, mensal)
- Dados sobre o impacto ambiental de suas escolhas de consumo.

Requisitos Não-Funcionais:

1. Acessibilidade Multi-Plataforma:
O sistema deve ser acessível em dispositivos móveis (smartphones, tablets) e desktops, adaptando-se automaticamente ao tipo de dispositivo utilizado.

2. Interface de Usuário Intuitiva:
A interface do sistema deve ser simples, clara e de fácil navegação, considerando usuários de diferentes faixas etárias e habilidades digitais. Deve ser projetada com foco na experiência do usuário, com design acessível e de fácil entendimento.

3. Segurança e Proteção de Dados:
O sistema deve garantir a segurança dos dados dos usuários, incluindo:

- Proteção de informações pessoais, como localização e hábitos de consumo, por meio de criptografia
- Autenticação segura para acesso ao sistema
- Proteção contra acessos não autorizados e vazamentos de dados.

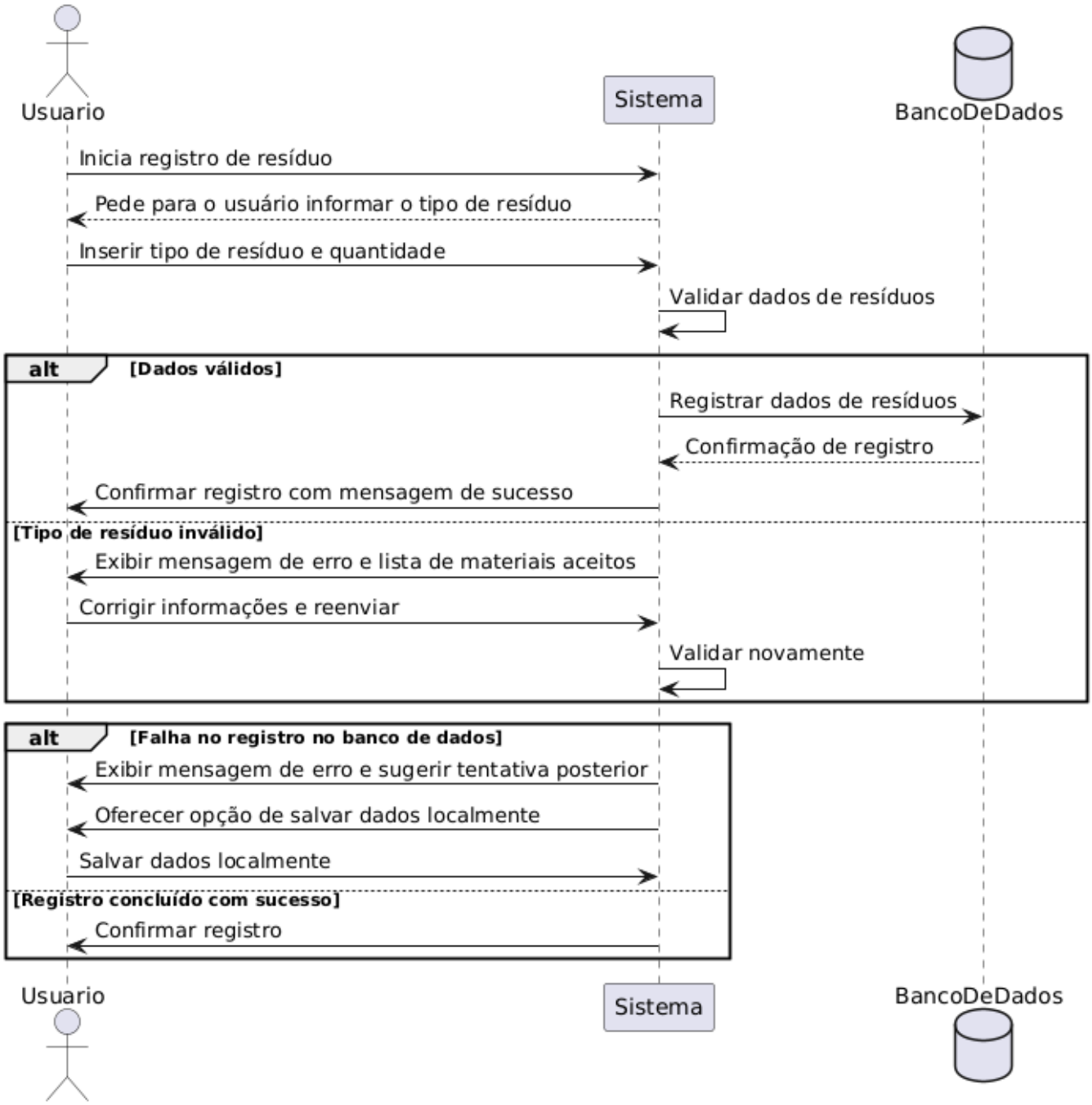
4. Escalabilidade e Desempenho:
O sistema deve ser capaz de lidar com grandes volumes de dados e usuários simultâneos, sem comprometer o desempenho. As operações de registro, consulta e geração de relatórios devem ser rápidas e eficientes.

5. Compatibilidade com Diferentes Idiomas:
O sistema deve oferecer suporte a múltiplos idiomas, permitindo que usuários de diferentes regiões e idiomas possam utilizá-lo de forma eficiente.

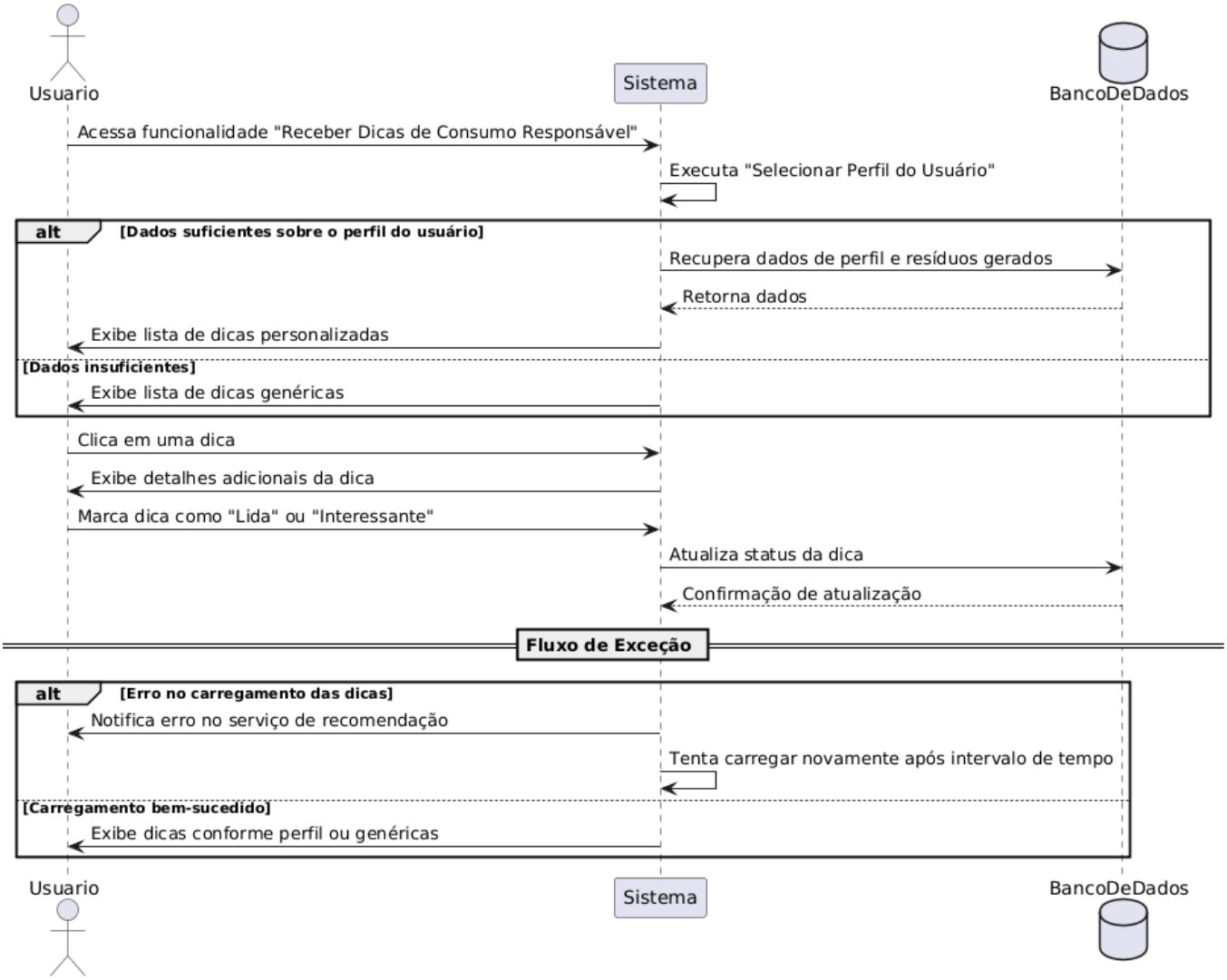
5. Modelos Dinâmicos – Diagramas de Interação

Diagramas de Sequência:

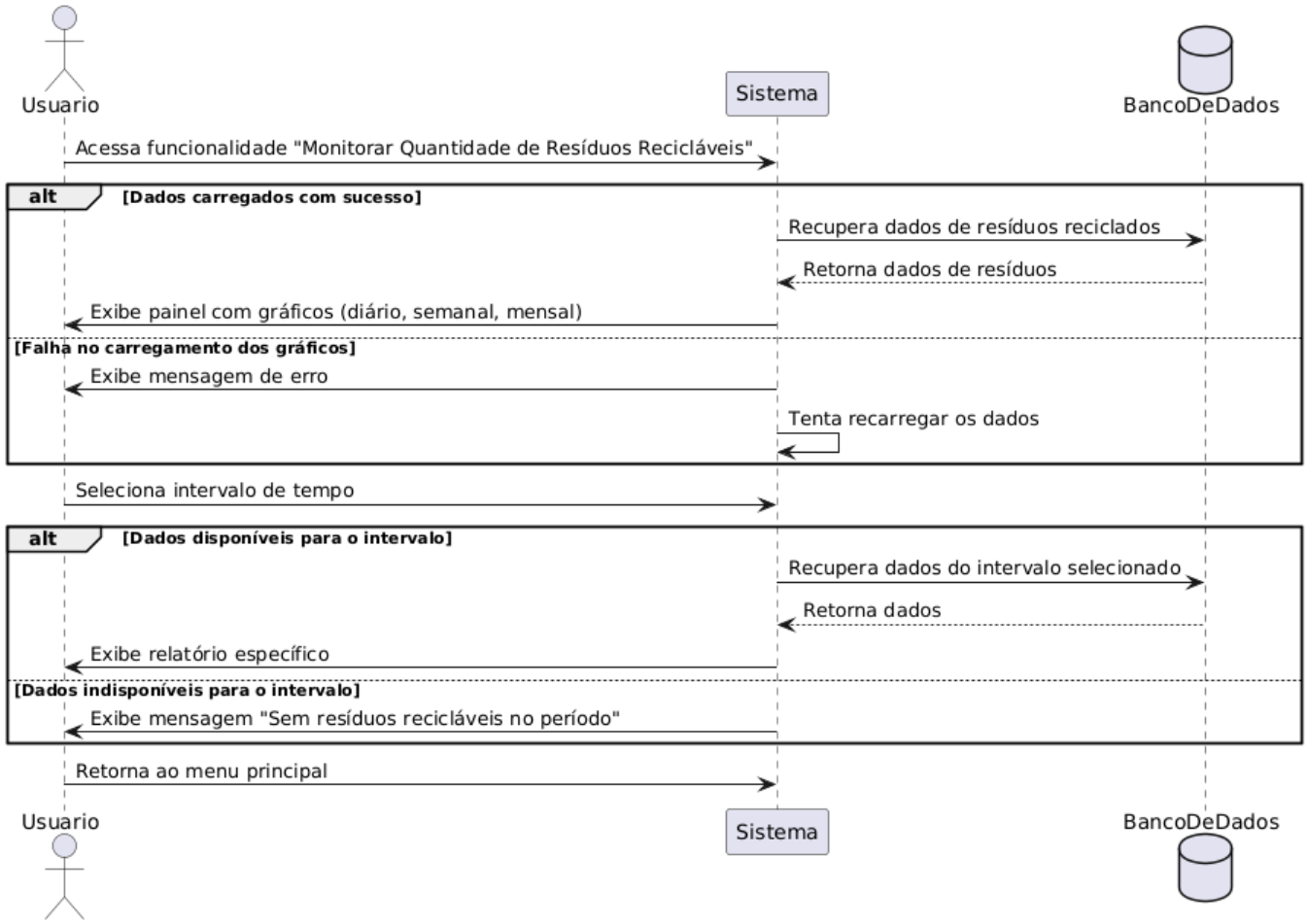
Caso de Uso 1: Registrar Resíduos



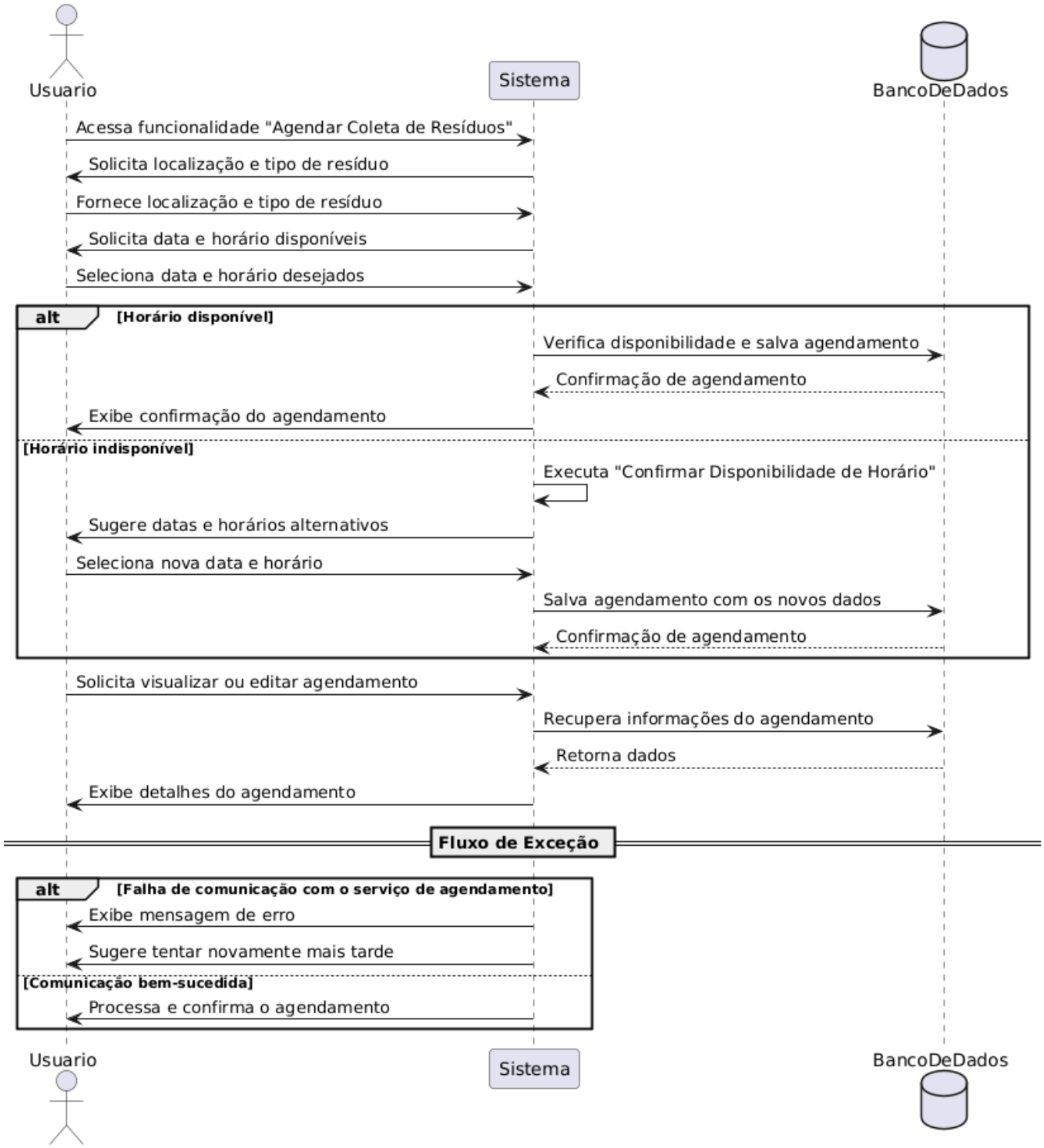
Caso de Uso 2: Receber Dicas de Consumo Responsável



Caso de Uso 3: Monitorar Quantidade de Resíduos Recicláveis

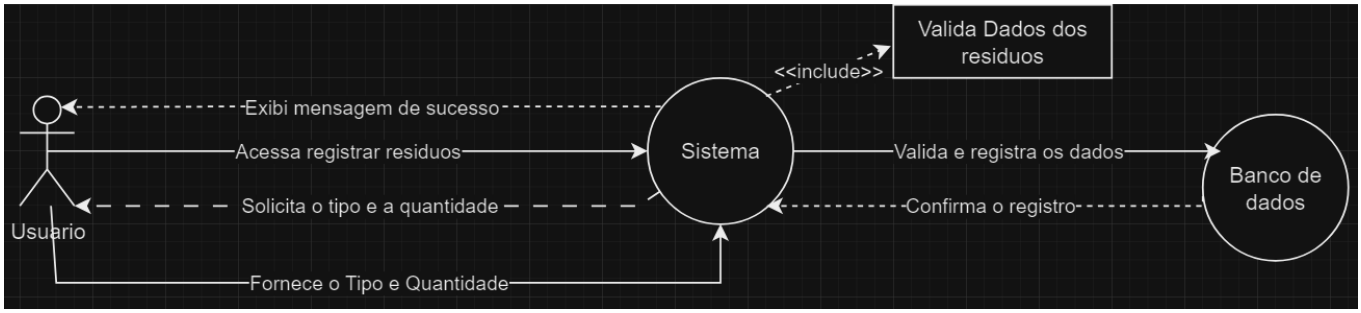


Caso de Uso 4: Agendar Coleta de Resíduos

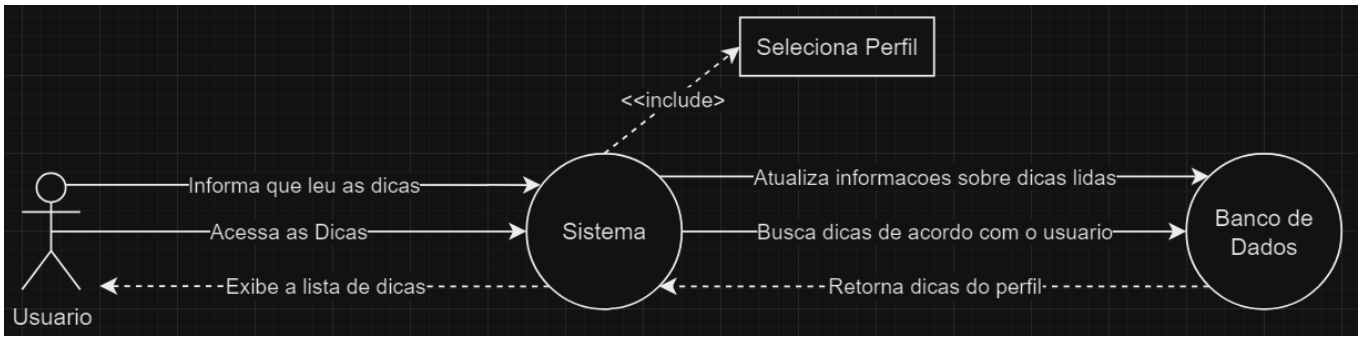


Diagramas de Comunicação:

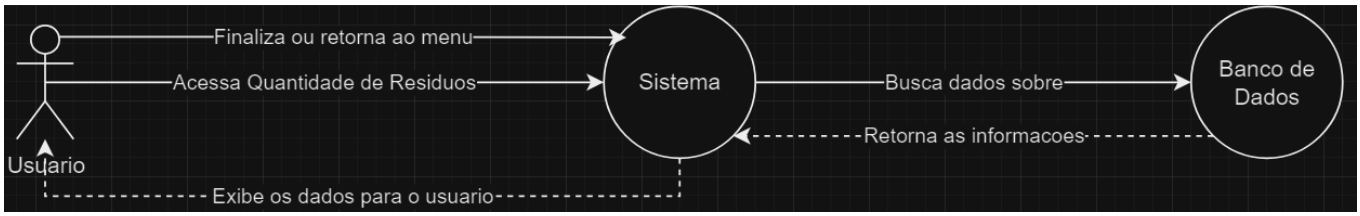
Caso de Uso 1: Registrar Resíduos



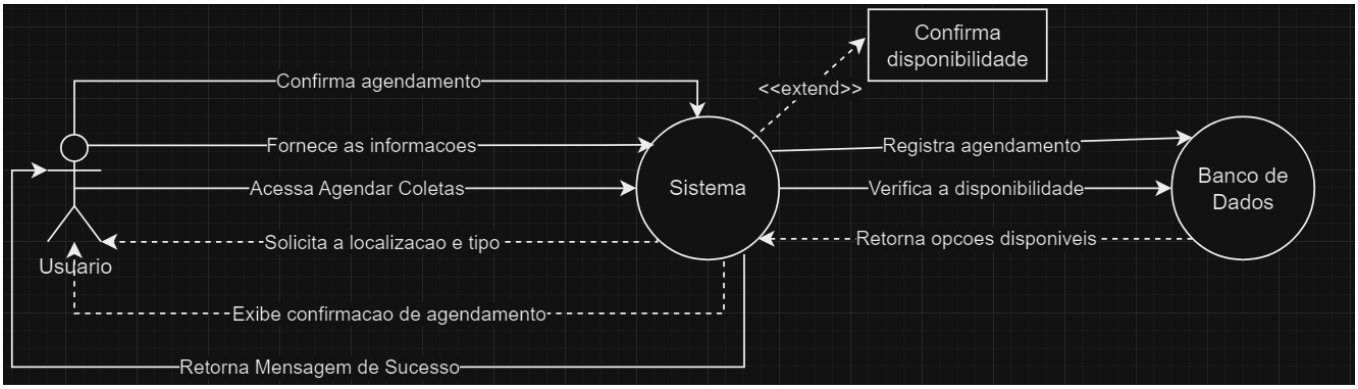
Caso de Uso 2: Receber Dicas de Consumo Responsável



Caso de Uso 3: Monitorar Quantidade de Resíduos Recicláveis



Caso de Uso 4: Agendar Coleta de Resíduos



6. Diagrama de Classes

Classe: Usuario

- **Atributos**:
  - id: int — Identificador único do usuário.
  - nome: string — Nome completo do usuário.
  - email: string — E-mail do usuário para contato e autenticação.
- **Métodos**:
  - registrarResiduos(): void — Método que permite o usuário registrar os resíduos que gera.
  - receberDicas(): void — Método que possibilita o usuário receber dicas sobre consumo responsável.
  - monitorarResiduos(): void — Método para que o usuário visualize seus dados de resíduos gerados.
  - agendarColeta(): void — Método para que o usuário agende a coleta de resíduos recicláveis.
- **Relacionamentos**:
  - **Associação com Resido**: Um usuário pode registrar vários resíduos, ou seja, há uma relação de **1 para N** entre Usuario e Resido.
  - **Associação com Dica**: O usuário recebe várias dicas personalizadas do sistema, formando uma relação de **1 para N** entre Usuario e Dica.
  - **Associação com Coleta**: O usuário pode agendar várias coletas, o que estabelece uma relação de **1 para N** entre Usuario e Coleta.

Classe: Sistema

- **Atributos**:
  - id: int — Identificador único do sistema.
  - nome: string — Nome do sistema.
- **Métodos**:
  - validarDadosResiduos(): bool — Valida os dados informados pelo usuário sobre os resíduos.
  - gerarRelatorios(): void — Gera relatórios sobre o volume e tipo de resíduos registrados.
  - enviarNotificacoes(): void — Envia notificações personalizadas aos usuários sobre consumo responsável.
  - agendarColeta(): void — Agenda coletas de resíduos recicláveis com base na disponibilidade.
- **Relacionamentos**:
  - **Associação com Resido**: O sistema valida os dados dos resíduos registrados pelos usuários, formando uma relação de **1 para N** entre Sistema e Resido.
  - **Associação com Dica**: O sistema gera dicas de consumo responsável com base nos resíduos registrados, criando uma relação de **1 para N** entre Sistema e Dica.
  - **Associação com Coleta**: O sistema agenda coletas de resíduos, estabelecendo uma relação de **1 para N** entre Sistema e Coleta.

Classe: Resido

- **Atributos**:
  - id: int — Identificador único do resíduo.
  - tipo: string — Tipo de resíduo (ex: plástico, papel, orgânico).
  - quantidade: float — Quantidade do resíduo registrado.
  - dataRegistro: Date — Data e horário do registro do resíduo.
- **Métodos**:
  - validarDados(): bool — Valida os dados informados sobre o resíduo antes de ser registrado no sistema.
- **Relacionamentos**:
  - **Associação com Usuario**: Um usuário pode registrar vários resíduos, formando uma relação de **1 para N** entre Usuario e Resido.
  - **Associação com Sistema**: O sistema valida os dados de resíduos, criando uma relação de **1 para N** entre Sistema e Resido.

Classe: Coleta

- **Atributos**:
  - id: int — Identificador único da coleta.
  - data: Date — Data da coleta.
  - horario: string — Horário agendado para a coleta.
  - localizacao: string — Localização onde o resíduo será coletado.
- **Métodos**:
  - confirmarDisponibilidade(): bool — Confirma se o horário e local da coleta estão disponíveis.
  - confirmarColeta(): void — Confirma o agendamento da coleta e informa o usuário.
- **Relacionamentos**:
  - **Associação com Usuario**: Um usuário pode agendar várias coletas, estabelecendo uma relação de **1 para N** entre Usuario e Coleta.
  - **Associação com Sistema**: O sistema é responsável por agendar e gerenciar as coletas, criando uma relação de **1 para N** entre Sistema e Coleta.

Classe: Dica

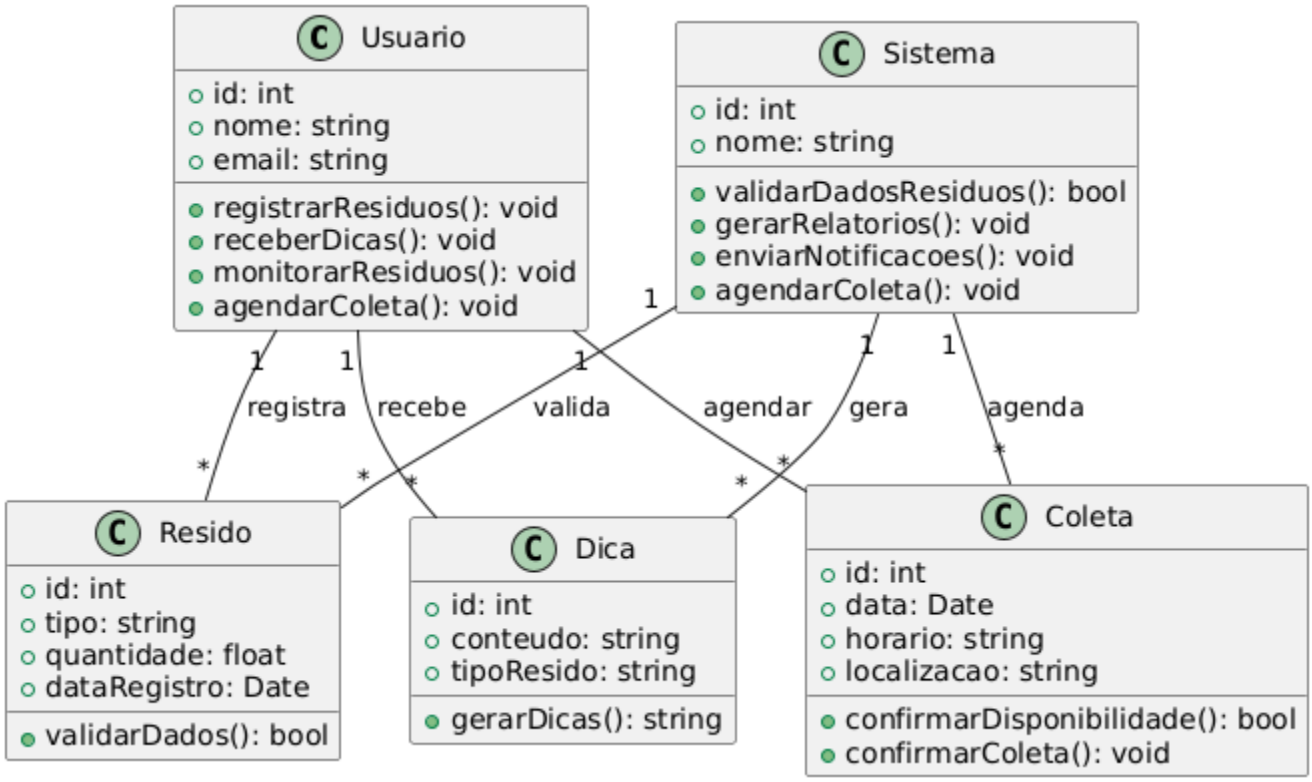
- **Atributos**:
    - id: int — Identificador único da dica.
    - conteudo: string — Conteúdo da dica sobre consumo responsável.
    - tipoResido: string — Tipo de resíduo para o qual a dica é direcionada.
- **Métodos**:
    - gerarDicas(): string — Gera uma lista de dicas personalizadas com base nos resíduos do usuário.
- **Relacionamentos**:
    - **Associação com Usuario**: O usuário pode receber várias dicas, estabelecendo uma relação de **1 para N** entre Usuario e Dica.
    - **Associação com Sistema**: O sistema gera as dicas com base no comportamento do usuário, criando uma relação de **1 para N** entre Sistema e Dica.

Associação: É a relação entre duas classes, onde objetos de uma classe podem estar associados a objetos de outra. Exemplos:

- **Usuario e Resido:** Um usuário pode registrar múltiplos resíduos.
- **Usuario e Coleta:** Um usuário pode agendar múltiplas coletas.

Herança: Não foi utilizado no modelo, pois todas as classes são independentes e não existe uma hierarquia entre elas.

Agregação/Composição: Não há casos explícitos de agregação ou composição no modelo, já que cada classe é independente e não contém outras classes como parte essencial de sua estrutura. No entanto, poderia ser considerada uma relação de agregação entre Sistema e as outras classes, visto que o sistema interage com os resíduos, coletas e dicas sem necessariamente possuí-los diretamente.



7. Diagrama de Atividades e Diagrama de Estados

Diagrama de Atividades

O Diagrama de Atividades ilustra o fluxo de trabalho no sistema, mostrando como os processos são executados e como eles interagem entre si. Vou estruturar com base nos casos de uso fornecidos.

Fluxo para "Registrar Resíduos":

- **Início**: O usuário acessa a opção "Registrar Resíduos".
- **Atividade**: O sistema solicita o tipo e a quantidade de resíduo.
- **Atividade**: O usuário insere os dados.
- **Decisão**: O sistema valida se os dados são válidos.
    - Se sim: Registra os dados.
    - Se não: Exibe mensagem de erro e solicita correção.
- **Fim**: O sistema confirma o registro com o usuário.

Fluxo para "Receber Dicas de Consumo Responsável":

- **Início**: O usuário acessa a opção "Receber Dicas".
- **Atividade**: O sistema determina o perfil do usuário.
- **Atividade**: O sistema exibe dicas personalizadas ou genéricas.
- **Decisão**: O usuário interage com as dicas.
    - Se sim: Marca como "Lida" ou "Interessante".
    - Se não: Ignora a dica.
- **Fim**: O sistema termina a entrega das dicas.

Fluxo para "Monitorar Quantidade de Resíduos Recicláveis":

- **Início**: O usuário acessa "Monitorar Quantidade de Resíduos".
- **Atividade**: O sistema exibe gráficos com dados de reciclagem.
- **Decisão**: O usuário escolhe o período para análise.
    - Se sim: O sistema atualiza os gráficos com o intervalo escolhido.

    - Se não: O sistema exibe mensagem de erro.
- **Fim**: O usuário visualiza os relatórios.

Fluxo para "Agendar Coleta de Resíduos":

- **Início**: O usuário acessa "Agendar Coleta".
- **Atividade**: O sistema solicita localização e tipo de resíduo.
- **Atividade**: O usuário escolhe a data e horário.
- **Decisão**: O sistema válida a disponibilidade do horário.
    - Se sim: Confirma o agendamento.
    - Se não: Oferece horários alternativos.
- **Fim**: O sistema confirma a coleta agendada.

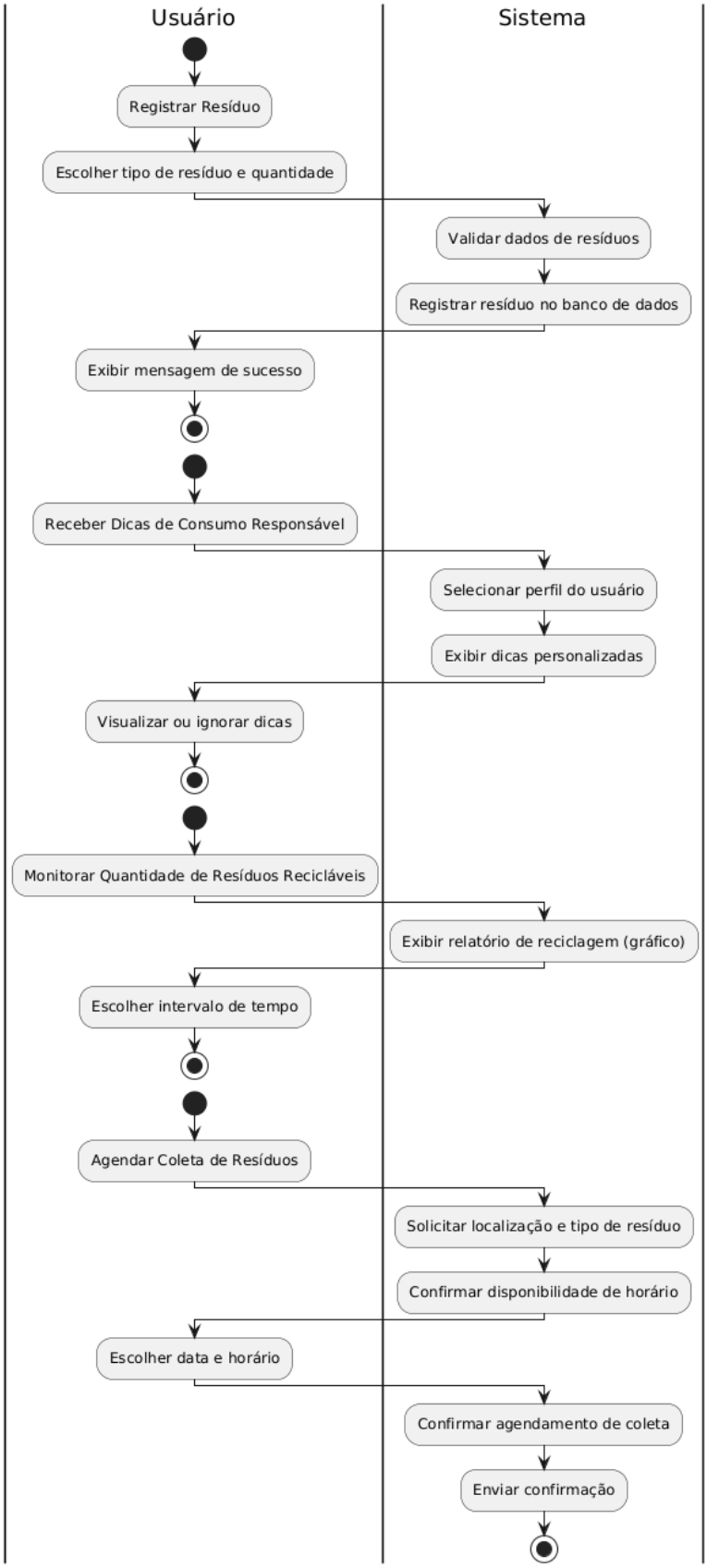

Usuário
Sistema
Registrar Resíduo
Escolher tipo de resíduo e quantidade
Validar dados de resíduos
Registrar resíduo no banco de dados
Exibir mensagem de sucesso
Receber Dicas de Consumo Responsável
Selecionar perfil do usuário
Exibir dicas personalizadas
Visualizar ou ignorar dicas
Monitorar Quantidade de Resíduos Recicláveis
Exibir relatório de reciclagem (gráfico)
Escolher intervalo de tempo
Agendar Coleta de Resíduos
Solicitar localização e tipo de resíduo
Confirmar disponibilidade de horário
Escolher data e horário
Confirmar agendamento de coleta
Enviar confirmação

Diagrama de Estados

O Diagrama de Estados mostra os estados pelos quais os objetos principais (como "Resido", "Usuario", "Coleta", e "Dica") passam durante o ciclo de vida no sistema.

Ciclo de Vida do Objeto "Resido":

- **Estado Inicial**: Resíduo não registrado.
- **Estado 1**: Dados do resíduo inseridos.
- **Estado 2**: Dados validados.
- **Estado 3**: Resíduo registrado.
- **Estado Final**: Resíduo armazenado no banco de dados ou erro de validação.

Ciclo de Vida do Objeto "Usuario":

- **Estado Inicial**: Usuário não registrado ou não autenticado.
- **Estado 1**: Usuário autenticado.
- **Estado 2**: Registrando resíduos.
- **Estado 3**: Recebendo dicas.
- **Estado 4**: Monitorando resíduos recicláveis.
- **Estado 5**: Agendando coleta.
- **Estado Final**: Usuário deslogado ou finalizado o uso do sistema.

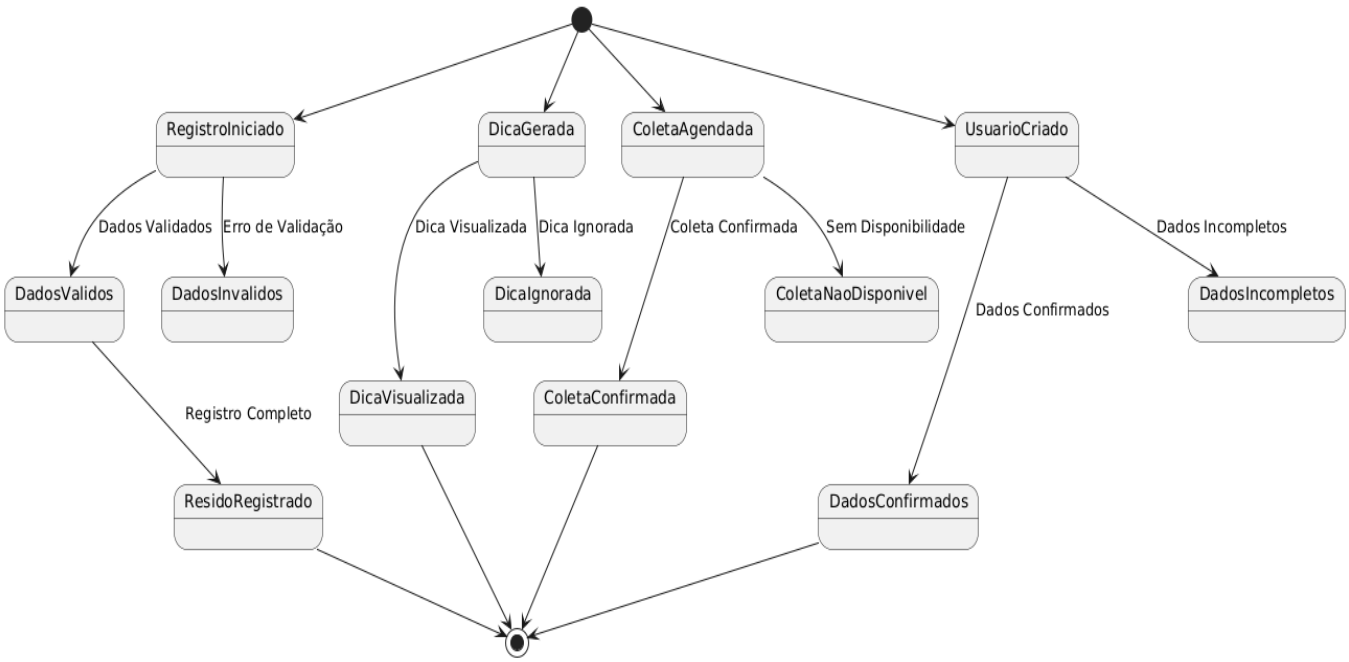
Ciclo de Vida do Objeto "Coleta":

- **Estado Inicial**: Coleta não agendada.
- **Estado 1**: Dados de coleta inseridos.
- **Estado 2**: Validação de disponibilidade de horário.
- **Estado 3**: Coleta agendada.
- **Estado Final**: Coleta realizada ou erro no agendamento.

Ciclo de Vida do Objeto "Dica":

- **Estado Inicial**: Dica não gerada.

- **Estado 1**: Dica gerada pelo sistema.
- **Estado 2**: Dica visualizada pelo usuário.
- **Estado 3**: Dica marcada como lida ou interessante.
- **Estado Final**: Dica descartada ou desconsiderada.



8. Implementação (Feito em Java)

Protótipo Funcional:
O código fornecido já implementa a lógica essencial para o gerenciamento de resíduos, coletas e dicas, usando as classes Usuario, Sistema, Resido, Coleta, Dica, e a classe principal Main.

Funcionalidades:

1. **Registrar Resíduo**: O usuário pode adicionar resíduos informando tipo e quantidade.
2. **Receber Dicas**: O usuário pode escolher entre dicas gerais sobre o projeto ou dicas específicas para seus resíduos registrados.
3. **Monitorar Resíduos**: Exibe todos os resíduos registrados pelo usuário.
4. **Agendar Coleta**: O usuário pode agendar uma coleta informando data, horário e localização.
5. **Exibir Coletas Agendadas**: Mostra as coletas agendadas pelo usuário.
6. **Inserir Dicas**: Permite que o usuário adicione dicas gerais ou específicas para seus resíduos.
7. **Exibir Dados do Usuário**: Mostra as informações do usuário, como nome, e-mail e ID.
8. **Sair**: Encerra o programa.

Documentação Técnica
Principais Classes

1. **Usuario.java**:
    - Armazena os dados do usuário, incluindo nome, e-mail e listas de resíduos, dicas e coletas.
    - Métodos principais: registrarResiduos(), receberDicas(), monitorarResiduos(), agendarColeta(), exibirColetasAgendadas().

2. **Sistema.java**
    - Gerencia os usuários e as dicas gerais do sistema.
    - Métodos principais: adicionarUsuario(), adicionarDicaGeral(), gerarDicas(), obterDicasGerais(), gerarRelatorios().

3. **Resido.java**
    - Representa um resíduo com tipo, quantidade, data de registro e dicas específicas.
    - Métodos principais: adicionarDica(), toString() (para exibição).

4. **Coleta.java**
    - Representa uma coleta com data, horário e localização.
    - Método principal: toString() (para exibição).

5. **Dica.java**
    - Representa uma dica com um conteúdo textual.
    - Método principal: getConteudo().

6. **Main.java**
    - Contém o fluxo principal do sistema, interagindo com o usuário via terminal (CLI).
    - Gerencia a entrada do usuário, mostrando o menu de opções e executando as ações selecionadas.

Como Executar

1. **Testes**: O sistema permite realizar as operações de registrar resíduos, agendar coletas e exibir dicas através do terminal. O sistema é projetado para ser simples e interativo.

Observações Técnicas:

- **Entrada de Dados**: O código utiliza a classe Scanner para coletar dados do usuário via linha de comando.

- **Validação de Dados**: Há verificações simples, como a validação do e-mail (deve conter "@" para ser considerado válido).

- **Uso de Listas**: As listas são usadas para armazenar resíduos, coletas e dicas.

- **Data e Hora**: A classe Date é utilizada para registrar a data e hora das coletas e resíduos.

Possíveis Melhorias:

- **Interface Gráfica**: A interface pode ser melhorada para permitir uma interação mais amigável, talvez com uma GUI (Interface Gráfica de Usuário).
- **Persistência de Dados**: A implementação atual usa apenas dados em memória. Para um sistema mais robusto, seria interessante persistir dados em um banco de dados.
- **Validação de Dados**: Melhorar a validação de entradas, como datas e horários de coleta, para evitar erros de formato.

Conclusão:

A implementação fornecida cumpre o objetivo de simular os principais casos de uso do sistema de gerenciamento de resíduos, com interações baseadas em texto e manipulação de objetos em Java.

Conclusão

O Sistema de Gerenciamento de Resíduos e Reciclagem Doméstica desenvolvido tem como objetivo apoiar os cidadãos na prática de um consumo mais responsável, incentivando o registro adequado de resíduos, o monitoramento da quantidade reciclada e o agendamento de coletas seletivas. Por meio das funcionalidades propostas, o sistema proporciona uma experiência intuitiva e acessível, ao mesmo tempo que oferece uma série de recursos para engajar os usuários na adoção de hábitos mais sustentáveis.

Com base na análise dos requisitos e no desenvolvimento das funcionalidades, foi possível observar que o sistema contribui significativamente para o alcance da meta do Objetivo de Desenvolvimento Sustentável (ODS) 12, ao estimular a prevenção, a reciclagem e o reuso de resíduos. A implementação de dicas personalizadas de consumo responsável e a geração de relatórios detalhados sobre a quantidade de resíduos reciclados são recursos importantes para educar e incentivar os usuários a tomar decisões mais conscientes sobre seus hábitos de consumo.

Além disso, a funcionalidade de agendamento de coletas seletivas promove a prática de descarte adequado, facilitando a logística e incentivando a separação dos resíduos de forma eficiente. A arquitetura do sistema foi projetada para garantir a escalabilidade, a segurança dos dados dos usuários e a acessibilidade em diferentes plataformas, assegurando uma experiência contínua e sem falhas para os usuários.

Por fim, o projeto não só contribui para a redução do impacto ambiental causado pelos resíduos, mas também oferece uma ferramenta prática para a promoção de um futuro mais sustentável, alinhado com as diretrizes globais de preservação ambiental. Ao incentivar a participação ativa dos cidadãos, o sistema representa um passo importante para a criação de um ciclo mais eficiente de consumo e descarte de resíduos.